Double Conformal Space-Time Algebra

by Robert Benjamin Easter

Abstract

This paper introduces the $\mathcal{G}_{4,8}$ Double Conformal Space-Time Algebra (DCSTA). $\mathcal{G}_{4,8}$ DCSTA is a straightforward extension of the $\mathcal{G}_{8,2}$ Double Conformal / Darboux Cyclide Geometric Algebra (DCGA). $\mathcal{G}_{4,8}$ DCSTA extends $\mathcal{G}_{8,2}$ DCGA with space-time boost operations and differential operators for differentiation with respect to the time-like $w = ct$ direction and time t. The spacetime boost operation can implement anisotropic dilation (directed non-uniform scaling) of quadric surface entities. Quadric surface entities can be boosted into moving surfaces with constant velocities that display the length contraction effect of special relativity. To demonstrate $\mathcal{G}_{4,8}$ DCSTA as concrete mathematics with possible applications, this paper includes sample code and example calculations using the symbolic computer algebra system *SymPy*.

Keywords: conformal geometric algebra, space-time algebra, Clifford algebra

A.M.S. subject classification: 15A66, 83A05, 53A30, 14J70, 51K05

1 Introduction

This paper[1] introduces the $\mathcal{G}_{4,8}$ Double Conformal Space-Time Algebra (DCSTA), which is a straightforward extension of the $\mathcal{G}_{8,2}$ Double Conformal / Darboux Cyclide Geometric Algebra (DCGA) into spacetime. $\mathcal{G}_{8,2}$ DCGA is introduced in the paper [6] and discussed further in the papers [4] and [5].

$\mathcal{G}_{4,8}$ DCSTA may offer new mathematical methods for some applications. However, the high-dimensionality of DCSTA incurs high computational cost and applications may require an efficient implementation using optimized hardware and software for DCSTA. Other works on algebras similar to $\mathcal{G}_{4,8}$ DCSTA may exist in the mathematical physics literature, but no specific works essentially the same as $\mathcal{G}_{4,8}$ DCSTA were known by this author at the time of researching and writing this paper.

$\mathcal{G}_{4,8}$ DCSTA is an application of the $\mathcal{G}_{4,8}$ Geometric Algebra. Geometric Algebra is introduced in the book [10] by Hestenes and Sobczyk.

$\mathcal{G}_{4,8}$ DCSTA contains two copies of the $\mathcal{G}_{2,4}$ Conformal Space-Time Algebra (CSTA). $\mathcal{G}_{2,4}$ CSTA is introduced by C.J.L. Doran and A.N. Lasenby in [2] as the *spacetime conformal group*. $\mathcal{G}_{2,4}$ CSTA embeds the $\mathcal{G}_{1,3}$ Space-Time Algebra (STA). $\mathcal{G}_{1,3}$ STA is introduced by Hestenes in [9].

$\mathcal{G}_{4,8}$ DCSTA has twelve unit vector elements $\mathbf{e}_i : 1 \leq i \leq 12$ with metric signature

$$[1, -1, -1, -1, 1, -1, 1, -1, -1, -1, 1, -1].$$

The first six elements are in the $\mathcal{G}_{2,4}$ Conformal Space-Time Algebra 1 (CSTA1)

$$\mathbf{e}_i^2 = \begin{cases} 1 & : \ i \in \{1, 5\} \\ -1 & : \ i \in \{2, 3, 4, 6\}. \end{cases}$$

1. First version v1, *February 10, 2016*, submitted to `http://vixra.org/author/robert_b_easter`. This version may be superseded at the above link by newer revised versions.

The next six elements are in the $\mathcal{G}_{2,4}$ Conformal Space-Time Algebra 2 (CSTA2)

$$\mathbf{e}_i^2 \;=\; \begin{cases} 1 & : \; i \in \{7, 11\} \\ -1 & : \; i \in \{8, 9, 10, 12\}. \end{cases}$$

$\mathcal{G}_{2,4}$ CSTA1 embeds the four elements of the $\mathcal{G}_{1,3}$ Space-Time Algebra 1 (STA1)

$$\mathbf{e}_i^2 \;=\; \begin{cases} 1 & : \; i \in \{1\} \\ -1 & : \; i \in \{2, 3, 4\}. \end{cases}$$

$\mathcal{G}_{2,4}$ CSTA2 embeds the four elements of the $\mathcal{G}_{1,3}$ Space-Time Algebra 2 (STA2)

$$\mathbf{e}_i^2 \;=\; \begin{cases} 1 & : \; i \in \{7\} \\ -1 & : \; i \in \{8, 9, 10\}. \end{cases}$$

$\mathcal{G}_{1,3}$ STA1 contains the three elements of the $\mathcal{G}_{0,3}$ Space Algebra 1 (SA1)

$$\mathbf{e}_i^2 \colon i \in \{2, 3, 4\} \;=\; -1.$$

$\mathcal{G}_{1,3}$ STA2 contains the three elements of the $\mathcal{G}_{0,3}$ Space Algebra 2 (SA2)

$$\mathbf{e}_i^2 \colon i \in \{8, 9, 10\} \;=\; -1.$$

The STA elements, the DIRAC gammas $\boldsymbol{\gamma}_i$ and PAULI sigmas $\boldsymbol{\sigma}_i$, can be defined in STA1 as

$$\boldsymbol{\gamma}_i \;=\; \begin{cases} \mathbf{e}_{i+1} & : \; i \in \{0, 1, 2, 3\} \\ \boldsymbol{\gamma}_0\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3 & : \; i = 5 \end{cases}$$

$$\begin{aligned} \boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_x &= \boldsymbol{\gamma}_1\boldsymbol{\gamma}_0 \\ \boldsymbol{\sigma}_2 = \boldsymbol{\sigma}_y &= \boldsymbol{\gamma}_2\boldsymbol{\gamma}_0 \\ \boldsymbol{\sigma}_3 = \boldsymbol{\sigma}_z &= \boldsymbol{\gamma}_3\boldsymbol{\gamma}_0. \end{aligned}$$

The STA elements $\boldsymbol{\gamma}_0$, $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ that are introduced in [9] are used in all general discussions of STA. The STA elements can be identified with either the STA1 or STA2 elements. The elements $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ can also be denoted $\boldsymbol{\gamma}_x$, $\boldsymbol{\gamma}_y$, and $\boldsymbol{\gamma}_z$ when emphasizing their usage as the conventional x, y, and z spatial directions.

2 Space Algebra (SA)

The $\mathcal{G}_{0,3}$ Space Algebra (SA) is very similar to the $\mathcal{G}_3$ Algebra of Physical Space (APS). Vectors in SA square negative, which causes sign flips in many formulas adapted from APS or DCGA. While DCGA has two Euclidean $\mathcal{G}_3$ APS spaces, DCSTA has two anti-Euclidean $\mathcal{G}_{0,3}$ SA spaces.

The subscript $\mathcal{S}$ denotes an element or operation in generic SA. The subscript $\mathcal{S}^1$ denotes an element or operation in SA1. The subscript $\mathcal{S}^2$ denotes an element or operation in SA2. In most formulas, these subscripts can simply be substituted to write formulas in SA, SA1, and SA2. Such duplication of similar formulas in each representation is avoided unless it adds clarity to the discussion.

The 3-D spatial vectors in $\mathcal{G}_{0,3}$ SA are generally **bold** lowercase letters, such as $\mathbf{p} = \mathbf{p}_{\mathcal{S}}$. The 4-D spacetime vectors in $\mathcal{G}_{1,3}$ STA are generally ***bold italic*** lowercase letters, such as $\boldsymbol{p} = \boldsymbol{p}_{\mathcal{M}}$.

2.1 SA unit pseudoscalar

The SA 3-vector *unit pseudoscalar* $\mathbf{I}_\mathcal{S}$ with signature $(---)$ is

$$\begin{aligned}\mathbf{I}_\mathcal{S} &= \boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3 \\ \mathbf{I}_\mathcal{S}^\sim &= (-1)^{3(3-1)/2}\mathbf{I}_\mathcal{S} = -\mathbf{I}_\mathcal{S} \\ \mathbf{I}_\mathcal{S}^2 &= -\mathbf{I}_\mathcal{S}\mathbf{I}_\mathcal{S}^\sim = 1 \\ \mathbf{I}_\mathcal{S}^{-1} &= \mathbf{I}_\mathcal{S}.\end{aligned}$$

The SA1 3-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{S}^1}$ with signature $(---)$ is

$$\mathbf{I}_{\mathcal{S}^1} = \mathbf{e}_2\mathbf{e}_3\mathbf{e}_4.$$

The SA2 3-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{S}^2}$ with signature $(---)$ is

$$\mathbf{I}_{\mathcal{S}^2} = \mathbf{e}_8\mathbf{e}_9\mathbf{e}_{10}.$$

The notation $A^\sim$ is the *reverse* of A. The $\mathcal{G}_{0,3}$ SA unit pseudoscalar $\mathbf{I}_\mathcal{S}$ is its own inverse and squares to 1 as a *hyperbolic unit*. In $\mathcal{G}_3$ APS, the unit pseudoscalar squares to -1 and is an *imaginary unit*. This difference affects how the SA dualization operations are defined.

2.2 SA dualization

The SA *dual* $A_\mathcal{S}^{*\mathcal{S}}$ of an SA multivector $A_\mathcal{S}$ is

$$A_\mathcal{S}^* = A_\mathcal{S}^{*\mathcal{S}} = A_\mathcal{S}\mathbf{I}_\mathcal{S}^\sim = -A_\mathcal{S}\mathbf{I}_\mathcal{S} = -A_\mathcal{S}\mathbf{I}_\mathcal{S}^{-1}.$$

The SA *undual* $A_\mathcal{S}$ of a dual SA multivector $A_\mathcal{S}^{*\mathcal{S}}$ is

$$A_\mathcal{S} = A_\mathcal{S}^*\mathbf{I}_\mathcal{S}^\sim = A_\mathcal{S}\mathbf{I}_\mathcal{S}^\sim\mathbf{I}_\mathcal{S}^\sim = A_\mathcal{S}\mathbf{I}_\mathcal{S}\mathbf{I}_\mathcal{S}^{-1}$$

The SA dual and undual operations are the same, and the SA dualization is an *involution*.

The notation $A_\mathcal{S}$ denotes an element of the algebra denoted by $\mathcal{S}$, which is SA. In later sections, we will encounter the algebras denoted by $\mathcal{M}$, $\mathcal{C}$, and $\mathcal{D}$, which are STA, CSTA, and DCSTA, respectively. For the subalgebras $\mathcal{S}$, $\mathcal{M}$, and $\mathcal{C}$ of $\mathcal{D}$, there are two copies of them in $\mathcal{D}$, which are denoted by $\mathcal{S}^1$ and $\mathcal{S}^2$ and similarly for other subalgebras that have a double in DCSTA. For example, the Pauli subalgebra of STA is denoted $\mathcal{P}$, and there are $\mathcal{P}^1$ in $\mathcal{M}^1$ and $\mathcal{P}^2$ in $\mathcal{M}^2$.

The explicit dualization notation $A_\mathcal{S}^{*\mathcal{S}}$ denotes the dual of $A_\mathcal{S}$ in subspace $\mathcal{S}$ using the unit pseudoscalar $\mathbf{I}_\mathcal{S}$ of the subspace $\mathcal{S}$. The implicit dualization notation $A_\mathcal{S}^*$ denotes the same dualization as indicated by the subscript $\mathcal{S}$.

To introduce the notation further, the explicit dualizations are

$$A^* = \begin{cases} A_\mathcal{S}^{*\mathcal{S}} = A_\mathcal{S}^* = -A_\mathcal{S}\mathbf{I}_\mathcal{S}^{-1} & : \text{SA dualization} \\ A_\mathcal{M}^{*\mathcal{M}} = A_\mathcal{M}^* = A_\mathcal{M}\mathbf{I}_\mathcal{M}^{-1} & : \text{STA dualization} \\ A_\mathcal{C}^{*\mathcal{C}} = A_\mathcal{C}^* = A_\mathcal{C}\mathbf{I}_\mathcal{C}^{-1} & : \text{CSTA dualization} \\ A_\mathcal{D}^{*\mathcal{D}} = A_\mathcal{D}^* = A_\mathcal{D}\mathbf{I}_\mathcal{D}^{-1} & : \text{DCSTA dualization.} \end{cases}$$

Duals are typically the result of division by the unit pseudoscalar. The SA dualization is defined as division by the negative unit pseudoscalar, and the reason is explained in (§2.6) on the SA rotor. These dualizations are discussed further in later sections.

2.3 SA test vector

The symbolic SA *test vector* $\mathbf{t}_{\mathcal{S}}$ is defined on the basis of the DIRAC gammas [9] as

$$\mathbf{t} = \mathbf{t}_{\mathcal{S}} \;=\; x\boldsymbol{\gamma}_1 + y\boldsymbol{\gamma}_2 + z\boldsymbol{\gamma}_3.$$

The symbolic SA1 *test vector* $\mathbf{t}_{\mathcal{S}^1}$ is defined as

$$\mathbf{t}_{\mathcal{S}^1} \;=\; x\mathbf{e}_2 + y\mathbf{e}_3 + z\mathbf{e}_4.$$

The symbolic SA2 *test vector* $\mathbf{t}_{\mathcal{S}^2}$ is defined as

$$\mathbf{t}_{\mathcal{S}^2} \;=\; x\mathbf{e}_8 + y\mathbf{e}_9 + z\mathbf{e}_{10}.$$

The *symbolic* scalars x, y, and z are the conventional coordinates in space. Numerical scalars are denoted p_x, p_y, and p_z for a vector $\mathbf{p}$. This distinction between symbolic values and numerical values is helpful in symbolic computations. Symbolic computations using a symbolic computer algebra software, such as *SymPy* [13] with the $\mathcal{GA}$lgebra [1] module, can assist in the study of DCSTA and other high-dimensional Geometric Algebras.

A test vector, or other test entity, holds symbolic coordinates and parameters. A non-test vector, or other non-test entity, holds numeric coordinates and parameters. A non-test entity, or simply an actual *entity*, can be evaluated against a symbolic test entity to obtain the symbolic algebraic expression, or implicit surface function, that is represented by the entity.

2.4 SA spatial velocity vector

An SA *spatial velocity vector* $\mathbf{v}_{\mathcal{S}}$ has the form

$$\mathbf{v} = \mathbf{v}_{\mathcal{S}} \;=\; v_x\boldsymbol{\gamma}_1 + v_y\boldsymbol{\gamma}_2 + v_z\boldsymbol{\gamma}_3.$$

An SA1 *spatial velocity vector* $\mathbf{v}_{\mathcal{S}^1}$ has the form

$$\mathbf{v}_{\mathcal{S}^1} \;=\; v_x\mathbf{e}_2 + v_y\mathbf{e}_3 + v_z\mathbf{e}_4.$$

An SA2 *spatial velocity vector* $\mathbf{v}_{\mathcal{S}^2}$ has the form

$$\mathbf{v}_{\mathcal{S}^2} \;=\; v_x\mathbf{e}_8 + v_y\mathbf{e}_9 + v_z\mathbf{e}_{10}.$$

The scalars v_x, v_y, and v_z are coordinate speeds in the conventional x, y, and z spatial directions.

The vector units $\mathbf{e}_1$ and $\mathbf{e}_7$ are in STA1 and STA2, respectively, where they serve as the unit directions for light at coordinate speed $v_w = c$ in the conventional $w = ct$ time direction. Time is measured in distance that light travels, and clock time is a rescaling of time by dividing out the constant speed of light c.

In SPECIAL RELATIVITY, the non-negative *norm* of an SA velocity

$$\|\mathbf{v}_{\mathcal{S}}\| \;=\; \sqrt{\mathbf{v}_{\mathcal{S}} \cdot \mathbf{v}_{\mathcal{S}}^{\dagger}} = \sqrt{-\mathbf{v}_{\mathcal{S}}^2} = \sqrt{v_x^2 + v_y^2 + v_z^2}$$

must not exceed light speed c

$$0 \le \|\mathbf{v}_{\mathcal{S}}\| \le c.$$

The *conjugate* $\mathbf{v}^{\dagger}$ is discussed by PERWASS in [11] as anti-Euclidean subspace conjugation.

The notations as used by Hestenes in [9] are *not* adopted here and conflict with the notations as they are adopted here. In [9], the notation $\mathbf{v}^\dagger$ is called *hermitian conjugation* and is the *reverse* of an element in STA. The *reverse* $\mathbf{v}^\sim$ is the notation that is adopted here. In [9], the notation $\mathbf{v}^*$ is called *space conjugation* and is anti-Euclidean subspace conjugation in STA. The notation $\mathbf{v}^*$ is adopted here as the *dual* of $\mathbf{v}$. The *conjugate* $\mathbf{v}^\dagger$ is adopted here, following Perwass in [11], and is discussed by this author in [7].

In general, a conjugation is an operation that selectively changes the signs on only certain elements and there are many kinds of conjugations and notations. It is thought that the notations that have been adopted here are the ones most commonly adopted in the current literature on Geometric Algebra. The notations of Hestenes in [9] are widely adopted in physics literature.

The conjugate, or space conjugation, of any STA multivector $A_{\mathcal{M}}$, including any SA multivector, is

$$A^\dagger_{\mathcal{M}} \;=\; \boldsymbol{\gamma}_0 A_{\mathcal{M}} \boldsymbol{\gamma}_0.$$

This conjugation formula is valid for any $\mathcal{G}_{1,q}$ Geometric Algebra, where $\boldsymbol{\gamma}_0 = \mathbf{e}_1$.

2.5 SA spatial position vector

An SA *spatial position vector* $\mathbf{p}_{\mathcal{S}}$ has the form

$$\mathbf{p} = \mathbf{p}_{\mathcal{S}} \;=\; p_x \boldsymbol{\gamma}_1 + p_y \boldsymbol{\gamma}_2 + p_z \boldsymbol{\gamma}_3.$$

An SA1 *spatial position vector* $\mathbf{p}_{\mathcal{S}^1}$ has the form

$$\mathbf{p}_{\mathcal{S}^1} \;=\; p_x \mathbf{e}_2 + p_y \mathbf{e}_3 + p_z \mathbf{e}_4.$$

An SA2 *spatial position vector* $\mathbf{p}_{\mathcal{S}^2}$ has the form

$$\mathbf{p}_{\mathcal{S}^2} \;=\; p_x \mathbf{e}_8 + p_y \mathbf{e}_9 + p_z \mathbf{e}_{10}.$$

The scalars p_x, p_y, and p_z are coordinate positions in the conventional x, y, and z spatial directions.

The vector units $\mathbf{e}_1$ and $\mathbf{e}_7$ are in STA1 and STA2, respectively, where they serve as the unit directions for light at coordinate position $p_w = v_w t = ct$ in the conventional $w = ct$ time direction.

In special relativity, an SA position vector has the form

$$\mathbf{p}_{\mathcal{S}} \;=\; \mathbf{v}_{\mathcal{S}} t$$

where $\mathbf{p}_{\mathcal{S}} = 0$ at time $t = 0$. This notation follows the simple statement, *position is the product of velocity and time*.

2.6 SA rotor

A rotation operator R, called a *rotor*, can be understood in terms of ratios, or products, of unit vectors, which are called *versors*. A rotor is isomorphic to a quaternion versor as discussed at length by this author in [7]. The concept of versors is generalized to k-versors in [10]. A k-versor is the product of k unit vectors. In DCSTA, we will encounter 4-versors for rotation, translation, dilation, and boost.

In SA, the unit bivector rotor elements are the ratios

$$\begin{aligned}\mathbf{i}=\mathbf{k}/\mathbf{j} &\Rightarrow \gamma_3/\gamma_2=-\gamma_3\gamma_2\\ \mathbf{j}=\mathbf{i}/\mathbf{k} &\Rightarrow \gamma_1/\gamma_3=-\gamma_1\gamma_3\\ \mathbf{k}=\mathbf{j}/\mathbf{i} &\Rightarrow \gamma_2/\gamma_1=-\gamma_2\gamma_1.\end{aligned}$$

The SA duals of the SA unit vector elements are

$$\begin{aligned}\gamma_1^* &= \gamma_1\mathbf{I}_{\mathcal{S}}^{\sim}=-\gamma_1\mathbf{I}_{\mathcal{S}}=-\gamma_1\mathbf{I}_{\mathcal{S}}^{-1}=-\gamma_1(\gamma_1\gamma_2\gamma_3)=-\gamma_3\gamma_2\Rightarrow\mathbf{i}\\ \gamma_2^* &= \gamma_2\mathbf{I}_{\mathcal{S}}^{\sim}=-\gamma_2\mathbf{I}_{\mathcal{S}}=-\gamma_2\mathbf{I}_{\mathcal{S}}^{-1}=-\gamma_2(\gamma_1\gamma_2\gamma_3)=-\gamma_1\gamma_3\Rightarrow\mathbf{j}\\ \gamma_3^* &= \gamma_3\mathbf{I}_{\mathcal{S}}^{\sim}=-\gamma_3\mathbf{I}_{\mathcal{S}}=-\gamma_3\mathbf{I}_{\mathcal{S}}^{-1}=-\gamma_3(\gamma_1\gamma_2\gamma_3)=-\gamma_2\gamma_1\Rightarrow\mathbf{k}.\end{aligned}$$

The SA dualization is defined such that the isomorphism to quaternion units is via duals.

The dual of an SA vector $\mathbf{x}_{\mathcal{S}}$ is

$$\mathbf{x}^*=\mathbf{x}_{\mathcal{S}}^* = -\mathbf{x}_{\mathcal{S}}\mathbf{I}_{\mathcal{S}}^{-1}.$$

The dual SA vector $\mathbf{x}^*$ is the rotor element or logarithm of a rotor $R=e^{\frac{1}{2}\mathbf{x}^*}$.

Given the SA unit vector

$$\hat{\mathbf{x}}=\hat{\mathbf{x}}_{\mathcal{S}} = \frac{\mathbf{x}_{\mathcal{S}}}{\|\mathbf{x}_{\mathcal{S}}\|}=\frac{\mathbf{x}_{\mathcal{S}}}{\sqrt{-\mathbf{x}_{\mathcal{S}}^2}}$$

as the *axis* of rotation, and $\theta=\|\mathbf{x}_{\mathcal{S}}\|$ as the non-negative *angle* of rotation, then the *rotor* $R_{\mathcal{S}}$ for the rotation is

$$\begin{aligned}R_{\mathcal{S}} &= e^{\frac{1}{2}\mathbf{x}^*}=e^{\frac{1}{2}\theta\hat{\mathbf{x}}_{\mathcal{S}}^*}=e^{\frac{1}{2}\theta\hat{\mathbf{x}}_{\mathcal{S}}\mathbf{I}_{\mathcal{S}}^{\sim}}\\ &= \cos\left(\frac{1}{2}\theta\right)+\sin\left(\frac{1}{2}\theta\right)\hat{\mathbf{x}}_{\mathcal{S}}\mathbf{I}_{\mathcal{S}}^{\sim}\\ &= \cos\left(\frac{1}{2}\theta\right)-\sin\left(\frac{1}{2}\theta\right)\hat{\mathbf{x}}_{\mathcal{S}}\mathbf{I}_{\mathcal{S}}^{-1}\end{aligned}$$

where $(\hat{\mathbf{x}}_{\mathcal{S}}^*)^2=-1$. The axis $\hat{\mathbf{x}}_{\mathcal{S}}$ and pseudoscalar $\mathbf{I}_{\mathcal{S}}$ may be in SA1 or SA2 by changing subscript $\mathcal{S}$ to subscript $\mathcal{S}^1$ or $\mathcal{S}^2$, respectively.

The *rotor operation* that rotates the SA multivector $A_{\mathcal{S}}$ around the axis $\hat{\mathbf{x}}_{\mathcal{S}}$ by angle θ is

$$A'_{\mathcal{S}} = R_{\mathcal{S}}A_{\mathcal{S}}R_{\mathcal{S}}^{-1}=R_{\mathcal{S}}A_{\mathcal{S}}R_{\mathcal{S}}^{\sim}.$$

The SA multivector $A_{\mathcal{S}}$ is typically a vector $\mathbf{a}_{\mathcal{S}}$, but it can be any multivector in SA. The vectors $\mathbf{a}_{i,j}$ of any $k(i)$-blade, $k(i)\in\{1,2,3\}$

$$\mathbf{A}_i = \bigwedge_{j=1}^{k(i)}\mathbf{a}_{i,j}$$

that is a term of

$$A_{\mathcal{S}} = \sum_i \mathbf{A}_i$$

are each rotated

$$A'_{\mathcal{S}}=R_{\mathcal{S}}A_{\mathcal{S}}R_{\mathcal{S}}^{-1} = \sum_i\left(\bigwedge_{j=1}^{k(i)}(R_{\mathcal{S}}\mathbf{a}_{i,j}R_{\mathcal{S}}^{-1})\right)$$

by the rotor operation by a process called *versor outermorphism*, which is discussed by PERWASS in [11] and by this author in[7].

The sense of positive rotation around an axis usually follows the *right-hand rule* on a right-handed axes model. The sense of positive rotation around an axis follows the similar *left-hand rule* on a left-handed axes model. The choice of axes model does *not* affect the rotation mathematics, but it affects the orientation, or handedness, of the axes and the *interpretation* of rotation results on the chosen axes model.

3 Space-Time Algebra (STA)

Space-Time Algebra (STA) is introduced in the book *Space-Time Algebra* by David Hestenes [9]. STA is also called Dirac Algebra (DA). As explained in [9], the *spacetime split* generates a Pauli Algebra (PA) on a unit bivector basis that could be used instead of, or in addition to, the Space Algebra (SA). PA is isomorphic to $\mathcal{G}_3$ APS. DCSTA contains two STA subalgebras, STA1 and STA2.

The $\mathcal{M}$ is for Minkowski spacetime (1,3) and is the subscript that denotes an element or operation in STA. The subscript $\mathcal{M}^1$ denotes an element or operation in STA1. The subscript $\mathcal{M}^2$ denotes an element or operation in STA2.

3.1 STA elements

3.1.1 Dirac gammas and Pauli sigmas in STA

The Dirac gammas and Pauli sigmas can be defined in STA1 as

$$\gamma_i = \begin{cases} \mathbf{e}_{i+1} & : \ i \in \{0,1,2,3\} \\ \gamma_0\gamma_1\gamma_2\gamma_3 & : \ i = 5 \end{cases}$$

$$\begin{aligned} \boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_x &= \gamma_1\gamma_0 \\ \boldsymbol{\sigma}_2 = \boldsymbol{\sigma}_y &= \gamma_2\gamma_0 \\ \boldsymbol{\sigma}_3 = \boldsymbol{\sigma}_z &= \gamma_3\gamma_0. \end{aligned}$$

The STA elements can also be defined similarly in STA2. The Dirac gammas and Pauli sigmas are represented as matrices in other literature, but they have multivector representations in STA. See reference [9] for more information about these representations.

The gammas are used to denote elements in STA subscripted $\mathcal{M}$, but it should be understood that all discussions of STA apply similarly in STA1 and STA2 by changing subscripting and elements

$$\begin{array}{lcll} \text{STA} & \Leftrightarrow & \text{STA1} & \Leftrightarrow \ \text{STA2} \\ \mathcal{M} & \Leftrightarrow & \mathcal{M}^1 & \Leftrightarrow \ \mathcal{M}^2 \\ \gamma_0 & \Leftrightarrow & \mathbf{e}_1 & \Leftrightarrow \ \mathbf{e}_7 \\ \gamma_1 & \Leftrightarrow & \mathbf{e}_2 & \Leftrightarrow \ \mathbf{e}_8 \\ \gamma_2 & \Leftrightarrow & \mathbf{e}_3 & \Leftrightarrow \ \mathbf{e}_9 \\ \gamma_3 & \Leftrightarrow & \mathbf{e}_4 & \Leftrightarrow \ \mathbf{e}_{10}. \end{array}$$

3.1.2 STA unit pseudoscalar

The $\mathcal{G}_{1,3}$ STA 4-vector *unit pseudoscalar* $\mathbf{I}_\mathcal{M}$ with signature $(+---)$ is

$$\begin{aligned} \mathbf{I}_\mathcal{M} &= \gamma_0\gamma_1\gamma_2\gamma_3 = \gamma_5 \\ \mathbf{I}_\mathcal{M}^\sim &= (-1)^{4(4-1)/2}\mathbf{I}_\mathcal{M} = \mathbf{I}_\mathcal{M} \\ \mathbf{I}_\mathcal{M}^2 &= -1 \\ \mathbf{I}_\mathcal{M}^{-1} &= -\mathbf{I}_\mathcal{M} = -\mathbf{I}_\mathcal{M}^\sim. \end{aligned}$$

The $\mathcal{G}_{1,3}$ STA1 4-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{M}^1}$ with signature $(+---)$ is

$$\mathbf{I}_{\mathcal{M}^1} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4.$$

The $\mathcal{G}_{1,3}$ STA2 4-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{M}^2}$ with signature $(+---)$ is

$$\mathbf{I}_{\mathcal{M}^2} = \mathbf{e}_7\mathbf{e}_8\mathbf{e}_9\mathbf{e}_{10}.$$

The PAULI Algebra 1 (PA1) and PAULI Algebra 2 (PA2) are denoted by $\mathcal{P}^1$ and $\mathcal{P}^2$, respectively. The unit pseudoscalars of PA1 and PA2 are

$$\begin{aligned}\mathbf{I}_{\mathcal{P}^1} &= \boldsymbol{\sigma}_{\mathcal{P}^1_1}\boldsymbol{\sigma}_{\mathcal{P}^1_2}\boldsymbol{\sigma}_{\mathcal{P}^1_3} = (\mathbf{e}_2\mathbf{e}_1)(\mathbf{e}_3\mathbf{e}_1)(\mathbf{e}_4\mathbf{e}_1) = \mathbf{I}_{\mathcal{M}^1}\\ \mathbf{I}_{\mathcal{P}^2} &= \boldsymbol{\sigma}_{\mathcal{P}^2_1}\boldsymbol{\sigma}_{\mathcal{P}^2_2}\boldsymbol{\sigma}_{\mathcal{P}^2_3} = (\mathbf{e}_8\mathbf{e}_7)(\mathbf{e}_9\mathbf{e}_7)(\mathbf{e}_{10}\mathbf{e}_7) = \mathbf{I}_{\mathcal{M}^2}.\end{aligned}$$

3.1.3 STA test vector

The symbolic STA *test vector* $\boldsymbol{t}_{\mathcal{M}}$ is defined on the basis of the DIRAC gammas [9] as

$$\boldsymbol{t} = \boldsymbol{t}_{\mathcal{M}} = w\boldsymbol{\gamma}_0 + x\boldsymbol{\gamma}_1 + y\boldsymbol{\gamma}_2 + z\boldsymbol{\gamma}_3 = ct\boldsymbol{\gamma}_0 + \mathbf{t}_{\mathcal{S}} = \boldsymbol{o}_{\mathcal{M}}t + \mathbf{t}_{\mathcal{S}}.$$

The symbolic STA1 *test vector* $\boldsymbol{t}_{\mathcal{M}^1}$ is defined as

$$\boldsymbol{t}_{\mathcal{M}^1} = w\mathbf{e}_1 + x\mathbf{e}_2 + y\mathbf{e}_3 + z\mathbf{e}_4 = ct\mathbf{e}_1 + \mathbf{t}_{\mathcal{S}^1} = \boldsymbol{o}_{\mathcal{M}^1}t + \mathbf{t}_{\mathcal{S}^1}.$$

The symbolic STA2 *test vector* $\boldsymbol{t}_{\mathcal{M}^2}$ is defined as

$$\boldsymbol{t}_{\mathcal{M}^2} = w\mathbf{e}_7 + x\mathbf{e}_8 + y\mathbf{e}_9 + z\mathbf{e}_{10} = ct\mathbf{e}_7 + \mathbf{t}_{\mathcal{S}^2} = \boldsymbol{o}_{\mathcal{M}^2}t + \mathbf{t}_{\mathcal{S}^2}.$$

The symbolic scalars w, x, y, and z are the conventional coordinates in spacetime. The timelike coordinate $w = ct$ is the coordinate position of light at light speed c at time t. The *observer*, as defined in the theory of SPECIAL RELATIVITY, is identified as the symbolic timelike velocity $\boldsymbol{o}_{\mathcal{M}}$.

The symbolic test vector $\boldsymbol{t}_{\mathcal{M}}$ is useful in symbolic computations and will be embedded as the $\mathcal{G}_{2,4}$ CSTA *test point* $\boldsymbol{T}_{\mathcal{C}}$. CSTA1 and CSTA2 test points $\boldsymbol{T}_{\mathcal{C}^1}$ and $\boldsymbol{T}_{\mathcal{C}^2}$, respectively, are wedged to form the $\mathcal{G}_{4,8}$ DCSTA *test point* $\boldsymbol{T}_{\mathcal{D}} = \boldsymbol{T}_{\mathcal{C}^1} \wedge \boldsymbol{T}_{\mathcal{C}^2}$. The DCSTA point value-*extraction elements* T_s are defined as elements that extract values from the DCSTA test point $\boldsymbol{T}_{\mathcal{D}}$ as $s = \boldsymbol{T}_{\mathcal{D}} \cdot T_s$.

3.1.4 STA observer

An STA observer $\boldsymbol{o}_{\mathcal{M}}$ has the normalized form

$$\boldsymbol{o} = \boldsymbol{o}_{\mathcal{M}} = c\boldsymbol{\gamma}_0.$$

An STA1 observer $\boldsymbol{o}_{\mathcal{M}^1}$ has the normalized form

$$\boldsymbol{o}_{\mathcal{M}^1} = c\mathbf{e}_1.$$

An STA2 observer $\boldsymbol{o}_{\mathcal{M}^2}$ has the normalized form

$$\boldsymbol{o}_{\mathcal{M}^2} = c\mathbf{e}_7.$$

In SPECIAL RELATIVITY, the observer is a timelike spacetime velocity, where c is the speed of light. An STA observer $\boldsymbol{o}_{\mathcal{M}}$ can be written on the basis of the DIRAC gammas as

$$\boldsymbol{o} = \boldsymbol{o}_{\mathcal{M}} = c\boldsymbol{\gamma}_0.$$

In International Standard (SI) measurement units, $c = 299,792,458\text{m/s}$. In natural units, which are often convenient for testing calculations on a smaller unit scale, $c = 1$.

3.1.5 STA spatial velocity

An STA *spatial velocity* $\mathbf{v}_{\mathcal{S}}$ has the form

$$\mathbf{v} = \mathbf{v}_{\mathcal{S}} = v_x\boldsymbol{\gamma}_1 + v_y\boldsymbol{\gamma}_2 + v_z\boldsymbol{\gamma}_3.$$

An STA1 *spatial velocity* $\mathbf{v}_{\mathcal{S}^1}$ has the form

$$\mathbf{v}_{\mathcal{S}^1} = v_x\mathbf{e}_2 + v_y\mathbf{e}_3 + v_z\mathbf{e}_4.$$

An STA2 *spatial velocity* $\mathbf{v}_{\mathcal{S}^2}$ has the form

$$\mathbf{v}_{\mathcal{S}^2} = v_x\mathbf{e}_8 + v_y\mathbf{e}_9 + v_z\mathbf{e}_{10}.$$

STA spatial velocities are the same as SA spatial velocities. The v_x, v_y, and v_z are coordinate speeds in the conventional x, y, and z directions.

The non-negative *norm* of an SA spatial velocity $\mathbf{v}_{\mathcal{S}}$ is

$$\|\mathbf{v}_{\mathcal{S}}\| = \sqrt{-\mathbf{v}_{\mathcal{S}}^2} = \sqrt{v_x^2 + v_y^2 + v_z^2}.$$

In special relativity,

$$0 \le \|\mathbf{v}_{\mathcal{S}}\| \le c.$$

The *unit* direction of an STA or SA *spatial velocity* $\mathbf{v}_{\mathcal{S}}$ is

$$\hat{\mathbf{v}}_{\mathcal{S}} = \frac{\mathbf{v}_{\mathcal{S}}}{\|\mathbf{v}_{\mathcal{S}}\|}.$$

3.1.6 STA spatial position

An STA *spatial position* $\mathbf{p}_{\mathcal{S}}$ has the form

$$\mathbf{p} = \mathbf{p}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}t = (v_x\boldsymbol{\gamma}_1 + v_y\boldsymbol{\gamma}_2 + v_z\boldsymbol{\gamma}_3)t = p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3.$$

An STA1 *spatial position* $\mathbf{p}_{\mathcal{S}^1}$ has the form

$$\mathbf{p}_{\mathcal{S}^1} = \mathbf{v}_{\mathcal{S}^1}t = (v_x\mathbf{e}_2 + v_y\mathbf{e}_3 + v_z\mathbf{e}_3)t = p_x\mathbf{e}_2 + p_y\mathbf{e}_3 + p_z\mathbf{e}_3.$$

An STA2 *spatial position* $\mathbf{p}_{\mathcal{S}^2}$ has the form

$$\mathbf{p}_{\mathcal{S}^2} = \mathbf{v}_{\mathcal{S}^2}t = (v_x\mathbf{e}_8 + v_y\mathbf{e}_9 + v_z\mathbf{e}_{10})t = p_x\mathbf{e}_8 + p_y\mathbf{e}_9 + p_z\mathbf{e}_{10}.$$

In special relativity, the time t is the *proper time* of the *observer* $\boldsymbol{o}_{\mathcal{M}}$, and the spatial position must be $\mathbf{p}_{\mathcal{S}} = 0$ at $t = 0$.

3.1.7 STA spacetime velocity

An STA *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}}$ has the normalized form

$$\boldsymbol{v} = \boldsymbol{v}_{\mathcal{M}} = \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}} = c\boldsymbol{\gamma}_0 + (v_x\boldsymbol{\gamma}_1 + v_y\boldsymbol{\gamma}_2 + v_z\boldsymbol{\gamma}_3).$$

An STA1 *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}^1}$ has the normalized form

$$\boldsymbol{v}_{\mathcal{M}^1} = \boldsymbol{o}_{\mathcal{M}^1} + \mathbf{v}_{\mathcal{S}^1} = c\mathbf{e}_1 + (v_x\mathbf{e}_2 + v_y\mathbf{e}_3 + v_z\mathbf{e}_4).$$

An STA2 *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}^2}$ has the normalized form

$$\boldsymbol{v}_{\mathcal{M}^2} \;=\; \boldsymbol{o}_{\mathcal{M}^2} + \mathbf{v}_{\mathcal{S}^2} = c\mathbf{e}_7 + (v_x\mathbf{e}_8 + v_y\mathbf{e}_9 + v_z\mathbf{e}_{10}).$$

In SPECIAL RELATIVITY, a spacetime velocity $\boldsymbol{v}_{\mathcal{M}}$ is the sum of an observer spacetime velocity $\boldsymbol{o}_{\mathcal{M}}$ and a spatial velocity $\mathbf{v}_{\mathcal{S}}$ relative to the observer, where $0 \le \|\mathbf{v}_{\mathcal{S}}\| \le c$.

The *modulus* of an STA *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}}$ is

$$|\boldsymbol{v}_{\mathcal{M}}| \;=\; \sqrt{\boldsymbol{v}_{\mathcal{M}}^2} = \sqrt{c^2 + \mathbf{v}_{\mathcal{S}}^2} = \sqrt{c^2 - \|\mathbf{v}_{\mathcal{S}}\|^2}.$$

The square of a spacetime velocity $\boldsymbol{v}^2 = c^2 - \|\mathbf{v}\|^2$ may be positive or negative, and is said to have *Minkowski signature* or metric $(1,3)$. A spacetime velocity with positive signature $0 < \boldsymbol{v}^2$ is *timelike*, with negative signature $\boldsymbol{v}^2 < 0$ is *spacelike*, and with null signature $\boldsymbol{v}^2 = 0$ is *lightlike*.

The *conjugate* of an STA *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}}$, or any STA multivector, is

$$\boldsymbol{v}_{\mathcal{M}}^\dagger \;=\; \boldsymbol{\gamma}_0 \boldsymbol{v}_{\mathcal{M}} \boldsymbol{\gamma}_0.$$

The *norm* of an STA *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}}$ is

$$\|\boldsymbol{v}_{\mathcal{M}}\| \;=\; \sqrt{\boldsymbol{v}_{\mathcal{M}} \cdot \boldsymbol{v}_{\mathcal{M}}^\dagger} = \sqrt{\boldsymbol{v}_{\mathcal{M}} \cdot (\boldsymbol{\gamma}_0 \boldsymbol{v}_{\mathcal{M}} \boldsymbol{\gamma}_0)} = \sqrt{c^2 - \mathbf{v}_{\mathcal{S}}^2} = \sqrt{c^2 + \|\mathbf{v}_{\mathcal{S}}\|^2}.$$

The *unit*, or *modulus-unit*, of an STA *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}}$ is

$$\hat{\boldsymbol{v}}_{\mathcal{M}} = \frac{\boldsymbol{v}_{\mathcal{M}}}{|\boldsymbol{v}_{\mathcal{M}}|} \;=\; \frac{\boldsymbol{v}_{\mathcal{M}}}{\sqrt{c^2 + \mathbf{v}_{\mathcal{S}}^2}} = \frac{\boldsymbol{v}_{\mathcal{M}}}{\sqrt{c^2 - v_x^2 - v_y^2 - v_z^2}}.$$

The *norm-unit* of an STA *spacetime velocity* $\boldsymbol{v}_{\mathcal{M}}$ is

$$\frac{\boldsymbol{v}_{\mathcal{M}}}{\|\boldsymbol{v}_{\mathcal{M}}\|} \;=\; \frac{\boldsymbol{v}_{\mathcal{M}}}{\sqrt{\boldsymbol{o}_{\mathcal{M}}^2 - \mathbf{v}_{\mathcal{S}}^2}} = \frac{\boldsymbol{v}_{\mathcal{M}}}{\sqrt{c^2 + v_x^2 + v_y^2 + v_z^2}}.$$

The overhat is on the modulus-unit of an STA spacetime vector $\boldsymbol{a}$ with $a_w \neq 0$ as $\hat{\boldsymbol{a}}$, but the overhat is on the norm-unit of an SA spatial vector $\mathbf{a}$ with $a_w = 0$ as $\hat{\mathbf{a}}$. In some contexts, it is explicitly noted when the overhat notation on spacetime vectors is taking the norm-unit.

3.1.8 STA spacetime position

An STA *spacetime position* $\boldsymbol{p}_{\mathcal{M}}$ has the normalized form

$$\boldsymbol{p} = \boldsymbol{p}_{\mathcal{M}} = \boldsymbol{v}_{\mathcal{M}} t \;=\; (\boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}})t = \boldsymbol{o}_{\mathcal{M}} t + \mathbf{p}_{\mathcal{S}}.$$

In SPECIAL RELATIVITY, the spacetime position must be $\boldsymbol{p}_{\mathcal{M}} = 0$ at time $t = 0$. A spacetime position is sometimes called a *particle*.

The overdot notation $\dot{\boldsymbol{p}}_{\mathcal{M}} = \boldsymbol{v}_{\mathcal{M}}$ denotes the *time t derivative* of spacetime position $\boldsymbol{p}_{\mathcal{M}} = \boldsymbol{v}_{\mathcal{M}} t$, where

$$\dot{\boldsymbol{p}}_{\mathcal{M}} \;=\; \partial_t \boldsymbol{p}_{\mathcal{M}} = \frac{\partial \boldsymbol{p}_{\mathcal{M}}}{\partial t} = \frac{\partial}{\partial t} \boldsymbol{v}_{\mathcal{M}} t = \boldsymbol{v}_{\mathcal{M}}.$$

The *modulus-unit*

$$\hat{\boldsymbol{p}}_{\mathcal{M}} \;=\; \frac{\boldsymbol{p}_{\mathcal{M}}}{|\boldsymbol{p}_{\mathcal{M}}|} = \frac{\boldsymbol{p}_{\mathcal{M}}}{\sqrt{\boldsymbol{p}_{\mathcal{M}}^2}} = \frac{\boldsymbol{p}_{\mathcal{M}}}{\sqrt{(ct)^2 - \|\mathbf{p}_{\mathcal{S}}\|^2}} = \frac{\boldsymbol{p}_{\mathcal{M}}}{\sqrt{p_w^2 - p_x^2 - p_y^2 - p_z^2}}$$

and the *norm-unit*

$$\frac{\boldsymbol{p}_{\mathcal{M}}}{\|\boldsymbol{p}_{\mathcal{M}}\|} \;=\; \frac{\boldsymbol{p}_{\mathcal{M}}}{\sqrt{\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{p}_{\mathcal{M}}^\dagger}} = \frac{\boldsymbol{p}_{\mathcal{M}}}{\sqrt{(ct)^2 + \|\mathbf{p}_{\mathcal{S}}\|^2}} = \frac{\boldsymbol{p}_{\mathcal{M}}}{\sqrt{p_w^2 + p_x^2 + p_y^2 + p_z^2}}$$

of an STA spacetime position $\boldsymbol{p}_{\mathcal{M}}$ are similar to those of an STA spacetime velocity $\boldsymbol{v}_{\mathcal{M}}$.

3.2 STA operations

3.2.1 STA spacetime velocity normalization

After boost operations $B_{\mathcal{M}}\boldsymbol{v}_{\mathcal{M}}B_{\widetilde{\mathcal{M}}}$ or a projection $\boldsymbol{v}_{\mathcal{M}}=\mathcal{C}^{-1}(\boldsymbol{V}_{\mathcal{C}})$, an STA velocity $\boldsymbol{v}_{\mathcal{M}}$ may require a normalization of the observer as

$$\boldsymbol{v}'_{\mathcal{M}} \;=\; c\frac{\boldsymbol{v}_{\mathcal{M}}}{\boldsymbol{v}_{\mathcal{M}}\cdot\boldsymbol{\gamma}_0}=c\boldsymbol{\gamma}_0+\mathbf{v}_{\mathcal{S}}=\boldsymbol{o}_{\mathcal{M}}+\mathbf{v}_{\mathcal{S}}.$$

The normalization brings $\boldsymbol{v}_{\mathcal{M}}$ into the normalized form $\boldsymbol{v}'_{\mathcal{M}}$. The prime mark may be dropped following normalization. Normalization scales the *observer* spacetime velocity $\boldsymbol{o}_{\mathcal{M}}$ into the normalized form

$$\boldsymbol{o}_{\mathcal{M}} \;=\; c\boldsymbol{\gamma}_0$$

where time t is the observer's *proper time*. The *observer spacetime position* has the normalized form

$$\boldsymbol{o}_{\mathcal{M}}t \;=\; ct\boldsymbol{\gamma}_0.$$

3.2.2 STA spacetime position normalization

After boost operations $B_{\mathcal{M}}\boldsymbol{p}_{\mathcal{M}}B_{\widetilde{\mathcal{M}}}$ or a projection $\boldsymbol{p}_{\mathcal{M}}=\mathcal{C}^{-1}(\boldsymbol{P}_{\mathcal{C}})$, an STA position $\boldsymbol{p}_{\mathcal{M}}=\boldsymbol{v}_{\mathcal{M}}t$ may require a normalization of the observer as

$$\boldsymbol{p}'_{\mathcal{M}} \;=\; ct\frac{\boldsymbol{p}_{\mathcal{M}}}{\boldsymbol{p}_{\mathcal{M}}\cdot\boldsymbol{\gamma}_0}=ct\boldsymbol{\gamma}_0+\mathbf{v}_{\mathcal{S}}t=(\boldsymbol{o}_{\mathcal{M}}+\mathbf{v}_{\mathcal{S}})t.$$

The normalization brings $\boldsymbol{p}_{\mathcal{M}}$ into the normalized form $\boldsymbol{p}'_{\mathcal{M}}$. The prime mark may be dropped following normalization. Normalization scales the *observer spacetime position* $\boldsymbol{o}_{\mathcal{M}}t$ into the normalized form

$$\boldsymbol{o}_{\mathcal{M}}t \;=\; ct\boldsymbol{\gamma}_0$$

where time t is the observer's *proper time*.

3.2.3 STA dualization

The STA *dual* $A^{*\mathcal{M}}_{\mathcal{M}}$ of an STA multivector $A_{\mathcal{M}}$ is

$$A^{*}_{\mathcal{M}}=A^{*\mathcal{M}}_{\mathcal{M}} \;=\; A_{\mathcal{M}}\mathbf{I}^{-1}_{\mathcal{M}}=-A_{\mathcal{M}}\mathbf{I}_{\mathcal{M}}.$$

The STA *undual* $A_{\mathcal{M}}$ of an STA multivector $A^{*\mathcal{M}}_{\mathcal{M}}$ is

$$A_{\mathcal{M}} \;=\; A^{*}_{\mathcal{M}}\mathbf{I}_{\mathcal{M}}=A_{\mathcal{M}}\mathbf{I}^{-1}_{\mathcal{M}}\mathbf{I}_{\mathcal{M}}.$$

The STA dualization is defined such that, if $A_{\mathcal{M}}$ is a unit polar bivector of the Pauli Algebra (PA) in terms of the Pauli units $\boldsymbol{\sigma}_x$, $\boldsymbol{\sigma}_y$, $\boldsymbol{\sigma}_z$, then its dual is the unit axial bivector rotor element that generates rotations around $A_{\mathcal{M}}$. The PA dualization in STA is isomorphic to the $\mathcal{G}_3$ APS dualization.

In APS, a vector $\mathbf{n}$ and its dual pseudovector $\mathbf{n}^* = \mathbf{n}\mathbf{I}^{-1} = \mathbf{n}\mathbf{e}_3\mathbf{e}_2\mathbf{e}_1$ are polar and axial elements, respectively. An APS rotor R for a rotation around a unit vector axis $\mathbf{n}$ by an angle θ is

$$R = e^{\frac{1}{2}\theta\mathbf{n}^*} = \cos\left(\frac{1}{2}\theta\right) + \sin\left(\frac{1}{2}\theta\right)\mathbf{n}^*.$$

The APS rotor operation is $\mathbf{p}' = R\mathbf{p}R^\sim$. The PA units and dualization in STA work similarly to the APS units and dualization by direct substitution of units.

The choice to use either SA or PA may depend on the application or personal preference. Theorems and formulas in PA, or APS, may require some sign changes when adapted into SA. PA has a bivector basis, while SA has a vector basis.

3.2.4 STA rotor

The STA spatial rotation operator, or *rotor*, $R_\mathcal{M} = R_\mathcal{S}$ is the SA rotor $R_\mathcal{S}$ (§2.6).

The STA 2-versor spatial *rotor* $R_\mathcal{S}$ for rotation in SA space around the SA unit vector *axis* $\hat{\mathbf{x}}_\mathcal{S}$ by *angle* θ is

$$\begin{aligned} R_\mathcal{S} &= e^{\frac{1}{2}\theta\hat{\mathbf{x}}^*_\mathcal{S}} = e^{\frac{1}{2}\theta\hat{\mathbf{x}}_\mathcal{S}\mathbf{I}_\mathcal{S}^\sim} \\ &= \cos\left(\frac{1}{2}\theta\right) + \sin\left(\frac{1}{2}\theta\right)\hat{\mathbf{x}}_\mathcal{S}\mathbf{I}_\mathcal{S}^\sim \\ &= \cos\left(\frac{1}{2}\theta\right) - \sin\left(\frac{1}{2}\theta\right)\hat{\mathbf{x}}_\mathcal{S}\mathbf{I}_\mathcal{S}^{-1}. \end{aligned}$$

The rotor operation

$$A'_\mathcal{M} = R_\mathcal{S} A_\mathcal{M} R_\mathcal{S}^\sim$$

rotates any multivector $A_\mathcal{M}$ in STA as expected in the spatial SA components, but leaves the STA timelike components unchanged.

In PA, the timelike component $w\boldsymbol{\gamma}_0$ becomes a scalar $w\boldsymbol{\gamma}_0\boldsymbol{\gamma}_0 = w = ct$ and is also unaffected by spatial rotations. The PA rotor is the same rotor $R_\mathcal{P} = R_\mathcal{S}$, but arrived at as

$$R_\mathcal{P} = e^{\frac{1}{2}\theta\hat{\mathbf{x}}^*_\mathcal{P}} = e^{\frac{1}{2}\theta\hat{\mathbf{x}}_\mathcal{P}\mathbf{I}_\mathcal{P}^{-1}} = e^{\frac{1}{2}\theta\hat{\mathbf{x}}_\mathcal{S}\boldsymbol{\gamma}_0\mathbf{I}_\mathcal{M}^{-1}} = e^{-\frac{1}{2}\theta\hat{\mathbf{x}}_\mathcal{S}\boldsymbol{\gamma}_0\mathbf{I}_\mathcal{M}} = e^{-\frac{1}{2}\theta\hat{\mathbf{x}}_\mathcal{S}\mathbf{I}_\mathcal{S}} = e^{\frac{1}{2}\theta\hat{\mathbf{x}}_\mathcal{S}\mathbf{I}_\mathcal{S}^\sim}.$$

3.2.5 STA spacetime boost

In this section, the STA boost operator B is derived. In the derivation, subscripting with $\mathcal{M}$ and $\mathcal{S}$ is omitted, but the notation should be understood in context.

An STA boost operation is a hyperbolic rotation operation that can turn an STA observer with spacetime velocity

$$\boldsymbol{o} = c\boldsymbol{\gamma}_0$$

into a moving observer, or particle, with spacetime velocity

$$\boldsymbol{v} = \boldsymbol{o} + \mathbf{v} = c\boldsymbol{\gamma}_0 + \|\mathbf{v}\|\hat{\mathbf{v}}$$

that is consistent with SPECIAL RELATIVITY. A boost can also be applied to a particle that is already moving to change the velocity of the particle, but a boost can never increase the speed of a particle to greater than light speed c relative to any observer.

To derive the boost operation, we can start by defining the *ratio* of spacetime velocities of a particle $\boldsymbol{v}$ and its observer $\boldsymbol{o}$ as the *hyperbolic biradial* $\boldsymbol{v}/\boldsymbol{o} = \boldsymbol{v}\boldsymbol{o}^{-1}$ ("$\boldsymbol{v}$ by $\boldsymbol{o}$"). The term *biradial* was coined by HAMILTON in his original work on Quaternions [8].

The *hyperbolic biradial*

$$\begin{aligned} H &= \boldsymbol{v}\boldsymbol{o}^{-1} = \frac{|\boldsymbol{v}|}{|\boldsymbol{o}|}\widehat{\boldsymbol{v}/\boldsymbol{o}} = \frac{\sqrt{c^2+\mathbf{v}^2}}{c}\hat{H} \\ &= \sqrt{1-\frac{\|\mathbf{v}\|^2}{c^2}}\hat{H} = \sqrt{1-\beta_{\mathbf{v}}^2}\hat{H} = \frac{1}{\gamma_{\mathbf{v}}}\hat{H} \end{aligned}$$

is an operator that *turns* the spacetime velocity of the observer $\boldsymbol{o}$ into the spacetime boost velocity of the boost particle

$$\boldsymbol{p}_1 = \boldsymbol{v}t = (\boldsymbol{o}+\mathbf{v})t$$

that is relative to the same observer $\boldsymbol{o}$ by the one-sided versor operation

$$\boldsymbol{v} = H\boldsymbol{o}.$$

The *natural speed* $\beta_{\mathbf{v}}$ of the velocity $\mathbf{v}$ is

$$\beta_{\mathbf{v}} = \frac{\|\mathbf{v}\|}{c}.$$

The *Lorentz factor* $\gamma_{\mathbf{v}}$ of the velocity $\mathbf{v}$ is

$$\gamma_{\mathbf{v}} = \frac{1}{\sqrt{1-\beta_{\mathbf{v}}^2}}.$$

The *length contraction*, to length L from an initial length L_0 in the direction of boost velocity $\mathbf{v}$, is given by

$$L = \frac{L_0}{\gamma_{\mathbf{v}}} = L_0\sqrt{1-\beta_{\mathbf{v}}^2}.$$

The *dilation factor* of the velocity $\mathbf{v}$ is

$$d = \frac{1}{\gamma_{\mathbf{v}}} = \sqrt{1-\beta_{\mathbf{v}}^2}.$$

For a dilation factor d, the required natural speed is $\beta_{\mathbf{v}} = \sqrt{1-d^2}$. For $d \leq 1$, dilation can be called contraction, which is the usual case. For $1 < d$, then $\beta_{\mathbf{v}}$ is an *imaginary* natural speed and it is possible to dilate lengths instead of contract lengths, but dilated lengths are only geometrical effects, not physics effects.

The *hyperbolic versor* $\hat{H}$ is the unit hyperbolic biradial

$$\begin{aligned} \hat{H} &= \gamma_{\mathbf{v}}H = \gamma_{\mathbf{v}}\boldsymbol{v}\boldsymbol{o}^{-1} = \frac{\gamma_{\mathbf{v}}}{c}\boldsymbol{v}\boldsymbol{\gamma}_0 \\ &= \frac{\gamma_{\mathbf{v}}}{c}(\boldsymbol{v}\cdot\boldsymbol{\gamma}_0 + \boldsymbol{v}\wedge\boldsymbol{\gamma}_0) = \gamma_{\mathbf{v}} + \frac{\gamma_{\mathbf{v}}}{c}\mathbf{v}\boldsymbol{\gamma}_0 \\ &= \gamma_{\mathbf{v}} + \gamma_{\mathbf{v}}\frac{\|\mathbf{v}\|}{c}\hat{\mathbf{v}}\boldsymbol{\gamma}_0 = \gamma_{\mathbf{v}} + \gamma_{\mathbf{v}}\beta_{\mathbf{v}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0 \\ &= \cosh(\varphi_{\mathbf{v}}) + \sinh(\varphi_{\mathbf{v}})\hat{\mathbf{v}}\boldsymbol{\gamma}_0 = e^{\varphi_{\mathbf{v}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0} \end{aligned}$$

where

$$\begin{aligned} (\hat{\mathbf{v}}\boldsymbol{\gamma}_0)^2 = (\hat{\mathbf{v}}\wedge\boldsymbol{\gamma}_0)^2 &= 1 \\ \gamma_{\mathbf{v}} &= \cosh(\varphi_{\mathbf{v}}) \\ \gamma_{\mathbf{v}}\beta_{\mathbf{v}} &= \sinh(\varphi_{\mathbf{v}}) \\ \beta_{\mathbf{v}} &= \tanh(\varphi_{\mathbf{v}}) = \frac{\sinh(\varphi_{\mathbf{v}})}{\cosh(\varphi_{\mathbf{v}})} \\ \varphi_{\mathbf{v}} &= \operatorname{atanh}(\beta_{\mathbf{v}}). \end{aligned}$$

Using half the *rapidity* $\frac{1}{2}\varphi_{\mathbf{v}}$, the *boost operator* B is

$$\begin{aligned} B = \hat{H}^{\frac{1}{2}} &= e^{\frac{1}{2}\varphi_{\mathbf{v}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0} \\ &= \cosh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right) + \sinh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right)\hat{\mathbf{v}}\boldsymbol{\gamma}_0 \\ &= \frac{(1+\beta_{\mathbf{v}}) + \sqrt{1-\beta_{\mathbf{v}}^2}}{2\sqrt{1+\beta_{\mathbf{v}}}\sqrt[4]{1-\beta_{\mathbf{v}}^2}} + \frac{(1+\beta_{\mathbf{v}}) - \sqrt{1-\beta_{\mathbf{v}}^2}}{2\sqrt{1+\beta_{\mathbf{v}}}\sqrt[4]{1-\beta_{\mathbf{v}}^2}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0 \\ &\simeq \frac{1}{2}\Big(1+\beta_{\mathbf{v}} + \sqrt{1-\beta_{\mathbf{v}}^2}\Big) + \frac{1}{2}\Big(1+\beta_{\mathbf{v}} - \sqrt{1-\beta_{\mathbf{v}}^2}\Big)\hat{\mathbf{v}}\boldsymbol{\gamma}_0. \end{aligned} \tag{1}$$

This last form (1) of B is valid for $\beta_{\mathbf{v}} = 1$ and may be more numerically stable than the hyperbolic functions for $\beta_{\mathbf{v}} \to 1$, where the hyperbolic functions approach ∞. However, (1) is valid only for $-1 < \beta_{\mathbf{v}} \leq 1$ and dilations using imaginary speeds and rapidities do not work with (1).

The STA *boost operation* $B(\boldsymbol{u}t)\,B^{-1}$ on STA position $\boldsymbol{p}_0 = \boldsymbol{u}t$ relative to observer $\boldsymbol{o}$

$$\boldsymbol{p}_0 = \boldsymbol{u}t \;=\; (\boldsymbol{o} + \mathbf{u})t = (c\boldsymbol{\gamma}_0 + \|\mathbf{u}\|\hat{\mathbf{u}})t$$

is

$$\boldsymbol{p}_0' = (\boldsymbol{u}t)' \;=\; B(\boldsymbol{u}t)B^{-1} = B(\boldsymbol{u}t)\,B^{\sim} = t\,B\boldsymbol{u}B^{\sim} = \boldsymbol{u}'t$$

where $B^{\sim}$ is the *reverse* of B

$$B^{\sim} \;=\; e^{\frac{1}{2}\varphi_{\mathbf{v}}\boldsymbol{\gamma}_0\hat{\mathbf{v}}} = e^{-\frac{1}{2}\varphi_{\mathbf{v}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0} = B^{-1}.$$

The STA *position normalization* of the boosted STA position $\boldsymbol{p}_0'$ is

$$\boldsymbol{p}_0'' \;=\; ct\frac{\boldsymbol{p}_0'}{\boldsymbol{p}_0' \cdot \boldsymbol{\gamma}_0} = c\frac{\boldsymbol{u}'}{\boldsymbol{u}' \cdot \boldsymbol{\gamma}_0}t = \boldsymbol{u}''t = (\boldsymbol{o} + \mathbf{u}'')t.$$

The time t is the *proper time* of the observer $\boldsymbol{o}$. The normalized boosted position $\boldsymbol{p}_0''$ has its initial spacetime velocity $\boldsymbol{u}$ boosted by the spatial boost velocity $\mathbf{v}$ relative to the observer $\boldsymbol{o}$. For boost speed $\|\mathbf{v}\| \ll c$ and initial speed $\|\mathbf{u}\| \ll c$, the boost is approximately an addition of velocities

$$\boldsymbol{u}'' \;\approx\; \boldsymbol{u} + \mathbf{v} = \boldsymbol{o} + \mathbf{u} + \mathbf{v}.$$

For $\mathbf{u} = 0$, or $\boldsymbol{u} = \boldsymbol{o}$, the normalized boosted position $\boldsymbol{p}_0''$ is exactly

$$\boldsymbol{p}_0'' = \boldsymbol{u}''t \;=\; (\boldsymbol{o} + \mathbf{v})t = \boldsymbol{v}t = \boldsymbol{p}_1$$

which is the observer $\boldsymbol{o}t$ boosted into the boost particle $\boldsymbol{p}_1$. The double prime marks can be dropped if the new $\boldsymbol{p}_0$ and its observer $\boldsymbol{o}$ are understood.

In SPECIAL RELATIVITY, the "velocity addition" operation, or formula, is the spatial velocity part $\mathbf{u}''$ of a normalized boost operation $\boldsymbol{p}_0''$. The boost of a velocity $\mathbf{u}$ by a boost velocity $\mathbf{v}$, both relative to an observer $\boldsymbol{o}$, is denoted $\mathbf{u}'' = \mathbf{u} \oplus \mathbf{v}$ and is such that light speed c can never be exceeded, $0 \leq \|\mathbf{u} \oplus \mathbf{v}\| \leq c$. For low speeds, $\mathbf{u} \oplus \mathbf{v} \approx \mathbf{u} + \mathbf{v}$.

The STA *boost operator* $B_{\mathcal{M}}$ for a boost by natural speed $\beta_{\mathbf{v}}$ in the SA unit spatial direction $\hat{\mathbf{v}}_{\mathcal{S}}$ relative to the observer $\boldsymbol{o}_{\mathcal{M}} = c\boldsymbol{\gamma}_0$ is defined as

$$\begin{aligned} B_{\mathcal{M}} &= e^{\frac{1}{2}\varphi_{\mathbf{v}}\hat{\mathbf{v}}_{\mathcal{S}}\boldsymbol{\gamma}_0} \\ &= \cosh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right) + \sinh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right)\hat{\mathbf{v}}_{\mathcal{S}} \wedge \boldsymbol{\gamma}_0 \end{aligned}$$

where the spacetime boost velocity $\boldsymbol{v}_{\mathcal{M}}$ of the boost operator $B_{\mathcal{M}}$ is

$$\boldsymbol{v}_{\mathcal{M}} \;=\; \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}} = c\boldsymbol{\gamma}_0 + \|\mathbf{v}_{\mathcal{S}}\|\hat{\mathbf{v}}_{\mathcal{S}} = c\boldsymbol{\gamma}_0 + \beta_{\mathbf{v}} c\hat{\mathbf{v}}_{\mathcal{S}}$$

and the rapidity $\varphi_{\mathbf{v}}$ of the spatial boost velocity $\mathbf{v}_{\mathcal{S}}$ is

$$\varphi_{\mathbf{v}} \;=\; \operatorname{atanh}(\beta_{\mathbf{v}}) = \operatorname{atanh}\left(\frac{\|\mathbf{v}_{\mathcal{S}}\|}{c}\right).$$

The STA *boost operation* with *position normalization* on an STA position $\boldsymbol{p}_{\mathcal{M}_0} = \boldsymbol{u}_{\mathcal{M}} t$ is

$$\boldsymbol{p}'_{\mathcal{M}_0} \;=\; ct\frac{B_{\mathcal{M}}\boldsymbol{p}_{\mathcal{M}_0}B_{\mathcal{M}}^{\sim}}{(B_{\mathcal{M}}\boldsymbol{p}_{\mathcal{M}_0}B_{\mathcal{M}}^{\sim})\cdot\boldsymbol{\gamma}_0}.$$

The *velocity normalization* uses factor c instead of ct in the above boost operation.

The *particle* of the STA boost is

$$\boldsymbol{p}_{\mathcal{M}_1} \;=\; \boldsymbol{v}_{\mathcal{M}} t$$

which is the particle that results by boosting the observer position $\boldsymbol{p}_{\mathcal{M}_0} = \boldsymbol{o}_{\mathcal{M}} t$ and normalizing to obtain the transformation $\boldsymbol{p}_{\mathcal{M}_0} \to \boldsymbol{p}_{\mathcal{M}_1}$.

Using the hyperbolic function composition identities

$$\begin{aligned}
\cosh\left(\frac{1}{2}\operatorname{atanh}(\beta_{\mathbf{v}})\right) &= \frac{(1+\beta_{\mathbf{v}}) + \sqrt{1-\beta_{\mathbf{v}}^2}}{2\sqrt{1+\beta_{\mathbf{v}}}\sqrt[4]{1-\beta_{\mathbf{v}}^2}} \quad \text{for } -1 < \beta_{\mathbf{v}} < 1 \\
\sinh\left(\frac{1}{2}\operatorname{atanh}(\beta_{\mathbf{v}})\right) &= \frac{(1+\beta_{\mathbf{v}}) - \sqrt{1-\beta_{\mathbf{v}}^2}}{2\sqrt{1+\beta_{\mathbf{v}}}\sqrt[4]{1-\beta_{\mathbf{v}}^2}} \quad \text{for } -1 < \beta_{\mathbf{v}} < 1 \\
\cosh(\operatorname{atanh}(\beta_{\mathbf{v}})) &= \frac{1}{\sqrt{1-\beta_{\mathbf{v}}^2}} \quad \text{for } -1 < \beta_{\mathbf{v}} < 1 \\
\sinh(\operatorname{atanh}(\beta_{\mathbf{v}})) &= \frac{\beta_{\mathbf{v}}}{\sqrt{1-\beta_{\mathbf{v}}^2}} \quad \text{for } -1 < \beta_{\mathbf{v}} < 1, \\
\cosh(2\operatorname{atanh}(\beta_{\mathbf{v}})) &= \frac{1+\beta_{\mathbf{v}}^2}{1-\beta_{\mathbf{v}}^2} \quad \text{for } -1 < \beta_{\mathbf{v}} < 1 \\
\sinh(2\operatorname{atanh}(\beta_{\mathbf{v}})) &= \frac{2\beta_{\mathbf{v}}}{1-\beta_{\mathbf{v}}^2} \quad \text{for } -1 < \beta_{\mathbf{v}} < 1
\end{aligned}$$

it can be shown that the boost operator that applies the boost $\beta_{\mathbf{v}}$ *twice* successively, and adds the rapidity $2\varphi_{\mathbf{v}}$, is

$$B_{\mathcal{M}}B_{\mathcal{M}} = B_{\beta_{\mathbf{v}}}B_{\beta_{\mathbf{v}}} \;=\; \frac{1}{\sqrt{1-\beta_{\mathbf{v}}^2}} + \frac{\beta_{\mathbf{v}}}{\sqrt{1-\beta_{\mathbf{v}}^2}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0$$

and the boost operator that applies the boost $\frac{1}{2}\beta_{\mathbf{v}}$ *twice* successively is

$$B_{\frac{1}{2}\beta_{\mathbf{v}}}B_{\frac{1}{2}\beta_{\mathbf{v}}} \;=\; \frac{2}{\sqrt{1-\beta_{\mathbf{v}}^2}} + \frac{\beta_{\mathbf{v}}}{\sqrt{1-\beta_{\mathbf{v}}^2}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0.$$

The boost operation that applies boosts by $\frac{1}{2}\beta_{\mathbf{v}}$ *twice* successively

$$(\boldsymbol{u}t)' \;\approx\; \left(B_{\frac{1}{2}\beta_{\mathbf{v}}}B_{\frac{1}{2}\beta_{\mathbf{v}}}\right)\boldsymbol{u}t\left(B_{\frac{1}{2}\beta_{\mathbf{v}}}B_{\frac{1}{2}\beta_{\mathbf{v}}}\right)^{\sim}$$

is *approximately equal*, for small $\beta_{\mathbf{v}} \ll c$, to the single boost by $\beta_{\mathbf{v}}$

$$(\boldsymbol{u}t)' \;=\; B_{\beta_{\mathbf{v}}}(\boldsymbol{u}t)B_{\beta_{\mathbf{v}}}^{\sim}.$$

The double boost operator $B_{\beta_\mathbf{v}}B_{\beta_\mathbf{v}}$ can be defined as successive reflections in two hyperbolic spacetime planes, where the first plane contains the observer and the second plane contains the boost particle. The two planes bound the hyperbolic angle $\varphi_\mathbf{v}$ that turns from the first plane at $\beta_{\boldsymbol{o}}=0$ into the second plane at $\beta_\mathbf{v}$, toward the direction in space of the boost velocity, or spatial axis, $\mathbf{v}$.

For very small $\beta_\mathbf{v}\ll c$, then $\varphi_\mathbf{v}=\operatorname{atanh}(\beta_\mathbf{v})\approx\beta_\mathbf{v}$ and then

$$\begin{aligned} B_\mathcal{M} &= e^{\frac{1}{2}\varphi_\mathbf{v}\hat{\mathbf{v}}\boldsymbol{\gamma}_0}=e^{\frac{1}{4}\varphi_\mathbf{v}\hat{\mathbf{v}}\boldsymbol{\gamma}_0}e^{\frac{1}{4}\varphi_\mathbf{v}\hat{\mathbf{v}}\boldsymbol{\gamma}_0}=B_{\frac{1}{2}\varphi_\mathbf{v}}B_{\frac{1}{2}\varphi_\mathbf{v}} \\ &\approx \frac{2}{\sqrt{1-\beta_\mathbf{v}^2}}+\frac{\beta_\mathbf{v}}{\sqrt{1-\beta_\mathbf{v}^2}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0=B_{\frac{1}{2}\beta_\mathbf{v}}B_{\frac{1}{2}\beta_\mathbf{v}} \\ &\approx e^{\frac{1}{2}\beta_\mathbf{v}\hat{\mathbf{v}}\boldsymbol{\gamma}_0}=\cosh\left(\frac{1}{2}\beta_\mathbf{v}\right)+\sinh\left(\frac{1}{2}\beta_\mathbf{v}\right)\hat{\mathbf{v}}\boldsymbol{\gamma}_0 \end{aligned}$$

The good approximation of $B_\mathcal{M}$ for very small $\beta_\mathbf{v}\ll c$ is

$$B_\mathcal{M} \approx \frac{2}{\sqrt{1-\beta_\mathbf{v}^2}}+\frac{\beta_\mathbf{v}}{\sqrt{1-\beta_\mathbf{v}^2}}\hat{\mathbf{v}}\boldsymbol{\gamma}_0.$$

Since the magnitude of the versor $B_\mathcal{M}$ is not important, it can be rescaled as

$$B_\mathcal{M}\approx B_\approx = 1+\frac{1}{2}\beta_\mathbf{v}\hat{\mathbf{v}}\boldsymbol{\gamma}_0=1+\frac{1}{2c}\mathbf{v}\boldsymbol{\gamma}_0.$$

The approximate boost operator $B_\approx$ continues to hold light speed c as the limit of successive boosts and has good accuracy for a wide range of low speeds.

3.2.6 STA reframe (reverse boost)

A reverse boost is a change in spacetime observer frame, or reframe, which is similar to a change of spatial basis using a reverse spatial rotation. If $\boldsymbol{p}_{\mathcal{M}_0}$ is a position relative to observer velocity $\boldsymbol{o}_\mathcal{M}$, then

$$\boldsymbol{p}'_{\mathcal{M}_0} = B_\mathcal{M}^{-1}\boldsymbol{p}_{\mathcal{M}_0}B_\mathcal{M}=B_\mathcal{M}^{\sim}\boldsymbol{p}_{\mathcal{M}_0}B_\mathcal{M}$$

is $\boldsymbol{p}_{\mathcal{M}_0}$ relative to the *new observer* particle $\boldsymbol{p}_{\mathcal{M}_1}=\boldsymbol{v}_\mathcal{M}t$ of the boost $B_\mathcal{M}$. The reverse boost turns $\boldsymbol{v}_\mathcal{M}$ into the new observer $\boldsymbol{o}'_\mathcal{M}$. Relative to the new observer, the *normalization* of $\boldsymbol{p}'_{\mathcal{M}_0}$ to a *new* $\boldsymbol{p}_{\mathcal{M}_0}$ (dropping primes) is

$$\boldsymbol{p}_{\mathcal{M}_0} = ct\frac{\boldsymbol{p}'_{\mathcal{M}_0}}{\boldsymbol{p}'_{\mathcal{M}_0}\cdot\boldsymbol{\gamma}_0}$$

such that the observer always has the normalized position $\boldsymbol{o}_\mathcal{M}t=ct\boldsymbol{\gamma}_0$ at its proper time t. The observer rests at the spatial origin of the frame.

For example, if the *old* position is $\boldsymbol{p}_{\mathcal{M}_0}=\boldsymbol{o}_\mathcal{M}t$ and the boost spacetime velocity of $B_\mathcal{M}$ is $\boldsymbol{v}_\mathcal{M}=\boldsymbol{o}_\mathcal{M}+\mathbf{v}_\mathcal{S}$, which is the new observer, then the *old*$\to$*new* position is $\boldsymbol{o}_\mathcal{M}t\to(\boldsymbol{o}_\mathcal{M}-\mathbf{v}_\mathcal{S})t$ relative to *old*$\to$*new* observer $\boldsymbol{v}_\mathcal{M}\to\boldsymbol{o}_\mathcal{M}$ after normalization and dropping prime marks. In this example, the old observer $\boldsymbol{o}_\mathcal{M}$ and particle $\boldsymbol{v}_\mathcal{M}$ exchange roles, such that the old particle becomes the new observer and the old observer becomes the new particle.

The STA *boost operator* $B_\mathcal{M}$ for a boost by natural speed $\beta_\mathbf{v}$ in the unit spatial direction $\hat{\mathbf{v}}_\mathcal{S}$ relative to the observer $\boldsymbol{o}_\mathcal{M}=c\boldsymbol{\gamma}_0$ is defined as

$$\begin{aligned} B_\mathcal{M} &= e^{\frac{1}{2}\varphi_\mathbf{v}\hat{\mathbf{v}}_\mathcal{S}\boldsymbol{\gamma}_0} \\ &= \cosh\left(\frac{1}{2}\varphi_\mathbf{v}\right)+\sinh\left(\frac{1}{2}\varphi_\mathbf{v}\right)\hat{\mathbf{v}}_\mathcal{S}\wedge\boldsymbol{\gamma}_0 \end{aligned}$$

where the *new observer* spacetime velocity $\boldsymbol{v}_{\mathcal{M}} \to \boldsymbol{o}_{\mathcal{M}}$ of the reframe boost operator $B_{\mathcal{M}}$ is

$$\boldsymbol{v}_{\mathcal{M}} \;=\; \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}} = c\boldsymbol{\gamma}_0 + \|\mathbf{v}_{\mathcal{S}}\|\hat{\mathbf{v}}_{\mathcal{S}} = c\boldsymbol{\gamma}_0 + \beta_{\mathbf{v}} c\hat{\mathbf{v}}_{\mathcal{S}}$$

and the rapidity $\varphi_{\mathbf{v}}$ of the spatial boost velocity $\mathbf{v}_{\mathcal{S}}$ is

$$\varphi_{\mathbf{v}} \;=\; \operatorname{atanh}(\beta_{\mathbf{v}}) = \operatorname{atanh}\left(\frac{\|\mathbf{v}_{\mathcal{S}}\|}{c}\right).$$

The STA *reframe boost operation* (reverse boost) with *position normalization* on an STA position $\boldsymbol{p}_{\mathcal{M}_0} = \boldsymbol{u}_{\mathcal{M}} t$ is

$$\boldsymbol{p}'_{\mathcal{M}_0} \;=\; ct\frac{B\widetilde{\mathcal{M}}\boldsymbol{p}_{\mathcal{M}_0}B_{\mathcal{M}}}{(B\widetilde{\mathcal{M}}\boldsymbol{p}_{\mathcal{M}_0}B_{\mathcal{M}})\cdot\boldsymbol{\gamma}_0}.$$

The *velocity normalization* uses factor c instead of ct in the above boost operation. This operation is just the boost operation in reverse, or is an inverse boost operation. The time t is the *proper time* of the new observer $(\boldsymbol{p}_{\mathcal{M}_1} = \boldsymbol{v}_{\mathcal{M}} t) \to \boldsymbol{o}_{\mathcal{M}} t$.

4 Conformal Space-Time Algebra (CSTA)

$\mathcal{G}_{2,4}$ Conformal Space-Time Algebra (CSTA) is introduced in [2] as the *spacetime conformal group*.

$\mathcal{G}_{2,4}$ CSTA is a straightforward extension and adaptation of the $\mathcal{G}_{4,1}$ Conformal Geometric Algebra (CGA). CGA is introduced by Hestenes, Li, and Rockwood in [14]. CGA is also discussed by Perwass in [11], and by Dorst, Fontijne, and Mann in [3].

$\mathcal{G}_{4,8}$ Double Conformal Space-Time Algebra (DCSTA) contains two copies of $\mathcal{G}_{2,4}$ CSTA, which are called CSTA1 and CSTA2. Elements and operations in CSTA1 are subscripted with $\mathcal{C}^1$. Elements and operations in CSTA2 are subscripted with $\mathcal{C}^2$. Elements and operations in generic CSTA are subscripted with $\mathcal{C}$.

Most formulas are expressed in CSTA and written explicitly in CSTA1 and CSTA2 only when helpful to see how the particular CSTA1 and CSTA2 elements are used in formulas. Most formulas in CSTA can be written in CSTA1 and CSTA2 by just changing subscripts. CSTA uses the origin $\mathbf{e}_{o\gamma}$ and infinity $\mathbf{e}_{\infty\gamma}$ points and the Dirac gammas $\boldsymbol{\gamma}_0$, $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$, $\boldsymbol{\gamma}_3$ for the timelike $w = ct$ and spatial x, y, and z axes, respectively. The generic CSTA point embedding is $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$.

4.1 CSTA unit pseudoscalar

The $\mathcal{G}_{2,4}$ CSTA 6-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{C}}$ with signature $(+---+-)$ is

$$\begin{aligned}
\mathbf{I}_{\mathcal{C}} &= \mathbf{I}_{\mathcal{M}}\mathbf{e}_{\infty\gamma}\mathbf{e}_{o\gamma} = \boldsymbol{\gamma}_0\mathbf{I}_{\mathcal{S}}\mathbf{e}_{\infty\gamma}\mathbf{e}_{o\gamma} = \boldsymbol{\gamma}_0\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3\mathbf{e}_{\infty\gamma}\mathbf{e}_{o\gamma} = \boldsymbol{\gamma}_0\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3\mathbf{e}_+\mathbf{e}_- \\
\mathbf{I}_{\mathcal{C}}\widetilde{} &= (-1)^{6(6-1)/2}\mathbf{I}_{\mathcal{C}} = -\mathbf{I}_{\mathcal{C}} \\
\mathbf{I}_{\mathcal{C}}^2 &= -1 \\
\mathbf{I}_{\mathcal{C}}^{-1} &= -\mathbf{I}_{\mathcal{C}} = \mathbf{I}_{\mathcal{C}}\widetilde{}.
\end{aligned}$$

The $\mathcal{G}_{2,4}$ CSTA1 6-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{C}^1}$ with signature $(+---+-)$ is

$$\mathbf{I}_{\mathcal{C}^1} \;=\; \mathbf{I}_{\mathcal{M}^1}\mathbf{e}_{\infty 1}\mathbf{e}_{o1} = \mathbf{e}_1\mathbf{I}_{\mathcal{S}^1}\mathbf{e}_{\infty 1}\mathbf{e}_{o1} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4\mathbf{e}_{\infty 1}\mathbf{e}_{o1} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4\mathbf{e}_5\mathbf{e}_6.$$

The $\mathcal{G}_{2,4}$ CSTA2 6-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{C}^2}$ with signature $(+---+-)$ is

$$\mathbf{I}_{\mathcal{C}^2} = \mathbf{I}_{\mathcal{M}^2}\mathbf{e}_{\infty 2}\mathbf{e}_{o2} = \mathbf{e}_7\mathbf{I}_{\mathcal{S}^2}\mathbf{e}_{\infty 2}\mathbf{e}_{o2} = \mathbf{e}_7\mathbf{e}_8\mathbf{e}_9\mathbf{e}_{10}\mathbf{e}_{\infty 2}\mathbf{e}_{o2} = \mathbf{e}_7\mathbf{e}_8\mathbf{e}_9\mathbf{e}_{10}\mathbf{e}_{11}\mathbf{e}_{12}.$$

4.2 CSTA point

The CSTA null 1-vector *point* entity is very similar to the CGA null 1-vector *point* entity. The following subsections define the CSTA points at the origin and at infinity, and the CSTA point embedding.

4.2.1 Stereographic embedding and homogenization

The embedding of an $\mathcal{G}_{1,3}$ STA position *vector* $\boldsymbol{p}_{\mathcal{M}}$ into a $\mathcal{G}_{2,4}$ CSTA null 1-vector *point* $\boldsymbol{P}_{\mathcal{C}}$ is done in exactly the same way a $\mathcal{G}_3$ APS point $\mathbf{p}$ is embedded into a $\mathcal{G}_{4,1}$ CGA point $\mathbf{P}_{\mathcal{C}}$. There are many references that explain the stereographic embedding and homogenization, such as [11] and the paper on $\mathcal{G}_{8,2}$ DCGA [6].

4.2.2 CSTA point at the origin

The CSTA null 1-vector *point at the origin* is defined as

$$\mathbf{e}_{o\gamma} = \frac{1}{2}(-\mathbf{e}_+ + \mathbf{e}_-)$$

where $\mathbf{e}_+$ is the stereographic unit and $\mathbf{e}_-$ is the homogeneous unit, and

$$\mathbf{e}_+ = \begin{cases} \mathbf{e}_5 & : \text{ in CSTA1} \\ \mathbf{e}_{11} & : \text{ in CSTA2} \end{cases}$$
$$\mathbf{e}_- = \begin{cases} \mathbf{e}_6 & : \text{ in CSTA1} \\ \mathbf{e}_{12} & : \text{ in CSTA2.} \end{cases}$$

The CSTA1 null 1-vector *point at the origin* is defined as

$$\mathbf{e}_{o1} = \frac{1}{2}(-\mathbf{e}_5 + \mathbf{e}_6).$$

The CSTA2 null 1-vector *point at the origin* is defined as

$$\mathbf{e}_{o2} = \frac{1}{2}(-\mathbf{e}_{11} + \mathbf{e}_{12}).$$

The CSTA null 1-vector *point at the origin* $\mathbf{e}_{o\gamma}$ represents either $\mathbf{e}_{o1}$ or $\mathbf{e}_{o2}$.

4.2.3 CSTA point at infinity

The CSTA null 1-vector *point at infinity* is defined as

$$\mathbf{e}_{\infty\gamma} = \mathbf{e}_+ + \mathbf{e}_-.$$

The CSTA1 null 1-vector *point at infinity* is defined as

$$\mathbf{e}_{\infty 1} = \mathbf{e}_5 + \mathbf{e}_6.$$

The CSTA2 null 1-vector *point at infinity* is defined as

$$\mathbf{e}_{\infty 2} = \mathbf{e}_{11} + \mathbf{e}_{12}.$$

The CSTA null 1-vector *point at infinity* $\mathbf{e}_{\infty\gamma}$ represents either $\mathbf{e}_{\infty 1}$ or $\mathbf{e}_{\infty 2}$.

4.2.4 CSTA point embedding

The generic CSTA null 1-vector *point* $\boldsymbol{P}_{\mathcal{C}}$ entity is the embedding of an STA *position* $\boldsymbol{p}_{\mathcal{M}}$ as

$$\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}}) \;=\; \boldsymbol{p}_{\mathcal{M}} + \frac{1}{2}\boldsymbol{p}_{\mathcal{M}}^2\mathbf{e}_{\infty\gamma} + \mathbf{e}_{o\gamma}.$$

The CSTA1 null 1-vector *point* $\boldsymbol{P}_{\mathcal{C}^1}$ entity is the embedding of an STA1 *position* $\boldsymbol{p}_{\mathcal{M}^1}$ as

$$\boldsymbol{P}_{\mathcal{C}^1} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}^1}) \;=\; \boldsymbol{p}_{\mathcal{M}^1} + \frac{1}{2}\boldsymbol{p}_{\mathcal{M}^1}^2\mathbf{e}_{\infty 1} + \mathbf{e}_{o1}.$$

The CSTA2 null 1-vector *point* $\boldsymbol{P}_{\mathcal{C}^2}$ entity is the embedding of an STA2 *position* $\boldsymbol{p}_{\mathcal{M}^2}$ as

$$\boldsymbol{P}_{\mathcal{C}^2} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}^2}) \;=\; \boldsymbol{p}_{\mathcal{M}^2} + \frac{1}{2}\boldsymbol{p}_{\mathcal{M}^2}^2\mathbf{e}_{\infty 2} + \mathbf{e}_{o2}.$$

The embedding function $\mathcal{C}$ is implemented as a piecewise embedding function that embeds an STA, STA1, or STA2 vector into the corresponding CSTA, CSTA1, or CSTA2 point. The generic CSTA embedding will used to avoid duplication in generic discussions that can apply just as well in either CSTA1 or CSTA2 by only changing the subscripts accordingly.

The CSTA point $\boldsymbol{P}_{\mathcal{C}}$ is similar to a CGA point $\mathbf{P}_{\mathcal{C}}$ as in [6] *when* $\boldsymbol{P}_{\mathcal{C}}$ is the embedding of a spatial point $\boldsymbol{p}_{\mathcal{M}} = \mathbf{p}_{\mathcal{S}}$ and we hold $w = ct = 0$.

As a GOPNS entity (§4.5.2), a CSTA point $\boldsymbol{P}_{\mathcal{C}}$ simply represents the *point*, as expected.

As a GIPNS entity, a finite CSTA point $\boldsymbol{P}_{\mathcal{C}}$, excluding $\mathbf{e}_{\infty\gamma}$, actually represents a *hypercone* in spacetime of the form

$$(w - p_w)^2 - (x - p_x)^2 - (y - p_y)^2 - (z - p_z)^2 \;=\; 0$$

where

$$\boldsymbol{p}_{\mathcal{M}} \;=\; p_w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3 = p_w\boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}}.$$

In general, a *hyper*surface in an n-D space has an $(n-1)$-D surface. A cone or other surface in 3-D space has a 2-D surface, but a hypercone or other hypersurface in 4-D spacetime has a 3-D surface. A hypersurface is treated and conceptualized in most respects the same as a 2-D surface, but it embeds extended dimensions and its mathematical forms contain an additional term per extended dimension.

The hypercone is a result of the Minkowski spacetime metric (1, 3), which can be seen in the hypercone equation. For comparison to $\mathcal{G}_{4,1}$ CGA, a CGA point embeds a 3-D Euclidean vector with metric (3, 0) and represents an implicit surface equation of a sphere with zero radius

$$(x - p_x)^2 + (y - p_y)^2 + (z - p_z)^2 \;=\; 0.$$

In 3-D spacetime with only two spatial dimensions by holding $z - p_z = 0$, the hypercone reduces to the circular cone

$$(x - p_x)^2 + (y - p_y)^2 - (w - p_w)^2 \;=\; 0$$

which is an expanding circle in the xy-plane as the time-like coordinate $w = ct$ increases past p_w. The hypercone is an expanding sphere in space that is expanding with time t in radius

$$r \;=\; w - p_w = ct - p_w$$

at the speed of light c. The point begins expanding after time $t = p_w / c$ and is contracting before that time.

A CSTA point, as an expanding sphere, represents a *light-cone* in spacetime that is centered at the vertex point $\boldsymbol{p}_{\mathcal{M}}$. In spacetime, the light-cone is a spherical hypercone, which is a cone with a 3-D hypersurface. A surface is usually 2-D, but a hypersurface is imagined as a surface while it is actually a higher-dimensional space. The light-cone is often depicted as a *cone* in a 3-D spacetime of two spatial dimensions and a time-like dimension, wherein the cone is a *circular wave front* of light that expands in space as time t increases. The expanding radius $r = ct - p_w$ of the wave front is centered at a point light source $\boldsymbol{p}_{\mathcal{M}}$. A CSTA point represents a *spherical wave front* of light in space, or light-cone in spacetime, centered at a point light source $\boldsymbol{p}_{\mathcal{M}}$ that flashes at time $t = p_w / c$.

4.2.5 CSTA point normalization

A homogeneous CSTA point embedding with scalar weight s is

$$s\boldsymbol{P}_{\mathcal{C}} = s\mathcal{C}(\boldsymbol{p}_{\mathcal{M}}) \;=\; s\boldsymbol{p}_{\mathcal{M}} + s\frac{1}{2}\boldsymbol{p}_{\mathcal{M}}^2\mathbf{e}_{\infty\gamma} + s\mathbf{e}_{o\gamma}.$$

A *normalized* point is scaled to weight $s = 1$.

The *normalization* of a weighted CSTA point $s\boldsymbol{P}_{\mathcal{C}}$ is

$$\boldsymbol{P}_{\mathcal{C}} \;=\; \frac{(s\boldsymbol{P}_{\mathcal{C}})}{-(s\boldsymbol{P}_{\mathcal{C}})\cdot\mathbf{e}_{\infty\gamma}} = \frac{s\boldsymbol{P}_{\mathcal{C}}}{s}.$$

Many formulas require points and other entities to be unit weight. The normalization of an entity can be particular to the type of the entity.

4.2.6 CSTA point projection (inverse embedding)

The projection of CSTA point $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ to STA position $\boldsymbol{p}_{\mathcal{M}}$ is

$$\boldsymbol{p}_{\mathcal{M}} = \mathcal{C}^{-1}(\boldsymbol{P}_{\mathcal{C}}) \;=\; \left(\frac{\boldsymbol{P}_{\mathcal{C}}}{-\boldsymbol{P}_{\mathcal{C}}\cdot\mathbf{e}_{\infty\gamma}}\cdot\mathbf{I}_{\mathcal{M}}\right)\mathbf{I}_{\mathcal{M}}^{-1}.$$

If $\boldsymbol{P}_{\mathcal{C}}$ is the embedding of an STA position, then an STA *position normalization* (§3.2.2) of $\boldsymbol{p}_{\mathcal{M}}$ may be required. If $\boldsymbol{P}_{\mathcal{C}}$ is the embedding of an STA velocity, then an STA *velocity normalization* (§3.2.1) of $\boldsymbol{p}_{\mathcal{M}}$ may be required.

4.2.7 CSTA test point

The symbolic CSTA *test point* $\boldsymbol{T}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{t}_{\mathcal{M}})$ is the embedding of the symbolic STA *test vector*

$$\boldsymbol{t}_{\mathcal{M}} \;=\; w\boldsymbol{\gamma}_0 + x\boldsymbol{\gamma}_1 + y\boldsymbol{\gamma}_2 + z\boldsymbol{\gamma}_3 = ct\boldsymbol{\gamma}_0 + \mathbf{t}_{\mathcal{S}} = \boldsymbol{o}_{\mathcal{M}}t + \mathbf{t}_{\mathcal{S}}.$$

The symbolic CSTA1 *test point* $\boldsymbol{T}_{\mathcal{C}^1} = \mathcal{C}(\boldsymbol{t}_{\mathcal{M}^1})$ is the embedding of the symbolic STA1 *test vector*

$$\boldsymbol{t}_{\mathcal{M}^1} \;=\; w\mathbf{e}_1 + x\mathbf{e}_2 + y\mathbf{e}_3 + z\mathbf{e}_4 = ct\mathbf{e}_1 + \mathbf{t}_{\mathcal{S}^1} = \boldsymbol{o}_{\mathcal{M}^1}t + \mathbf{t}_{\mathcal{S}^1}.$$

The symbolic CSTA2 *test point* $\boldsymbol{T}_{\mathcal{C}^2} = \mathcal{C}(\boldsymbol{t}_{\mathcal{M}^2})$ is the embedding of the symbolic STA2 *test vector*

$$\boldsymbol{t}_{\mathcal{M}^2} \;=\; w\mathbf{e}_7 + x\mathbf{e}_8 + y\mathbf{e}_9 + z\mathbf{e}_{10} = ct\mathbf{e}_7 + \mathbf{t}_{\mathcal{S}^2} = \boldsymbol{o}_{\mathcal{M}^2}t + \mathbf{t}_{\mathcal{S}^2}.$$

The symbolic scalars w, x, y, and z are the conventional coordinates in spacetime. The time-like coordinate $w = ct$ is the coordinate position of light at light speed c at time t. The *observer*, as defined in the theory of special relativity, is identified as the symbolic time-like velocity $\boldsymbol{o}_{\mathcal{M}}$.

CSTA1 and CSTA2 test points $\boldsymbol{T}_{\mathcal{C}^1}$ and $\boldsymbol{T}_{\mathcal{C}^2}$, respectively, are wedged to form the $\mathcal{G}_{4,8}$ DCSTA *test point* $\boldsymbol{T}_{\mathcal{D}} = \boldsymbol{T}_{\mathcal{C}^1} \wedge \boldsymbol{T}_{\mathcal{C}^2}$. The DCSTA point value-*extraction elements* T_s are defined as elements that extract values from the DCSTA test point $\boldsymbol{T}_{\mathcal{D}}$ as $s = \boldsymbol{T}_{\mathcal{D}} \cdot T_s$.

4.3 CSTA point value-extraction elements

The CSTA point value-*extraction elements* C_s extract the value s from a test point $\boldsymbol{T}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{t}_{\mathcal{M}})$ as $s = \boldsymbol{T}_{\mathcal{C}} \cdot C_s$. The CSTA value-extraction elements are

$$\begin{aligned} C_1 &= -\mathbf{e}_{\infty\gamma} \\ C_w &= \boldsymbol{\gamma}_0 \\ C_t &= \frac{1}{c} C_w \\ C_x &= -\boldsymbol{\gamma}_1 \\ C_y &= -\boldsymbol{\gamma}_2 \\ C_z &= -\boldsymbol{\gamma}_3 \\ C_{\boldsymbol{t}^2} &= -2\mathbf{e}_{o\gamma}. \end{aligned}$$

These elements are straightforward to verify. When $w = ct$, the extraction C_t gives t. The extraction

$$\boldsymbol{T}_{\mathcal{C}} \cdot C_{\boldsymbol{t}^2} = \boldsymbol{t}_{\mathcal{M}}^2 = |\boldsymbol{t}_{\mathcal{M}}|^2 = w^2 - r^2 = (ct)^2 - x^2 - y^2 - z^2$$

is the squared modulus of the STA test vector $\boldsymbol{t}_{\mathcal{M}}$.

The CSTA geometric inner product null space (GIPNS) 1-vector surface entities can be defined in terms of these extraction elements by writing their implicit surface functions. Two of these entities are the CSTA GIPNS 1-vector *hyperplane* $\boldsymbol{E}_{\mathcal{C}}$ and the CSTA GIPNS 1-vector *hyperhyperboloid of one sheet (hyperpseudosphere)* $\boldsymbol{\Sigma}_{\mathcal{C}}$. A hyperhyperboloid can degenerate into a hypercone, which is a CSTA GIPNS null 1-vector point entity $\boldsymbol{P}_{\mathcal{C}}$. The CSTA GIPNS 1-vector entities $\boldsymbol{\Sigma}_{\mathcal{C}}$ and $\boldsymbol{E}_{\mathcal{C}}$ are similar to the CGA sphere $\mathbf{S}$ and plane $\boldsymbol{\Pi}$. The other CSTA GIPNS entities are of grades 2 to 5 and are formed as intersections (wedges) of hyperpseudospheres and hyperplanes or by specific formulas.

4.4 CSTA GIPNS entities

The $\mathcal{G}_{2,4}$ CSTA GIPNS entities are similar to $\mathcal{G}_{4,1}$ CGA GIPNS entities, but with some changes to account for the anti-Euclidean signature (0, 3) of $\mathcal{G}_{0,3}$ SA and the pseudo-Euclidean, or Minkowski spacetime, signature (1, 3) of $\mathcal{G}_{1,3}$ STA in a 4-D spacetime. The CSTA GIPNS entities of forms similar to CGA GIPNS entities are representing hypersurfaces in 4-D spacetime.

4.4.1 Geometric inner product null space (GIPNS)

Geometric inner product null space (GIPNS) entities are introduced by Perwass in [11], and are reviewed by this author in [6] and [7].

4.4.2 CSTA GIPNS 1-vector hypercone

The implicit quadric surface equation for a circular *hypercone* is

$$(w-p_w)^2-(x-p_x)^2-(y-p_y)^2-(z-p_z)^2 \;=\; 0.$$

The CSTA GIPNS null 1-vector *hypercone* $\boldsymbol{K}_{\mathcal{C}}$ is the point embedding

$$\boldsymbol{K}_{\mathcal{C}} \;=\; \boldsymbol{P}_{\mathcal{C}}=\mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$$

with center vertex point $\boldsymbol{p}_{\mathcal{M}}$. The hypercone is a sphere in space that expands from a point at $\boldsymbol{p}_{\mathcal{M}}$ with squared radius

$$r^2 \;=\; (w-p_w)^2=(ct-p_w)^2.$$

4.4.3 CSTA GIPNS 1-vector hyperplane

A hyperplane is a linear subspace of dimension $(n-1)$ in a space of dimension n. In 4-D spacetime, a hyperplane is a 3-D subspace. The hyperplane space can be a MINKOWSKI spacetime $(1,2)$ or an anti-Euclidean space $(0,3)$.

An implicit surface equation for a hyperplane in spacetime through the origin can be written

$$\begin{aligned}\boldsymbol{t}_{\mathcal{M}}\cdot\boldsymbol{n}_{\mathcal{M}} \;&=\; \\ n_w w-n_x x-n_y y-n_z z \;&=\; 0.\end{aligned}$$

The STA vector

$$\boldsymbol{n}_{\mathcal{M}} \;=\; n_w\boldsymbol{\gamma}_0+n_x\boldsymbol{\gamma}_1+n_y\boldsymbol{\gamma}_2+n_z\boldsymbol{\gamma}_3$$

is the *normal vector* to the hyperplane. Only the direction of $\boldsymbol{n}_{\mathcal{M}}$ is significant, and its magnitude can be arbitrary. The STA test vector $\boldsymbol{t}_{\mathcal{M}}$ is

$$\boldsymbol{t}_{\mathcal{M}} \;=\; w\boldsymbol{\gamma}_0+x\boldsymbol{\gamma}_1+y\boldsymbol{\gamma}_2+z\boldsymbol{\gamma}_3.$$

The equation holds good for any point $\boldsymbol{t}_{\mathcal{M}}$ on the hyperplane through the origin orthogonal to $\boldsymbol{n}_{\mathcal{M}}$. Using the CSTA point value-extraction elements (§4.3), the hyperplane implicit surface function can be written as the CSTA GIPNS entity

$$\begin{aligned}n_wC_w-n_xC_x-n_yC_y-n_zC_z \;&=\; \\ n_w\boldsymbol{\gamma}_0+n_x\boldsymbol{\gamma}_1+n_y\boldsymbol{\gamma}_2+n_z\boldsymbol{\gamma}_3 \;&=\; \\ \boldsymbol{n}_{\mathcal{M}} \;&.\end{aligned}$$

The CSTA GIPNS 1-vector *hyperplane* $\boldsymbol{E}_{\mathcal{C}}$ through the origin with normal vector $\boldsymbol{n}_{\mathcal{M}}$ is defined as

$$\boldsymbol{E}_{\mathcal{C}} \;=\; \boldsymbol{n}_{\mathcal{M}}.$$

The hyperplane through the origin $\boldsymbol{n}_{\mathcal{M}}$ can be translated from the origin to a point $\boldsymbol{d}_{\mathcal{M}}$ using the translator (§4.6.4) operation

$$\begin{aligned}T_{\mathcal{C}}\boldsymbol{n}_{\mathcal{M}}\widetilde{T_{\mathcal{C}}} \;&=\; \\ \left(1-\frac{1}{2}\boldsymbol{d}_{\mathcal{M}}\mathbf{e}_{\infty\gamma}\right)\boldsymbol{n}_{\mathcal{M}}\left(1-\frac{1}{2}\mathbf{e}_{\infty\gamma}\boldsymbol{d}_{\mathcal{M}}\right) \;&=\; \\ \boldsymbol{n}_{\mathcal{M}}+(\boldsymbol{d}_{\mathcal{M}}\cdot\boldsymbol{n}_{\mathcal{M}})\mathbf{e}_{\infty\gamma} \;&.\end{aligned}$$

The CSTA GIPNS 1-vector *hyperplane* $\boldsymbol{E}_{\mathcal{C}}$ through the point $\boldsymbol{p}_{\mathcal{M}}$ with normal vector $\boldsymbol{n}_{\mathcal{M}}$ is defined as

$$\begin{aligned}\boldsymbol{E}_{\mathcal{C}} &= \boldsymbol{n}_{\mathcal{M}} + (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{n}_{\mathcal{M}})\mathbf{e}_{\infty\gamma} \\ &= \boldsymbol{n}_{\mathcal{M}} + d^2\mathbf{e}_{\infty\gamma} \\ &\simeq \boldsymbol{E}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}\end{aligned}$$

and is equal to the CSTA undual of the dual CSTA GOPNS 5-vector *hyperplane* $\boldsymbol{E}^*_{\mathcal{C}}$ (§4.5.13) up to a homogeneous scalar factor. The squared hyperbolic distance (squared modulus) $d^2 = \boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{n}_{\mathcal{M}}$ from the origin is constant for all points on the hyperplane. The hyperplane $\boldsymbol{E}_{\mathcal{C}}$ is the set of points

$$\mathbb{NI}_G(\boldsymbol{E}_{\mathcal{C}}) \;=\; \{\; \boldsymbol{T}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{t}_{\mathcal{M}}) \;:\; \boldsymbol{T}_{\mathcal{C}} \cdot \boldsymbol{E}_{\mathcal{C}} = 0 \;\}$$

of the geometric inner product null space of $\boldsymbol{E}_{\mathcal{C}}$, denoted $\mathbb{NI}_G(\boldsymbol{E}_{\mathcal{C}})$ [11]. A similar set holds for all other GIPNS entities.

Two hyperplanes intersect as a CSTA GIPNS 2-vector *plane* $\boldsymbol{\Pi}_{\mathcal{C}}$

$$\begin{aligned}\boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2} &= \\ (\boldsymbol{n}_1 + d_1^2\mathbf{e}_{\infty\gamma}) \wedge (\boldsymbol{n}_2 + d_2^2\mathbf{e}_{\infty\gamma}) &= \\ \boldsymbol{n}_1 \wedge \boldsymbol{n}_2 + (d_2^2\boldsymbol{n}_1 - d_1^2\boldsymbol{n}_2)\mathbf{e}_{\infty\gamma} &= \\ \boldsymbol{D}^{*\mathcal{M}} - (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{D}^{*\mathcal{M}})\mathbf{e}_{\infty\gamma} &= \boldsymbol{\Pi}_{\mathcal{C}}\end{aligned}$$

where $\boldsymbol{D}^{*\mathcal{M}} = \boldsymbol{n}_1 \wedge \boldsymbol{n}_2$ is the STA dual of the plane $\boldsymbol{\Pi}_{\mathcal{C}}$ direction bivector $\boldsymbol{D}$.

Three hyperplanes intersect as a CSTA GIPNS 3-vector *line* $\boldsymbol{L}_{\mathcal{C}}$

$$\begin{aligned}\boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2} \wedge \boldsymbol{E}_{\mathcal{C}_3} &= \\ (\boldsymbol{n}_1 \wedge \boldsymbol{n}_2 + (d_2^2\boldsymbol{n}_1 - d_1^2\boldsymbol{n}_2)\mathbf{e}_{\infty\gamma}) \wedge (\boldsymbol{n}_3 + d_3^2\mathbf{e}_{\infty\gamma}) &= \\ \boldsymbol{n}_1 \wedge \boldsymbol{n}_2 \wedge \boldsymbol{n}_3 + (d_2^2\boldsymbol{n}_1 - d_1^2\boldsymbol{n}_2) \wedge \mathbf{e}_{\infty\gamma} \wedge \boldsymbol{n}_3 + d_3^2\boldsymbol{n}_1 \wedge \boldsymbol{n}_2 \wedge \mathbf{e}_{\infty\gamma} &= \\ \boldsymbol{n}_1 \wedge \boldsymbol{n}_2 \wedge \boldsymbol{n}_3 + (d_1^2\boldsymbol{n}_2 \wedge \boldsymbol{n}_3 - d_2^2\boldsymbol{n}_1 \wedge \boldsymbol{n}_3 + d_3^2\boldsymbol{n}_1 \wedge \boldsymbol{n}_2)\mathbf{e}_{\infty\gamma} &= \\ \boldsymbol{d}^{*\mathcal{M}} + (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{d}^{*\mathcal{M}}) \wedge \mathbf{e}_{\infty\gamma} &= \boldsymbol{L}_{\mathcal{C}}\end{aligned}$$

where $\boldsymbol{d}^{*\mathcal{M}} = \boldsymbol{n}_1 \wedge \boldsymbol{n}_2 \wedge \boldsymbol{n}_3$ is the STA dual of the line $\boldsymbol{L}_{\mathcal{C}}$ direction vector $\boldsymbol{d}$.

Four hyperplanes intersect as a CST GIPNS 4-vector *flat point*

$$\begin{aligned}\boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2} \wedge \boldsymbol{E}_{\mathcal{C}_3} \wedge \boldsymbol{E}_{\mathcal{C}_4} &= \\ (\boldsymbol{n}_1 \wedge \boldsymbol{n}_2 \wedge \boldsymbol{n}_3 + (d_1^2\boldsymbol{n}_2 \wedge \boldsymbol{n}_3 - d_2^2\boldsymbol{n}_1 \wedge \boldsymbol{n}_3 + d_3^2\boldsymbol{n}_1 \wedge \boldsymbol{n}_2)\mathbf{e}_{\infty\gamma}) \wedge (\boldsymbol{n}_4 + d_4^2\mathbf{e}_{\infty\gamma}) &= \\ \alpha\mathbf{I}_{\mathcal{M}} - \alpha(\boldsymbol{p}_{\mathcal{M}} \cdot \mathbf{I}_{\mathcal{M}})\mathbf{e}_{\infty\gamma} &\simeq \\ \mathbf{I}_{\mathcal{M}} + \boldsymbol{p}^*_{\mathcal{M}}\mathbf{e}_{\infty\gamma} &= \mathbb{P}_{\mathcal{C}}.\end{aligned}$$

The CSTA dual of the flat point is

$$\begin{aligned}(\mathbf{I}_{\mathcal{M}} + \boldsymbol{p}^*_{\mathcal{M}}\mathbf{e}_{\infty\gamma})\mathbf{I}_{\mathcal{C}}^{-1} &= \\ \mathbf{e}_{\infty\gamma} \wedge \boldsymbol{P}_{\mathcal{C}} &= \mathbb{P}^*_{\mathcal{C}}.\end{aligned}$$

Five hyperplanes intersect as the CSTA GIPNS 5-vector *point* at infinity $\mathbf{e}^\star_{\infty\gamma}$

$$\begin{aligned}\boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2} \wedge \boldsymbol{E}_{\mathcal{C}_3} \wedge \boldsymbol{E}_{\mathcal{C}_4} \wedge \boldsymbol{E}_{\mathcal{C}_5} &= \\ (\mathbf{I}_{\mathcal{M}} + \boldsymbol{p}^*_{\mathcal{M}}\mathbf{e}_{\infty\gamma}) \wedge (\boldsymbol{n}_4 + d_4^2\mathbf{e}_{\infty\gamma}) &= \\ \boldsymbol{p}^*_{\mathcal{M}} \wedge \mathbf{e}_{\infty\gamma} \wedge \boldsymbol{n}_4 + d_4^2\mathbf{I}_{\mathcal{M}} \wedge \mathbf{e}_{\infty\gamma} &\simeq \\ \mathbf{I}_{\mathcal{M}} \wedge \mathbf{e}_{\infty\gamma} &= \\ \mathbf{e}_{\infty\gamma}\mathbf{I}_{\mathcal{C}} &= \mathbf{e}^\star_{\infty\gamma}.\end{aligned}$$

4.4.4 CSTA GIPNS 1-vector hyperhyperboloid of one sheet

The implicit quadric surface equation for a circular *hyperhyperboloid of one sheet* is

$$r_0^2+(w-p_w)^2-(x-p_x)^2-(y-p_y)^2-(z-p_z)^2 \;=\; 0$$

where r_0 is the initial radius of the expanding sphere in space with time-varying radius

$$r \;=\; \sqrt{r_0^2+(w-p_w)^2}=\sqrt{r_0^2+(ct-p_w)^2}$$

and center position

$$\boldsymbol{p}_{\mathcal{M}} \;=\; p_w\boldsymbol{\gamma}_0+p_x\boldsymbol{\gamma}_1+p_y\boldsymbol{\gamma}_2+p_z\boldsymbol{\gamma}_3=p_w\boldsymbol{\gamma}_0+\mathbf{p}_{\mathcal{S}}$$

in 4-D spacetime.

When $w-p_w=0$, the surface is a sphere with radius r_0. The circular hyperhyperboloid of one sheet can also be called a *hyperpseudosphere*. Like a sphere, a hyperpseudosphere does not include the point at infinity.

In 3-D spacetime with only two spatial dimensions by holding $(z-p_z)=0$, the circular hyperhyperboloid of one sheet reduces to the circular *hyperboloid of one sheet*

$$\frac{(x-p_x)^2}{r_0^2}+\frac{(y-p_y)^2}{r_0^2}-\frac{(w-p_w)^2}{r_0^2} \;=\; 1.$$

When $(w-p_w)=0$, the hyperboloid of one sheet is a circle in the xy-plane with initial radius r_0 at initial time $t=p_w/c$, or $w=p_w$. The circle radius $r=\sqrt{r_0^2+(w-p_w)^2}$ is expanding after time $t=p_w/c$ and is contracting before that time. The radius is expanding with time t at the rate

$$\dot{r}=\frac{\partial r}{\partial t} \;=\; \frac{1}{2}r^{-1}2(ct-p_w)c=\frac{ct-p_w}{r}c=\frac{ct-p_w}{\sqrt{(ct-p_w)^2+r_0^2}}c.$$

The initial rate at time $t=p_w/c$ is $\dot{r}(p_w/c)=0$ and increases to $\dot{r}(\infty)=c$ as $t\to\infty$. In natural units, $c=1$ and the hyperhyperboloid of one sheet is asymptotically the hypercone of a spherically expanding point $\boldsymbol{P}_{\mathcal{C}}$. The acceleration of the radius r is

$$\ddot{r}=\frac{\partial \dot{r}}{\partial t} \;=\; \partial_t\frac{ct-p_w}{r}c=\frac{cr-\dot{r}(ct-p_w)}{r^2}c=\frac{c^2-\dot{r}^2}{r}.$$

The initial acceleration at $t=p_w/c$ is $\ddot{r}(p_w/c)=c^2/r_0$ and decreases to $\ddot{r}(\infty)=0$ as $t\to\infty$. In natural units, $c=1$ and $\ddot{r}$ is a measure of circular, or spherical, curvature at time t.

Using the CSTA point value-extraction elements (§4.3) the hyperhyperboloid of one sheet implicit surface function entity can be written

$$\begin{aligned}
r_0^2+(w-p_w)^2-(x-p_x)^2-(y-p_y)^2-(z-p_x)^2 \;&=\\
(r_0^2+\boldsymbol{p}_{\mathcal{M}}^2)C_1+C_{\boldsymbol{t}^2}-2p_wC_w-2p_xC_x-2p_yC_y-2p_zC_z \;&=\\
-(r_0^2+\boldsymbol{p}_{\mathcal{M}}^2)\mathbf{e}_{\infty\gamma}-2\mathbf{e}_{o\gamma}-2p_w\boldsymbol{\gamma}_0-2p_x\boldsymbol{\gamma}_1-2p_y\boldsymbol{\gamma}_2-2p_z\boldsymbol{\gamma}_3 \;&=\\
-2\boldsymbol{p}_{\mathcal{M}}-(r_0^2+\boldsymbol{p}_{\mathcal{M}}^2)\mathbf{e}_{\infty\gamma}-2\mathbf{e}_{o\gamma} \quad .&
\end{aligned}$$

Normalizing $\mathbf{e}_{o\gamma}$ by scaling $-1/2$ gives

$$\begin{aligned}
\boldsymbol{p}_{\mathcal{M}}+\frac{1}{2}(r_0^2+\boldsymbol{p}_{\mathcal{M}}^2)\mathbf{e}_{\infty\gamma}+\mathbf{e}_{o\gamma} \;&=\\
\boldsymbol{P}_{\mathcal{C}}+\frac{1}{2}r_0^2\mathbf{e}_{\infty\gamma} \quad .&
\end{aligned}$$

The CSTA GIPNS 1-vector *hyperhyperboloid of one sheet (hyperpseudosphere)* $\boldsymbol{\Sigma}_{\mathcal{C}}$ in spacetime with initial radius r_0 centered at CSTA point $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ is defined as

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}} + \frac{1}{2} r_0^2 \mathbf{e}_{\infty\gamma} \\ &\simeq \boldsymbol{\Sigma}_{\mathcal{C}}^* \mathbf{I}_{\mathcal{C}} \end{aligned}$$

and equals the CSTA undual of the dual CSTA GOPNS 5-vector *hyperpseudosphere* $\boldsymbol{\Sigma}_{\mathcal{C}}^*$ up to a homogeneous scalar factor.

The CSTA hyperpseudosphere $\boldsymbol{\Sigma}_{\mathcal{C}}$ is similar to a CGA sphere $\mathbf{S}$ discussed in [6] *when* $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\mathbf{p}_{\mathcal{S}})$ is the embedding of a spatial point $\boldsymbol{p}_{\mathcal{M}} = \mathbf{p}_{\mathcal{S}}$ with $w = ct = 0$. When $r_0 = 0$, $\boldsymbol{\Sigma}_{\mathcal{C}} = \boldsymbol{K}_{\mathcal{C}}$ is a *hypercone*, which is the CSTA point embedding $\boldsymbol{P}_{\mathcal{C}} = \boldsymbol{K}_{\mathcal{C}}$ as a GIPNS entity. Two hyperpseudospheres can intersect in a spatial sphere, spacetime pseudosphere, or spacetime cone.

4.4.5 CSTA GIPNS 1-vector hyperhyperboloid of two sheets

The implicit quadric surface equation for a circular *hyperhyperboloid of two sheets* is

$$-r_0^2 + (w - p_w)^2 - (x - p_x)^2 - (y - p_y)^2 - (z - p_z)^2 = 0.$$

The CSTA GIPNS 1-vector *hyperhyperboloid of two sheets (imaginary hyperpseudosphere)* is

$$\boldsymbol{\Xi}_{\mathcal{C}} = \boldsymbol{P}_{\mathcal{C}} - \frac{1}{2} r_0^2 \mathbf{e}_{\infty\gamma}.$$

The imaginary radius is $\sqrt{-1} r_0$.

The intersection of $\boldsymbol{\Xi}_{\mathcal{C}}$ and hyperplane $\boldsymbol{E}_{\mathcal{C}} = \boldsymbol{\gamma}_0 + p_w \mathbf{e}_{\infty\gamma}$ holds $w = p_w$ and produces an imaginary sphere. The intersection of $\boldsymbol{\Xi}_{\mathcal{C}}$ and hyperplane $\boldsymbol{E}_{\mathcal{C}} = \boldsymbol{\gamma}_3 - p_z \mathbf{e}_{\infty\gamma}$ holds $z = p_z$ and produces a hyperboloid of two sheets in wxy-spacetime opening up and down the w-axis. The intersection of $\boldsymbol{\Xi}_{\mathcal{C}}$ and spacetime plane $\boldsymbol{\Pi}_{\mathcal{C}}$ produces a hyperbola in the spacetime plane that opens up and down the time axis.

4.4.6 CSTA GIPNS 2-vector spatial sphere

The implicit quadric surface equation for a *hyperpseudosphere* is

$$r_0^2 + (w - p_w)^2 - (x - p_x)^2 - (y - p_y)^2 - (z - p_z)^2 = 0.$$

If we set the time coordinate w, then we get the implicit surface equation for a *sphere* in xyz-space

$$(x - p_x)^2 + (y - p_y)^2 + (z - p_z)^2 - r^2 = 0$$

with radius

$$r = \sqrt{r_0^2 + (w - p_w)^2}.$$

To set w, we can intersect a hyperpseudosphere

$$\boldsymbol{\Sigma}_{\mathcal{C}} = \boldsymbol{P}_{\mathcal{C}} + \frac{1}{2} r_0^2 \mathbf{e}_{\infty\gamma}$$

with radius r_0 centered at

$$\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}}) = \mathcal{C}(p_w \boldsymbol{\gamma}_0 + p_x \boldsymbol{\gamma}_1 + p_y \boldsymbol{\gamma}_2 + p_z \boldsymbol{\gamma}_3) = \mathcal{C}(p_w \boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}})$$

with the hyperplane

$$\boldsymbol{E}_{\mathcal{C}} \;=\; \boldsymbol{\gamma}_0 + w\mathbf{e}_{\infty\gamma}$$

of xyz-space at w. The sphere of radius r_0 centered at $\mathbf{p}_{\mathcal{S}}$ is at the time $w = p_w$.

The CSTA GIPNS 2-vector *sphere* $\boldsymbol{S}_{\mathcal{C}}$ centered at $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ with radius r_0 is the intersection

$$\begin{aligned}
\boldsymbol{S}_{\mathcal{C}} &= \boldsymbol{\Sigma}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}} \\
&= \left(\boldsymbol{P}_{\mathcal{C}} + \frac{1}{2}r_0^2\mathbf{e}_{\infty\gamma}\right) \wedge (\boldsymbol{\gamma}_0 + (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{\gamma}_0)\mathbf{e}_{\infty\gamma}) \\
&= \left(\boldsymbol{P}_{\mathcal{C}} + \frac{1}{2}r_0^2\mathbf{e}_{\infty\gamma}\right) \wedge (\boldsymbol{\gamma}_0 + p_w\mathbf{e}_{\infty\gamma}) \\
&\simeq \boldsymbol{S}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}
\end{aligned}$$

which is equal to the CSTA undual of the dual CSTA GOPNS 4-vector *sphere* $\boldsymbol{S}^*_{\mathcal{C}}$ up to a homogeneous scalar factor.

4.4.7 CSTA GIPNS 2-vector spacetime hyperboloid of one sheet

The implicit quadric surface equation for a *hyperpseudosphere* is

$$r_0^2 + (w - p_w)^2 - (x - p_x)^2 - (y - p_y)^2 - (z - p_z)^2 \;=\; 0.$$

If we set one spatial coordinate, for instance z, then we get the implicit surface equation for a circular *hyperboloid of one sheet* in wxy-spacetime

$$\begin{aligned}
(x - p_x)^2 + (y - p_y)^2 - (w - p_w)^2 - (r_0^2 - (z - p_z)^2) \;&=\; \\
\frac{(x - p_x)^2}{r^2} + \frac{(y - p_y)^2}{r^2} - \frac{(w - p_w)^2}{r^2} - 1 \;&=\; 0
\end{aligned}$$

with central radius

$$r \;=\; \sqrt{r_0^2 - (z - p_z)^2}$$

and circular conic section radius at w, z

$$r_c \;=\; \sqrt{(w - p_w)^2 + (r_0^2 - (z - p_z)^2)} = \sqrt{(w - p_w)^2 + r^2}.$$

The spacetime hyperboloid of one sheet is also called a *pseudosphere*. Like a sphere, a pseudosphere does not include the point at infinity. The pseudosphere in wxy-spacetime is a *circle* in xy-space that changes in radius with w, z. To set z, we can intersect a hyperpseudosphere

$$\boldsymbol{\Sigma}_{\mathcal{C}} \;=\; \boldsymbol{P}_{\mathcal{C}} + \frac{1}{2}r_0^2\mathbf{e}_{\infty\gamma}$$

with radius r_0 centered at

$$\boldsymbol{P}_{\mathcal{C}} \;=\; \mathcal{C}(\boldsymbol{p}_{\mathcal{M}}) = \mathcal{C}(p_w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3) = \mathcal{C}(p_w\boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}})$$

with the hyperplane

$$\begin{aligned}
\boldsymbol{E}_{\mathcal{C}} &= z\boldsymbol{\gamma}_3 - z^2\mathbf{e}_{\infty\gamma} \\
&\simeq \boldsymbol{\gamma}_3 - z\mathbf{e}_{\infty\gamma}
\end{aligned}$$

of wxy-space at $z\boldsymbol{\gamma}_3$. To have pseudosphere central radius r_0, set $z = p_z$. The xy-plane circle of radius r_0 centered at $\mathbf{p}_{\mathcal{S}}$ is at the time $w = p_w$. More generally, the hyperplane can be through point $\boldsymbol{p}_{\mathcal{M}}$ with spatial normal vector $\mathbf{n}_{\mathcal{S}}$ as

$$\boldsymbol{E}_{\mathcal{C}} \;=\; \mathbf{n}_{\mathcal{S}} + (\boldsymbol{p}_{\mathcal{M}} \cdot \mathbf{n}_{\mathcal{S}})\mathbf{e}_\infty$$

which makes the hyperboloid of one sheet in the $\boldsymbol{\gamma}_0\mathbf{n}^*_{\mathcal{S}}$-space. The $\boldsymbol{\gamma}_0\mathbf{n}^*_{\mathcal{S}}$-space can also be called the wuv-space with time coordinate w and spatial plane coordinates u and v orthogonal to normal vector $n\mathbf{n}_{\mathcal{S}}$.

The CSTA GIPNS 2-vector *spacetime hyperboloid of one sheet (pseudosphere)* $\boldsymbol{S}_{\mathcal{C}}$ centered at $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ with central radius r_0 in the spatial plane orthogonal to normal vector $\mathbf{n}_{\mathcal{S}}$ is the intersection

$$\begin{aligned}
\boldsymbol{S}_{\mathcal{C}} &= \boldsymbol{\Sigma}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}} \\
&= \left(\boldsymbol{P}_{\mathcal{C}} + \frac{1}{2} r_0 \mathbf{e}_\infty\right) \wedge (\mathbf{n}_{\mathcal{S}} + (\boldsymbol{p}_{\mathcal{M}} \cdot \mathbf{n}_{\mathcal{S}})\mathbf{e}_\infty) \\
&\simeq \boldsymbol{S}^*_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}
\end{aligned}$$

which is equal to the CSTA undual of the dual CSTA GOPNS 4-vector *pseudosphere* $\boldsymbol{S}^*_{\mathcal{C}}$ up to a homogeneous scalar factor.

The CSTA GIPNS null 2-vector *spacetime cone (null cone)* is the pseudosphere $\boldsymbol{S}_{\mathcal{C}}$ with central radius $r = 0$.

The spacetime hyperboloid of one sheet is always in wuv-spacetime and is not a spatial surface, but is a circle in the $\mathbf{n}^*_{\mathcal{S}}$-plane centered at $\mathbf{p}_{\mathcal{S}}$ with radius $r_c = \sqrt{(w - p_w)^2 + r_0^2}$. The CSTA GIPNS 3-vector *circle* entity $\boldsymbol{C}_{\mathcal{C}}$ with radius r_0 is obtained by another intersection with the hyperplane at $p_w\boldsymbol{\gamma}_0$.

4.4.8 CSTA GIPNS 2-vector spacetime hyperboloid of two sheets

The CSTA GIPNS 2-vector *spacetime hyperboloid of two sheets (imaginary pseudosphere)* $\boldsymbol{S}_{\mathcal{C}}$ centered at $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ with central radius r_0 in the spatial plane orthogonal to normal vector $\mathbf{n}_{\mathcal{S}}$ is the intersection

$$\begin{aligned}
\boldsymbol{S}_{\mathcal{C}} &= \boldsymbol{\Xi}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}} \\
&= \left(\boldsymbol{P}_{\mathcal{C}} - \frac{1}{2} r_0 \mathbf{e}_\infty\right) \wedge (\mathbf{n}_{\mathcal{S}} + (\boldsymbol{p}_{\mathcal{M}} \cdot \mathbf{n}_{\mathcal{S}})\mathbf{e}_\infty) \\
&\simeq \boldsymbol{S}^*_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}
\end{aligned}$$

which is equal to the CSTA undual of the dual CSTA GOPNS 4-vector *imaginary pseudosphere* $\boldsymbol{S}^*_{\mathcal{C}}$ up to a homogeneous scalar factor. The two sheets open up and down the w-axis and have circular sections in the $\mathbf{n}^*_{\mathcal{S}}$-plane.

4.4.9 CSTA GIPNS 2-vector plane

A plane in spacetime can be defined by two orthogonal unit-norm direction vectors

$$\begin{aligned}
\boldsymbol{d}_{\mathcal{M}_1} &= \frac{\boldsymbol{d}_{\mathcal{M}_1}}{\|\boldsymbol{d}_{\mathcal{M}_1}\|} = d_{w_1}\boldsymbol{\gamma}_0 + d_{x_1}\boldsymbol{\gamma}_1 + d_{y_1}\boldsymbol{\gamma}_2 + d_{z_1}\boldsymbol{\gamma}_3 \\
\boldsymbol{d}_{\mathcal{M}_2} &= \frac{\boldsymbol{d}_{\mathcal{M}_2}}{\|\boldsymbol{d}_{\mathcal{M}_2}\|} = d_{w_2}\boldsymbol{\gamma}_0 + d_{x_2}\boldsymbol{\gamma}_1 + d_{y_2}\boldsymbol{\gamma}_2 + d_{z_2}\boldsymbol{\gamma}_3 \\
\boldsymbol{d}_{\mathcal{M}_1} \cdot \boldsymbol{d}^\dagger_{\mathcal{M}_2} &= 0
\end{aligned}$$

and a point

$$\boldsymbol{p}_{\mathcal{M}} \;=\; p_w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3$$

on the plane. The direction of the plane is represented by the normalized unit bivector

$$\begin{aligned}\boldsymbol{D} &= \boldsymbol{d}_{\mathcal{M}_1} \wedge \boldsymbol{d}_{\mathcal{M}_2} \\ &= \frac{\boldsymbol{D}}{\sqrt{\boldsymbol{D}\cdot\boldsymbol{D}^\dagger}} = \frac{\boldsymbol{D}}{\sqrt{\boldsymbol{D}\cdot(\boldsymbol{\gamma}_0\boldsymbol{D}^\sim\boldsymbol{\gamma}_0)}}.\end{aligned}$$

The notation $\mathbf{A}^\dagger_{\mathcal{M}} = \boldsymbol{\gamma}_0\mathbf{A}^\sim_{\mathcal{M}}\boldsymbol{\gamma}_0$ is the anti-Euclidean space conjugation, or SA space conjugation, which is necessary for the case where $\boldsymbol{D}$ is a null bivector. For blade $\mathbf{A}_{\mathcal{M}}$ in spacetime, the *conjugate* [11] has the property

$$\mathbf{A}_{\mathcal{M}}\cdot\mathbf{A}^\dagger_{\mathcal{M}} \;=\; \mathbf{A}_{\mathcal{M}}\cdot\left(\boldsymbol{\gamma}_0\widetilde{\mathbf{A}_{\mathcal{M}}}\boldsymbol{\gamma}_0\right) = \|\mathbf{A}_{\mathcal{M}}\|^2.$$

Any test point

$$\boldsymbol{t}_{\mathcal{M}} \;=\; w\boldsymbol{\gamma}_0 + x\boldsymbol{\gamma}_1 + y\boldsymbol{\gamma}_2 + z\boldsymbol{\gamma}_3$$

on the plane must satisfy the plane equation

$$(\boldsymbol{t}_{\mathcal{M}} - \boldsymbol{p}_{\mathcal{M}})\wedge\boldsymbol{D} \;=\; 0$$

which can also be written in the dual form

$$(\boldsymbol{t}_{\mathcal{M}} - \boldsymbol{p}_{\mathcal{M}})\cdot\boldsymbol{D}^{*\mathcal{M}} \;=\; \boldsymbol{t}_{\mathcal{M}}\cdot\boldsymbol{D}^{*\mathcal{M}} - \boldsymbol{p}_{\mathcal{M}}\cdot\boldsymbol{D}^{*\mathcal{M}} = 0.$$

The dual form plane equation is vector-valued and the components represent a system of implicit surface equations for an intersection of hyperplanes that gives the plane.

The CSTA GIPNS 2-vector *plane* $\boldsymbol{\Pi}_{\mathcal{C}}$ through point $\boldsymbol{p}_{\mathcal{M}}$ in the planar direction of the unit bivector $\boldsymbol{D}$ in spacetime can be defined as

$$\begin{aligned}\boldsymbol{\Pi}_{\mathcal{C}} &= \boldsymbol{D}^{*\mathcal{M}} - (\boldsymbol{p}_{\mathcal{M}}\cdot\boldsymbol{D}^{*\mathcal{M}})\wedge\mathbf{e}_{\infty\gamma} \\ &\simeq \boldsymbol{\Pi}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}\end{aligned}$$

which is equal to the CSTA undual of the dual CSTA GOPNS 4-vector *plane* $\boldsymbol{\Pi}^*_{\mathcal{C}}$ up to a homogeneous scalar factor.

The CSTA translation operation on any CSTA entity can be defined as its successive reflections in two parallel CSTA planes. The CSTA 2-versor *translator* (translation operator) $T_{\mathcal{C}}$ can be defined by two parallel planes $\boldsymbol{\Pi}_{\mathcal{C}_1}$ and $\boldsymbol{\Pi}_{\mathcal{C}_2}$ that are separated by a spacetime displacement vector $\frac{1}{2}\boldsymbol{d}_{\mathcal{M}}$ from $\boldsymbol{\Pi}_{\mathcal{C}_1}$ to $\boldsymbol{\Pi}_{\mathcal{C}_2}$ as

$$T_{\mathcal{C}} \;=\; \boldsymbol{\Pi}_{\mathcal{C}_2}\boldsymbol{\Pi}_{\mathcal{C}_1}.$$

The translator versor operation on a CSTA point $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$, for example, is

$$\boldsymbol{P}'_{\mathcal{C}} \;=\; T_{\mathcal{C}}\boldsymbol{P}_{\mathcal{C}}T^\sim_{\mathcal{C}} = \boldsymbol{\Pi}_{\mathcal{C}_2}\boldsymbol{\Pi}_{\mathcal{C}_1}\boldsymbol{P}_{\mathcal{C}}\boldsymbol{\Pi}^\sim_{\mathcal{C}_1}\boldsymbol{\Pi}^\sim_{\mathcal{C}_2} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + \boldsymbol{d}_{\mathcal{M}}).$$

The successive reflections in two parallel planes translates by *twice* the spacetime displacement between the parallel planes.

The *rotor* (spatial rotation operator) $R_{\mathcal{S}}$ for a rotation by *twice* the angle between two non-parallel spatial planes $\boldsymbol{\Pi}_{\mathcal{C}_1}$ and $\boldsymbol{\Pi}_{\mathcal{C}_2}$ can be defined as

$$R_{\mathcal{S}} \;=\; \boldsymbol{\Pi}_{\mathcal{C}_2}\boldsymbol{\Pi}_{\mathcal{C}_1}.$$

The spatial rotation operator $R_{\mathcal{S}}$ is equivalent to the SA rotor and is the same spatial rotor that is used in STA and CSTA. The spatial rotor $R_{\mathcal{S}}$ can spatially rotate any multivector by *versor outermorphism* [11] that rotates all vectors within outer products.

The double *boost* operator $B_{\mathcal{M}}B_{\mathcal{M}} = B_{\varphi_{\mathbf{v}}}B_{\varphi_{\mathbf{v}}}$ that adds the double rapidity $2\varphi_{\mathbf{v}} = 2\text{atanh}(\beta_{\mathbf{v}})$ in the direction $\mathbf{v}$ can be defined as the successive reflections in two non-parallel spacetime planes. The first plane $\mathbf{\Pi}_{\mathcal{C}_1}$ should represent the observer position as a plane through the origin and observer that spans the time axis and another axis perpendicular to $\mathbf{v}$. The second plane $\mathbf{\Pi}_{\mathcal{C}_2}$ should represent the boost particle position $\boldsymbol{p}_1 = \boldsymbol{v}t = \boldsymbol{o}t + \mathbf{v}t$ by passing through the origin and $\boldsymbol{p}_1$ and spanning the same direction perpendicular to $\mathbf{v}$ as the first plane. The two planes contain an angle, the hyperbolic rapidity angle or area $\varphi_{\mathbf{v}}$, that turns positive from $\mathbf{\Pi}_{\mathcal{C}_1}$ toward $\mathbf{\Pi}_{\mathcal{C}_2}$ into the direction of $\mathbf{v}$. The boost $B_{\mathcal{M}}$ by $\beta_{\mathbf{v}}$, or rotation by $\varphi_{\mathbf{v}}$, that is applied *twice* for addition of rapidity $2\varphi_{\mathbf{v}}$ is obtained by the successive reflections

$$\boldsymbol{p}'_1 \;=\; \mathbf{\Pi}_{\mathcal{C}_2}\mathbf{\Pi}_{\mathcal{C}_1}\boldsymbol{v}t\mathbf{\Pi}^{\sim}_{\mathcal{C}_1}\mathbf{\Pi}^{\sim}_{\mathcal{C}_2} = B_{\mathcal{M}}B_{\mathcal{M}}\boldsymbol{v}\mathrm{t}\mathrm{B}^{\sim}_{\mathcal{M}}B^{\sim}_{\mathcal{M}}.$$

The position normalization is then applied as

$$\boldsymbol{p}''_1 \;=\; ct\frac{\boldsymbol{p}'_1}{\boldsymbol{p}'_1 \cdot \boldsymbol{\gamma}_0}$$

where t is the proper time of the same observer $\boldsymbol{o}$ that has not changed.

4.4.10 CSTA GIPNS 3-vector line

An implicit equation for a line in spacetime through two points can be written as

$$(\boldsymbol{t} - \boldsymbol{p}_1) \cdot (\boldsymbol{p}_2 - \boldsymbol{p}_1)^{*\mathcal{M}} \;=\; 0$$

where $\boldsymbol{t}$ is the CSTA test point. The equation holds good for any $\boldsymbol{t}$ on the line of the two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. The unit norm direction $\boldsymbol{d}$ of the line can be written as

$$\begin{aligned}
\boldsymbol{d} \;&=\; \frac{\boldsymbol{p}_2 - \boldsymbol{p}_1}{\sqrt{(\boldsymbol{p}_2 - \boldsymbol{p}_1) \cdot (\boldsymbol{p}_2 - \boldsymbol{p}_1)^{\dagger}}} \\
&=\; \frac{\boldsymbol{p}_2 - \boldsymbol{p}_1}{\sqrt{(\boldsymbol{p}_2 - \boldsymbol{p}_1) \cdot (\boldsymbol{\gamma}_0(\boldsymbol{p}_2 - \boldsymbol{p}_1)\boldsymbol{\gamma}_0)}} \\
&=\; \frac{\boldsymbol{p}_2 - \boldsymbol{p}_1}{\|\boldsymbol{p}_2 - \boldsymbol{p}_1\|}.
\end{aligned}$$

The unit norm trivector dual to the line direction is

$$\boldsymbol{d}^{*\mathcal{M}} \;=\; \boldsymbol{d}\mathbf{I}^{-1}_{\mathcal{M}}.$$

The implicit equation can be rewritten as

$$\begin{aligned}
(\boldsymbol{t} - \boldsymbol{p}) \cdot \boldsymbol{d}^{*\mathcal{M}} \;&=\; \\
\boldsymbol{t} \cdot \boldsymbol{d}^{*\mathcal{M}} - \boldsymbol{p} \cdot \boldsymbol{d}^{*\mathcal{M}} \;&=\; 0
\end{aligned}$$

where $\boldsymbol{p}$ is any point on the line.

The CSTA 3-vector *line* entity $\boldsymbol{L}_{\mathcal{C}}$ through point $\boldsymbol{p}_{\mathcal{M}}$ in the direction of unit norm vector $\boldsymbol{d}_{\mathcal{M}}$ can be defined as

$$\begin{aligned}
\boldsymbol{L}_{\mathcal{C}} \;&=\; \boldsymbol{d}^{*\mathcal{M}} + (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{d}^{*\mathcal{M}}) \wedge \mathbf{e}_{\infty\gamma} \\
&\simeq\; \boldsymbol{L}^{*}_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}
\end{aligned}$$

which is equal to the CSTA undual of the dual CSTA GOPNS 3-vector *line* $\boldsymbol{L}^*_{\mathcal{C}}$ up to a homogeneous scalar factor.

4.4.11 CSTA GIPNS 3-vector spatial circle

The CSTA GIPNS 1-vector *hyperpseudosphere* with radius r_0

$$\boldsymbol{\Sigma}_{\mathcal{C}} \;=\; \boldsymbol{P}_{\mathcal{C}} + \frac{1}{2} r_0^2 \mathbf{e}_{\infty\gamma}$$

centered at

$$\boldsymbol{P}_{\mathcal{C}} \;=\; \mathcal{C}(\boldsymbol{p}_{\mathcal{M}}) = \mathcal{C}(p_w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3)$$

can be intersected with two CSTA GIPNS 1-vector *hyperplanes*

$$\begin{aligned} \boldsymbol{E}_{\mathcal{C}_1} &= \mathbf{n}_{\mathcal{S}} + (\boldsymbol{p}_{\mathcal{M}} \cdot \mathbf{n}_{\mathcal{S}})\mathbf{e}_{\infty\gamma} \\ \boldsymbol{E}_{\mathcal{C}_2} &= \boldsymbol{\gamma}_0 + (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{\gamma}_0)\mathbf{e}_{\infty\gamma} \end{aligned}$$

to obtain a circle with radius r_0 centered at $\boldsymbol{p}_{\mathcal{M}}$ in the spatial plane through $\boldsymbol{p}_{\mathcal{M}}$ with direction bivector $\mathbf{N}_{\mathcal{S}} = \mathbf{n}^*_{\mathcal{S}} = -\mathbf{n}_{\mathcal{S}}\mathbf{I}_{\mathcal{S}}^{-1}$.

The CSTA GIPNS 3-vector *circle* entity $\boldsymbol{C}_{\mathcal{C}}$ centered at $\boldsymbol{p}_{\mathcal{M}}$ with radius r_0 at time p_w in the plane of bivector $\mathbf{N}_{\mathcal{S}} = \mathbf{n}^*_{\mathcal{S}} = -\mathbf{n}_{\mathcal{S}}\mathbf{I}_{\mathcal{S}}^{-1}$ dual to normal vector $\mathbf{n}_{\mathcal{S}}$ can be formed as

$$\begin{aligned} \boldsymbol{C}_{\mathcal{C}} &= \boldsymbol{\Sigma}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2} \\ &= \boldsymbol{S}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}_2} \end{aligned}$$

Without setting the time $w = p_w$ by intersecting $\boldsymbol{E}_{\mathcal{C}_2}$, the circle changes radius with time as the CSTA GIPNS 2-vector *hyperboloid (pseudosphere)* $\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{\Sigma}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}_1}$ (§4.4.7).

The CSTA GIPNS 3-vector *circle* $\boldsymbol{C}_{\mathcal{C}}$ can also be represented as the intersection of the CSTA GIPNS 1-vector *hyperpseudosphere* $\boldsymbol{\Sigma}_{\mathcal{C}}$ and CSTA GIPNS 2-vector *plane* $\boldsymbol{\Pi}_{\mathcal{C}}$ as

$$\boldsymbol{C}_{\mathcal{C}} \;=\; \boldsymbol{\Sigma}_{\mathcal{C}} \wedge \boldsymbol{\Pi}_{\mathcal{C}}$$

where the hyperpseudosphere $\boldsymbol{\Sigma}_{\mathcal{C}}$ is the same as above and sets the center $\boldsymbol{p}_{\mathcal{M}}$ and radius r_0, and the plane $\boldsymbol{\Pi}_{\mathcal{C}}$ with spatial direction bivector $\mathbf{N}_{\mathcal{S}}$ through point $\boldsymbol{p}_{\mathcal{M}}$ is

$$\begin{aligned} \boldsymbol{\Pi}_{\mathcal{C}} &= \boldsymbol{D}^{*\mathcal{M}} - (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{D}^{*\mathcal{M}}) \wedge \mathbf{e}_{\infty\gamma} \\ &= \mathbf{N}^{*\mathcal{M}}_{\mathcal{S}} - (\boldsymbol{p}_{\mathcal{M}} \cdot \mathbf{N}^{*\mathcal{M}}_{\mathcal{S}}) \wedge \mathbf{e}_{\infty\gamma} \\ &= \boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2}. \end{aligned}$$

The CSTA GIPNS 3-vector *circle* $\boldsymbol{C}_{\mathcal{C}}$ is equal to the CSTA undual of the dual CSTA GOPNS 3-vector *circle* $\boldsymbol{C}^*_{\mathcal{C}}$

$$\boldsymbol{C}_{\mathcal{C}} \;\simeq\; \boldsymbol{C}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}$$

up to a homogeneous scalar factor.

A CSTA GIPNS 3-vector *circle* $\boldsymbol{C}_{\mathcal{C}}$ can also be formed as the intersection of a CSTA GIPNS 1-vector *imaginary hyperpseudosphere* $\boldsymbol{\Xi}_{\mathcal{C}}$ or CSTA GIPNS 1-vector *hypercone* $\boldsymbol{K}_{\mathcal{C}}$ and CSTA GIPNS 2-vector *spatial plane* $\boldsymbol{\Pi}_{\mathcal{C}}$ as

$$\begin{aligned} \boldsymbol{C}_{\mathcal{C}} &= \boldsymbol{\Xi}_{\mathcal{C}} \wedge \boldsymbol{\Pi}_{\mathcal{C}} \\ &= \boldsymbol{K}_{\mathcal{C}} \wedge \boldsymbol{\Pi}_{\mathcal{C}}. \end{aligned}$$

4.4.12 CSTA GIPNS 3-vector spacetime hyperbola (pseudocircle)

The *circle* $\boldsymbol{C}_{\mathcal{C}}$ with radius r_0 centered at $\boldsymbol{p}_{\mathcal{M}} = p_w\boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}}$ in spatial plane $\mathbf{N}_{\mathcal{S}} = \mathbf{n}^*_{\mathcal{S}}$ is formed by intersecting the plane $\boldsymbol{\Pi}_{\mathcal{C}}$ of $\mathbf{N}_{\mathcal{S}}$ through $\boldsymbol{p}_{\mathcal{M}}$ with the hyperpseudosphere $\boldsymbol{\Sigma}_{\mathcal{C}}$ of radius r_0 at $\boldsymbol{p}_{\mathcal{M}}$. Similarly, the *pseudocircle* $\boldsymbol{C}_{\mathcal{C}}$ with central radius r_0 centered at $\boldsymbol{p}_{\mathcal{M}} = p_w\boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}}$ in Minkowski spacetime plane $\boldsymbol{D}_{\mathcal{M}} = \boldsymbol{\gamma}_0\mathbf{d}_{\mathcal{S}}$ is formed by intersecting the plane $\boldsymbol{\Pi}_{\mathcal{C}}$ of $\boldsymbol{D}_{\mathcal{M}}$ through $\boldsymbol{p}_{\mathcal{M}}$ with the hyperpseudosphere $\boldsymbol{\Sigma}_{\mathcal{C}}$ of radius r_0 at $\boldsymbol{p}_{\mathcal{M}}$. The hyperbola opens up and down the spatial vector axis $\mathbf{d}_{\mathcal{S}}$ for a hyperpseudosphere $\boldsymbol{\Sigma}_{\mathcal{C}}$, and it opens up and down the time axis $\boldsymbol{\gamma}_0$ for an imaginary hyperpseudosphere $\boldsymbol{\Xi}_{\mathcal{C}}$.

The CSTA GIPNS 3-vector *spacetime hyperbola (pseudocircle)* $\boldsymbol{C}_{\mathcal{C}}$ can be defined as

$$\boldsymbol{C}_{\mathcal{C}} \;=\; \boldsymbol{\Sigma}_{\mathcal{C}} \wedge \boldsymbol{\Pi}_{\mathcal{C}}$$

where the hyperpseudosphere $\boldsymbol{\Sigma}_{\mathcal{C}}$ sets the central position $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ and initial radius r_0 as

$$\boldsymbol{\Sigma}_{\mathcal{C}} \;=\; \boldsymbol{P}_{\mathcal{C}} + \frac{1}{2} r_0 \mathbf{e}_{\infty\gamma}$$

and the plane $\boldsymbol{\Pi}_{\mathcal{C}}$ sets the Minkowski spacetime plane $\boldsymbol{D} = \boldsymbol{\gamma}_0\mathbf{d}_{\mathcal{S}}$ of spatial unit direction vector $\mathbf{d}_{\mathcal{S}}$ and time direction $\boldsymbol{\gamma}_0$ as

$$\boldsymbol{\Pi}_{\mathcal{C}} \;=\; \boldsymbol{D}^{*\mathcal{M}} - (\boldsymbol{p}_{\mathcal{M}} \cdot \boldsymbol{D}^{*\mathcal{M}}) \wedge \mathbf{e}_{\infty\gamma}.$$

The hyperbola can be visualized as a point pair on the spatial line $\mathbf{d}_{\mathcal{S}}$, centered on $\mathbf{p}_{\mathcal{S}}$, and separated by an initial distance $2r = 2r_0$ at time $w = p_w$. As time w changes away from the initial time p_w, the radius r increases to $r = \sqrt{r_0^2 + (w - p_w)^2}$. The CSTA GIPNS 4-vector spacetime point pair can be obtained as

$$\mathbf{2}_{\mathcal{C}} \;=\; \boldsymbol{S}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}}$$

where $\boldsymbol{E}_{\mathcal{C}}$ is the xyz-space hyperplane

$$\boldsymbol{E}_{\mathcal{C}} \;=\; \boldsymbol{\gamma}_0 + w\mathbf{e}_{\infty\gamma}$$

at the time w for the point pair with radius r around $\mathbf{p}_{\mathcal{S}}$ on the line direction $\mathbf{d}_{\mathcal{S}}$. The hyperplane sets the time w component of the points in the spacetime point pair. The points appear to move apart spatially with time away from p_w.

A CSTA GIPNS 3-vector *spacetime hyperbola (pseudocircle)* $\boldsymbol{C}_{\mathcal{C}}$ can also be formed as the intersection of a CSTA GIPNS 1-vector *imaginary hyperpseudosphere* $\boldsymbol{\Xi}_{\mathcal{C}}$ or CSTA GIPNS 1-vector *hypercone* $\boldsymbol{K}_{\mathcal{C}}$ and CSTA GIPNS 2-vector *spacetime plane* $\boldsymbol{\Pi}_{\mathcal{C}}$ as

$$\begin{aligned} \boldsymbol{C}_{\mathcal{C}} \;&=\; \boldsymbol{\Xi}_{\mathcal{C}} \wedge \boldsymbol{\Pi}_{\mathcal{C}} \\ &=\; \boldsymbol{K}_{\mathcal{C}} \wedge \boldsymbol{\Pi}_{\mathcal{C}} \end{aligned}$$

which open up and down the time w-axis.

4.4.13 CSTA GIPNS 4-vector point pair

The CSTA GIPNS 4-vector *point pair* $\mathbf{2}_{\mathcal{C}}$ is

$$\begin{aligned} \mathbf{2}_{\mathcal{C}} \;&=\; -\boldsymbol{P}_{\mathcal{C}_1} \cdot \boldsymbol{P}^*_{\mathcal{C}_2} \\ &=\; -\boldsymbol{P}_{\mathcal{C}_1} \cdot (\boldsymbol{P}_{\mathcal{C}_2}\mathbf{I}_{\mathcal{C}}^{-1}) = \boldsymbol{P}_{\mathcal{C}_1} \cdot (\boldsymbol{P}_{\mathcal{C}_2}\mathbf{I}_{\mathcal{C}}) = (\boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2})\mathbf{I}_{\mathcal{C}} \\ &=\; \mathbf{2}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}} \end{aligned}$$

which is exactly the CSTA undual of the dual CSTA GOPNS 2-vector *point pair* $\mathbf{2}^*_{\mathcal{C}}$.

If the two points are *relatively lightlike*, then the point pair is actually the CSTA GIPNS null 4-vector *light-line (null line)* $\mathcal{L}_{\mathcal{C}}$ that is exactly undual to the dual CSTA GOPNS null 2-vector *light-line (null line)* $\mathcal{L}^*_{\mathcal{C}}$ (§4.5.4). The point pair $\mathbf{2}_{\mathcal{C}}$ of two not relatively lightlike points is *non-null*.

If one of the two points is $\mathbf{e}_{\infty\gamma}$, then the point pair is actually the CSTA GIPNS 4-vector *flat point* $\mathbb{P}_{\mathcal{C}}$ that is exactly undual to the dual CSTA GOPNS 2-vector *flat point* $\mathbb{P}^*_{\mathcal{C}}$ (§4.5.5). A flat point is non-null.

4.4.14 CSTA GIPNS null 4-vector light-line (null line)

The CSTA GIPNS null 4-vector *null line (light-line)* $\mathcal{L}_{\mathcal{C}}$ is exactly the undual $\mathcal{L}_{\mathcal{C}} = \mathcal{L}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}$ of the dual CSTA GOPNS null 2-vector *null line (light-line)* $\mathcal{L}^*_{\mathcal{C}}$ (§4.5.4).

The CSTA GIPNS null 4-vector *light-line (null line)* $\mathcal{L}_{\mathcal{C}}$ is

$$\begin{aligned}\mathcal{L}_{\mathcal{C}} &= -\boldsymbol{P}_{\mathcal{L}_1}\cdot\boldsymbol{P}^*_{\mathcal{L}_2}\\ &= -\boldsymbol{P}_{\mathcal{L}_1}\cdot(\boldsymbol{P}_{\mathcal{L}_2}\mathbf{I}_{\mathcal{C}}^{-1}) = \boldsymbol{P}_{\mathcal{L}_1}\cdot(\boldsymbol{P}_{\mathcal{L}_2}\mathbf{I}_{\mathcal{C}}) = (\boldsymbol{P}_{\mathcal{L}_1}\wedge\boldsymbol{P}_{\mathcal{L}_2})\mathbf{I}_{\mathcal{C}}\\ &= \mathcal{L}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}\end{aligned}$$

where $\boldsymbol{P}_{\mathcal{L}_i} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}_i}) = \mathcal{C}(p_{w_i}\boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}_i})$ denotes points that are *relatively lightlike* in spacetime positions. The two relatively lightlike points $\boldsymbol{P}_{\mathcal{L}_1}$ and $\boldsymbol{P}_{\mathcal{L}_2}$ are on a light-line in spacetime having equal changes in time components $|p_{w_1} - p_{w_2}|$ to space components $\|\mathbf{p}_{\mathcal{S}_1} - \mathbf{p}_{\mathcal{S}_2}\|$,

$$\begin{aligned}\frac{|p_{w_1} - p_{w_2}|}{\|\mathbf{p}_{\mathcal{S}_1} - \mathbf{p}_{\mathcal{S}_2}\|} &= 1\\ \partial_t\|\mathbf{p}_{\mathcal{S}_1} - \mathbf{p}_{\mathcal{S}_2}\| = \partial_t|p_{w_1} - p_{w_2}| &= c.\end{aligned}$$

Light speed c is required to travel between the two points, or any two points on a light-line, in spacetime. The vector $\boldsymbol{p}_{\mathcal{M}_1} - \boldsymbol{p}_{\mathcal{M}_2}$ is a null vector in spacetime, and any two points in spacetime with a null difference vector are relatively lightlike.

A null 4-vector *light-line* $\mathcal{L}_{\mathcal{C}}$ can be converted into a non-null 3-vector *line* $\boldsymbol{L}_{\mathcal{C}}$ as

$$\boldsymbol{L}_{\mathcal{C}} = (\mathcal{L}^*_{\mathcal{C}}\wedge\mathbf{e}_{\infty\gamma})\mathbf{I}_{\mathcal{C}} = \boldsymbol{L}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}.$$

4.4.15 CSTA GIPNS 4-vector flat point

The CSTA GIPNS 4-vector *flat point* $\mathbb{P}_{\mathcal{C}}$ is

$$\begin{aligned}\mathbb{P}_{\mathcal{C}} &= -\boldsymbol{P}_{\mathcal{C}}\cdot\mathbf{e}^*_{\infty\gamma}\\ &= -\boldsymbol{P}_{\mathcal{C}}\cdot(\mathbf{e}_{\infty\gamma}\mathbf{I}_{\mathcal{C}}^{-1}) = \boldsymbol{P}_{\mathcal{C}}\cdot(\mathbf{e}_{\infty\gamma}\mathbf{I}_{\mathcal{C}}) = (\boldsymbol{P}_{\mathcal{C}}\wedge\mathbf{e}_{\infty\gamma})\mathbf{I}_{\mathcal{C}}\\ &= \mathbb{P}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}\end{aligned}$$

which is exactly the undual of the dual CSTA GOPNS 2-vector *flat point* $\mathbb{P}^*_{\mathcal{C}}$.

A flat spatial point $\mathbb{P}_{\mathcal{C}} = \mathcal{C}(\mathbf{p}_{\mathcal{S}})\wedge\mathbf{e}_{\infty\gamma}$ at $w = 0$ can be represented as the intersection of a CSTA GIPNS 2-vector plane $\boldsymbol{\Pi}_{\mathcal{C}}$ and CSTA GIPNS 3-vector line $\boldsymbol{L}_{\mathcal{C}}$ that are in the common xyz-space hyperplane at any times in spacetime as

$$\begin{aligned}\mathbb{P}_{\mathcal{C}} &= \boldsymbol{\gamma}_0\wedge(\boldsymbol{\gamma}_0\cdot\boldsymbol{\Pi}_{\mathcal{C}})\wedge(\boldsymbol{\gamma}_0\cdot\boldsymbol{L}_{\mathcal{C}})\\ &= \mathbb{P}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}\end{aligned}$$

where the common hyperplane of xyz-space $\boldsymbol{E}_{\mathcal{C}} = \boldsymbol{\gamma}_0$ is contracted out of the plane and line before they are intersected, and then $\boldsymbol{\gamma}_0$ is intersected back into the result. The time components of the plane and line do not affect the result, which is spatial intersection at $w = 0$. The flat spatial point $\mathbb{P}_{\mathcal{C}}$ can exactly match the undual $\mathbb{P}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}$. The flat spatial point represents the point $\boldsymbol{P}_{\mathcal{C}}$ of intersection on the plane where the line passes through, and it also represents $\mathbf{e}_\infty$ where the plane and line also intersect. A flat spacetime point as intersections may also be possible but is not considered here.

4.4.16 CSTA GIPNS 5-vector point

The CSTA null 1-vector *point embedding* $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ is the

- CSTA *GIPNS* null 1-vector *hypercone* $\boldsymbol{P}_{\mathcal{C}}$ centered at $\boldsymbol{p}_{\mathcal{M}}$
- CSTA *GOPNS* null 1-vector *point* $\boldsymbol{P}_{\mathcal{C}}$ representing the point $\boldsymbol{p}_{\mathcal{M}}$.

Therefore, the CSTA GIPNS null 5-vector *point* $\boldsymbol{P}^\star_{\mathcal{C}}$ is the undual

$$\boldsymbol{P}^\star_{\mathcal{C}} \;=\; \boldsymbol{P}_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}$$

which also introduces a notation for the undual operation. The undual notation has been omitted on other undual entities. For the 5-vector point, the undual notation avoids a notational conflict since $\boldsymbol{P}_{\mathcal{C}}$ is the dual, not $\boldsymbol{P}^*_{\mathcal{C}}$.

For STA vectors, the undual is

$$\begin{aligned}
\boldsymbol{\gamma}^\star_0 &= \boldsymbol{\gamma}_0\mathbf{I}_{\mathcal{M}} = \boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3 \\
\boldsymbol{\gamma}^\star_1 &= \boldsymbol{\gamma}_1\mathbf{I}_{\mathcal{M}} = \boldsymbol{\gamma}_0\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3 \\
\boldsymbol{\gamma}^\star_2 &= \boldsymbol{\gamma}_2\mathbf{I}_{\mathcal{M}} = \boldsymbol{\gamma}_0\boldsymbol{\gamma}_3\boldsymbol{\gamma}_1 \\
\boldsymbol{\gamma}^\star_3 &= \boldsymbol{\gamma}_3\mathbf{I}_{\mathcal{M}} = \boldsymbol{\gamma}_0\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2
\end{aligned}$$

which is consistent, in this case, with *Hodge dual* that is denoted $\star A$ in other literature.

The CSTA GIPNS null 5-vector *point* $\boldsymbol{P}^\star_{\mathcal{C}}$ can also be represented as the intersection of a hypercone $\boldsymbol{P}_{\mathcal{C}}$ with the four hyperplanes $\boldsymbol{E}_{\mathcal{C}_i}$ through the hypercone vertex $\boldsymbol{p}_{\mathcal{M}}$ as

$$\boldsymbol{P}^\star_{\mathcal{C}} \;=\; \boldsymbol{P}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2} \wedge \boldsymbol{E}_{\mathcal{C}_3} \wedge \boldsymbol{E}_{\mathcal{C}_4}$$

where

$$\begin{aligned}
\boldsymbol{P}_{\mathcal{C}} &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}}) = \mathcal{C}(p_w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3) \\
\boldsymbol{E}_{\mathcal{C}_1} &= \boldsymbol{\gamma}_0 + p_w\mathbf{e}_{\infty\gamma} \\
\boldsymbol{E}_{\mathcal{C}_2} &= -\boldsymbol{\gamma}_1 + p_x\mathbf{e}_{\infty\gamma} \\
\boldsymbol{E}_{\mathcal{C}_3} &= -\boldsymbol{\gamma}_2 + p_y\mathbf{e}_{\infty\gamma} \\
\boldsymbol{E}_{\mathcal{C}_4} &= -\boldsymbol{\gamma}_3 + p_z\mathbf{e}_{\infty\gamma}.
\end{aligned}$$

Each hyperplane fixes one coordinate to hold a value.

4.5 CSTA GOPNS entities

In $\mathcal{G}_{2,4}$ CSTA, *five or less points* can be wedged into CSTA GOPNS entities, allowing a greater variety of entities than in $\mathcal{G}_{4,1}$ CGA, which uses four or less points.

The familiar CGA GOPNS entities at time $w = ct = 0$ can be formed by wedging four or less points that are the embeddings of spatial points $\mathbf{P}_\mathcal{C} = \mathcal{C}(\mathbf{p}_\mathcal{S})$. The dilator operation, or successive inversions in two concentric spheres for a dilation by factor $r_2^2 \,/\, r_1^2$, can *isotropically dilate* CGA entities in space *and time*. The translator operation, or successive reflections in parallel spacetime planes, can translate CGA entities in space and time. The rotor operation, or successive reflections in non-parallel spatial planes, can rotate CGA entities in space, leaving the time unaffected. Boosting the CSTA entities does not work as may be expected since the contraction of a sphere into an ellipsoid is not supported by CSTA. CSTA has no ellipsoid entity to represent a contracted sphere. In DCSTA, the DCSTA GIPNS 2-vector quadric surface entities can be boosted as expected.

The CSTA GOPNS entities represent hypersurfaces of the CGA surfaces. The hyperdimension is time w. A sphere changes radius with time as a hypercone or hyperpseudosphere in spacetime. A circle changes radius with time as a pseudosphere in spacetime. A plane is the intersection of two hyperplanes in spacetime. A line is the intersection of three hyperplanes.

The GOPNS entities are called *dual* to the *undual* GIPNS entities, but this naming is quite often reversed in other literature. This naming is chosen to be consistent with DCSTA entities, where the DCSTA GIPNS entities are unduals and the DCSTA GOPNS entities are duals.

4.5.1 Geometric outer product null space (GOPNS)

Geometric outer product null space (GOPNS) entities are introduced by PERWASS in [11], and are reviewed by this author in [6] and [7].

The $\mathcal{G}_{2,4}$ CSTA unit pseudoscalar $\mathbf{I}_\mathcal{C}$ is grade 6. Therefore, CSTA GOPNS entities can be formed as the wedge of *five or less* CSTA points. In $\mathcal{G}_{4,1}$ CGA, the CGA GOPNS entities are formed as the wedge of four or less points. Compared to CGA, CSTA has a larger set of GOPNS entities.

The subset of CSTA GOPNS entities that are similar to CGA GOPNS entities are defined as the *wedge*, or outer product, of four or less CSTA spatial points $\mathbf{P}_{\mathcal{C}_i} = \mathcal{C}(\mathbf{p}_\mathcal{S})$ that are on the surface and that also *span the surface* of the entity.

The wedge of five CSTA points creates hyperhyperboloid and hyperplane entities as explained in later subsections.

4.5.2 CSTA GOPNS 1-vector point

As a GOPNS entity, the CSTA null 1-vector *point embedding* $\boldsymbol{P}_\mathcal{C} = \mathcal{C}(\boldsymbol{p}_\mathcal{M})$ represents the *point* of the embedded STA position $\boldsymbol{p}_\mathcal{M}$. The GOPNS test

$$\boldsymbol{T}_\mathcal{C} \wedge \boldsymbol{P}_\mathcal{C} \;=\; 0$$

holds good *if and only if* (iff)

$$\boldsymbol{T}_\mathcal{C} \;\equiv\; \boldsymbol{P}_\mathcal{C}.$$

As a GIPNS entity, a point $\boldsymbol{P}_\mathcal{C}$ represents a null *hypercone* in spacetime (§4.2.4). The GIPNS test

$$\boldsymbol{T}_\mathcal{C} \cdot \boldsymbol{P}_\mathcal{C} \;=\; 0$$

holds good for any point $\boldsymbol{T}_\mathcal{C}$ on the hypercone with vertex $\boldsymbol{P}_\mathcal{C}$. A point $\boldsymbol{T}_\mathcal{C}$ on the hypercone is a point that is located at a diagonal *lightlike* position relative to the vertex $\boldsymbol{P}_\mathcal{C}$. The hypercone is a sphere in space, centered at $\boldsymbol{P}_\mathcal{C}$, with time-varying radius $r = w - p_w = ct - p_w$.

A CSTA null 1-vector *point embedding* $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$ represents

$$\boldsymbol{T}_{\mathcal{C}} \circ \boldsymbol{P}_{\mathcal{C}} \;=\; \begin{cases} \text{null } \textit{hypercone} \text{ centered at vertex } \boldsymbol{p}_{\mathcal{M}} & : \circ \text{ is } \cdot \\ \text{null } \textit{point} \text{ at } \boldsymbol{p}_{\mathcal{M}}. & : \circ \text{ is } \wedge \end{cases}$$

4.5.3 CSTA GOPNS 2-vector point pair

The CSTA GOPNS 2-vector *point pair* $\mathbf{2}^*_{\mathcal{C}}$ is the wedge of two *not relatively lightlike and finite* CSTA points

$$\begin{aligned} \mathbf{2}^*_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \\ &= \mathbf{2}_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}^{-1} \end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 4-vector *point pair* $\mathbf{2}_{\mathcal{C}}$. The GOPNS test

$$\boldsymbol{T}_{\mathcal{C}} \cdot \mathbf{2}^*_{\mathcal{C}} \;=\; 0$$

holds good for two *not relatively lightlike and finite* points *if and only if* (iff)

$$\boldsymbol{T}_{\mathcal{C}} \;\in\; \{\boldsymbol{P}_{\mathcal{C}_1}, \boldsymbol{P}_{\mathcal{C}_2}\}.$$

A valid *point pair* $\mathbf{2}^*_{\mathcal{C}}$ represents the two distinct points as a single entity.

The *point pair decomposition* [3]

$$\boldsymbol{P}_{\mathcal{C}_\pm} \;=\; \frac{\mathbf{2}^*_{\mathcal{C}} \mp \sqrt{(\mathbf{2}^*_{\mathcal{C}})^2}}{-\mathbf{e}_{\infty\gamma} \cdot \mathbf{2}^*_{\mathcal{C}}} = (\mathbf{2}^*_{\mathcal{C}} \mp \sqrt{\mathbf{2}^*_{\mathcal{C}} \cdot \mathbf{2}^*_{\mathcal{C}}})(-\mathbf{e}_{\infty\gamma} \cdot \mathbf{2}^*_{\mathcal{C}})^{-1}$$

gives the two finite points $\boldsymbol{P}_{\mathcal{C}_+}$ and $\boldsymbol{P}_{\mathcal{C}_-}$ of the point pair with unit weight.

A *light-line (null line)* $\mathcal{L}^*_{\mathcal{C}} = \boldsymbol{P}_{\mathcal{L}_1} \wedge \boldsymbol{P}_{\mathcal{L}_2}$ (§4.5.4) is the wedge of two *relatively lightlike points* $\boldsymbol{P}_{\mathcal{L}_i}$ and represents the line of the two points, excluding $\mathbf{e}_{\infty\gamma}$. A *flat point* $\mathbb{P}^*_{\mathcal{C}} = \boldsymbol{P}_{\mathcal{C}} \wedge \mathbf{e}_{\infty\gamma}$ (§4.5.5) is the wedge of one finite point $\boldsymbol{P}_{\mathcal{C}}$ and the point at infinity $\mathbf{e}_{\infty\gamma}$.

4.5.4 CSTA GOPNS 2-vector light-line (null line)

An STA null *lightlike position* vector $\boldsymbol{l}_{\mathcal{M}}$ relative to the origin has the form

$$\boldsymbol{l}_{\mathcal{M}} \;=\; ct(\boldsymbol{\gamma}_0 + \hat{\mathbf{n}}_{\mathcal{S}}) = w(\boldsymbol{\gamma}_0 + \hat{\mathbf{n}}_{\mathcal{S}}) = ct\boldsymbol{n}_{\mathcal{M}}.$$

It can be verified that $\boldsymbol{n}^2_{\mathcal{M}} = 0$ for any spatial unit direction vector $\mathbf{n}_{\mathcal{S}}$. A lightlike position relative to an STA position vector $\boldsymbol{p}_{\mathcal{M}}$ is

$$\boldsymbol{p}_{\mathcal{L}} \;=\; \boldsymbol{p}_{\mathcal{M}} + \boldsymbol{l}_{\mathcal{M}}.$$

Let any three relatively lightlike positions and their embeddings be

$$\begin{aligned} \boldsymbol{P}_{\mathcal{L}_1} = \mathcal{C}(\boldsymbol{p}_{\mathcal{L}_1}) &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + w_1\boldsymbol{n}_{\mathcal{M}}) = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + ct_1\boldsymbol{n}_{\mathcal{M}}) \\ \boldsymbol{P}_{\mathcal{L}_2} = \mathcal{C}(\boldsymbol{p}_{\mathcal{L}_2}) &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + w_2\boldsymbol{n}_{\mathcal{M}}) = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + ct_2\boldsymbol{n}_{\mathcal{M}}) \\ \boldsymbol{P}_{\mathcal{L}_3} = \mathcal{C}(\boldsymbol{p}_{\mathcal{L}_3}) &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + w_3\boldsymbol{n}_{\mathcal{M}}) = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + ct_3\boldsymbol{n}_{\mathcal{M}}). \end{aligned}$$

These three points, called *relatively lightlike* points, are along a light-line in the direction $\boldsymbol{n}_{\mathcal{M}}$ on a light-cone with vertex $\boldsymbol{p}_{\mathcal{M}}$. It can be verified that for any three such points

$$\boldsymbol{P}_{\mathcal{L}_1} \wedge \boldsymbol{P}_{\mathcal{L}_2} \wedge \boldsymbol{P}_{\mathcal{L}_3} \;=\; 0.$$

Therefore, the *light-line* $\mathcal{L}_\mathcal{C}$ in the direction of $\boldsymbol{n}_\mathcal{M}$ through the point $\boldsymbol{p}_\mathcal{M}$ is characterized by the wedge of any two points on the light-line. The point at infinity $\mathbf{e}_{\infty\gamma}$ is *not* a point on a light-line that is represented like a point pair.

The CSTA GOPNS null 2-vector *light-line* $\mathcal{L}_\mathcal{C}$ is the wedge of any two *relatively light-like* points on the light-line

$$\begin{aligned}\mathcal{L}^*_\mathcal{C} &= \boldsymbol{P}_{\mathcal{L}_1} \wedge \boldsymbol{P}_{\mathcal{L}_2} \\ &\simeq \mathcal{L}_\mathcal{C}\mathbf{I}_\mathcal{C}^{-1}\end{aligned}$$

and is the CSTA dual of the CSTA null 4-vector *light-line* $\mathcal{L}_\mathcal{C}$ up to a homogeneous scalar factor.

The light-line $\mathcal{L}^*_\mathcal{C}$ does *not* include the point at infinity $\mathbf{e}_{\infty\gamma}$. A light-line exists only in spacetime. In general, the two points of $\mathcal{L}_\mathcal{C}$ are along a light-line, which is a line through spacetime with slope $m = \pm 1$ of time to space distance on a light-cone. A light-line is also called a *null line* since

$$(\mathcal{L}^*_\mathcal{C})^2 = 0.$$

For any two *coplanar light-lines* $\mathcal{L}^*_{\mathcal{C}_1}$ and $\mathcal{L}^*_{\mathcal{C}_2}$, the lines share a light-cone vertex $\boldsymbol{p}_\mathcal{M}$ and

$$\mathcal{L}^*_{\mathcal{C}_1} \wedge \mathcal{L}^*_{\mathcal{C}_2} = 0$$

which is a result that holds in general for all coplanar lines.

A *light-line* $\mathcal{L}^*_\mathcal{C}$ is a special type of line that requires only two points to define the line. A light-line is also called a *lightlike line*. Other lines in spacetime are *timelike* or *spacelike* lines and require the wedge of three collinear points to define them. The CSTA GOPNS 3-vector line $\boldsymbol{L}^*_\mathcal{C}$ (§4.5.6) always includes $\mathbf{e}_{\infty\gamma}$ on the line. A light-line $\mathcal{L}^*_\mathcal{C}$ can be extended to include $\mathbf{e}_{\infty\gamma}$ as a 3-vector line $\boldsymbol{L}^*_\mathcal{C}$.

A *lightlike line* $\mathcal{L}^*_\mathcal{C}$ is converted into a *line* $\boldsymbol{L}^*_\mathcal{C}$ as

$$\boldsymbol{L}^*_\mathcal{C} = \mathcal{L}^*_\mathcal{C} \wedge \mathbf{e}_{\infty\gamma}.$$

The CSTA GOPNS 3-vector *line* $\boldsymbol{L}^*_\mathcal{C}$ always includes $\mathbf{e}_{\infty\gamma}$ as a point of $\boldsymbol{L}^*_\mathcal{C}$.

The two points of a lightlike line cannot be decomposed into two points since the lightlike line is a *null line* that has no inverse.

4.5.5 CSTA GOPNS 2-vector flat point

A CSTA *flat point* $\mathbb{P}^*_\mathcal{C}$ is the wedge of a *finite* CSTA point $\boldsymbol{P}_\mathcal{C}$ and the CSTA *point at infinity* $\mathbf{e}_{\infty\gamma}$

$$\begin{aligned}\mathbb{P}^*_\mathcal{C} &= \boldsymbol{P}_\mathcal{C} \wedge \mathbf{e}_{\infty\gamma} = \mathcal{C}(\boldsymbol{p}_\mathcal{M}) \wedge \mathbf{e}_{\infty\gamma} \\ &\simeq \mathbb{P}_\mathcal{C}\mathbf{I}_\mathcal{C}^{-1}\end{aligned}$$

and equals the CSTA dual of the CSTA 4-vector *flat point* $\mathbb{P}_\mathcal{C}$ up to a homogeneous scalar factor.

As introduced in [3] in the context of $\mathcal{G}_{4,1}$ CGA, a flat point is the intersection point of a plane and line in space. However, a plane and line both also include the point at infinity. Therefore, a flat point represents the *two points* where a line and plane intersect in space. In $\mathcal{G}_{2,4}$ CSTA, a line and plane are in spacetime and may intersect at a spacetime flat point. The CSTA GIPNS 2-vector *plane* $\boldsymbol{\Pi}_\mathcal{C}$ is the intersection of two hyperplanes

$$\boldsymbol{\Pi}_\mathcal{C} = \boldsymbol{E}_{\mathcal{C}_1} \wedge \boldsymbol{E}_{\mathcal{C}_2}$$

and the CSTA GIPNS 3-vector *line* $\boldsymbol{L}_{\mathcal{C}}$ is the intersection of three hyperplanes

$$\boldsymbol{L}_{\mathcal{C}} \;=\; \boldsymbol{E}_{\mathcal{C}_3} \wedge \boldsymbol{E}_{\mathcal{C}_4} \wedge \boldsymbol{E}_{\mathcal{C}_5}.$$

In CGA, the intersection of a line and plane is simply $\mathbf{L} \wedge \mathbf{\Pi}$, but this form cannot work as simply in CSTA. There can be *zero, one, or two* hyperplanes that are the same in the line and plane. If *zero* are the same, then $\boldsymbol{\Pi}_{\mathcal{C}} \wedge \boldsymbol{L}_{\mathcal{C}} \neq 0$ and the intersection is $\boldsymbol{\Pi}_{\mathcal{C}} \wedge \boldsymbol{L}_{\mathcal{C}} \simeq \mathbf{e}^{\star}_{\infty\gamma}$. If *two* are the same, then $\boldsymbol{L}_{\mathcal{C}} = \boldsymbol{\Pi}_{\mathcal{C}} \wedge \boldsymbol{E}_{\mathcal{C}}$ and the intersection is $\boldsymbol{L}_{\mathcal{C}}$. If *one* hyperplane is the same, then the intersection is a finite spacetime point $\boldsymbol{P}_{\mathcal{C}}$ and $\mathbf{e}_{\infty\gamma}$, which are represented as a CSTA GOPNS 2-vector *flat point* $\mathbb{P}^{*}_{\mathcal{C}} = \boldsymbol{P}_{\mathcal{C}} \wedge \mathbf{e}_{\infty\gamma}$. In all three cases, a line and plane intersect at $\mathbf{e}_{\infty\gamma}$.

Assume, for now, that there is only *one* common hyperplane

$$\boldsymbol{E}_{\mathcal{C}} = \boldsymbol{E}_{\mathcal{C}_1} = \boldsymbol{E}_{\mathcal{C}_3} \;=\; \boldsymbol{n}^{\dagger}_{\mathcal{M}} + d\mathbf{e}_{\infty\gamma}.$$

We expect to obtain a CSTA GIPNS 4-vector *flat point* $\mathbb{P}_{\mathcal{C}}$ as the intersection of the CSTA GIPNS 3-vector *line* $\boldsymbol{L}_{\mathcal{C}}$ and CSTA GIPNS 2-vector *plane* $\boldsymbol{\Pi}_{\mathcal{C}}$. If we contract $\boldsymbol{E}_{\mathcal{C}}$ into the line or the plane and then wedge them, then we get the 4-vector flat point. The pseudoscalar of the spacetime projections of $\boldsymbol{\Pi}_{\mathcal{C}}$ and $\boldsymbol{L}_{\mathcal{C}}$ is $\mathbf{I}_{\mathcal{M}}$. The conjugate normal vector $\boldsymbol{n}^{\dagger}_{\mathcal{M}}$ of $\boldsymbol{E}_{\mathcal{C}}$ is given by the spacetime *meet* product $\vee_{\mathcal{M}}$ of the plane and line, and the normal vector $\boldsymbol{n}_{\mathcal{M}}$ is

$$\begin{aligned}\boldsymbol{n}_{\mathcal{M}} \;&=\; (\boldsymbol{\Pi}_{\mathcal{C}} \vee_{\mathcal{M}} \boldsymbol{L}_{\mathcal{C}})^{\dagger} \\ &=\; \boldsymbol{\gamma}_0(((\boldsymbol{\Pi}_{\mathcal{C}} \cdot \mathbf{I}^{-1}_{\mathcal{M}}) \wedge (\boldsymbol{L}_{\mathcal{C}} \cdot \mathbf{I}^{-1}_{\mathcal{M}})) \cdot \mathbf{I}_{\mathcal{M}})\boldsymbol{\gamma}_0.\end{aligned}$$

The normal vector $\boldsymbol{n}_{\mathcal{M}}$ can be used to contract $\boldsymbol{E}_{\mathcal{C}}$ in either the plane or line, which then allows intersections of the plane and line to be formed as the two flat points

$$\begin{aligned}\mathbb{P}_{\mathcal{C}_1} \;&=\; (\boldsymbol{n}_{\mathcal{M}} \cdot \boldsymbol{\Pi}_{\mathcal{C}}) \wedge \boldsymbol{L}_{\mathcal{C}} \\ \mathbb{P}_{\mathcal{C}_2} \;&=\; \boldsymbol{\Pi}_{\mathcal{C}} \wedge (\boldsymbol{n}_{\mathcal{M}} \cdot \boldsymbol{L}_{\mathcal{C}}).\end{aligned}$$

For *one* common hyperplane $\boldsymbol{E}_{\mathcal{C}}$, as assumed, then $\mathbb{P}_{\mathcal{C}_1} = \pm\mathbb{P}_{\mathcal{C}_2}$. Now, if *two* hyperplanes are common in the plane and line, then the spacetime meet produces zero and the flat points are zero. These results allow the following definition for the intersection $\boldsymbol{\Pi}_{\mathcal{C}} \cap \boldsymbol{L}_{\mathcal{C}}$ of a line and plane.

The CSTA GIPNS intersection $\boldsymbol{\Pi}_{\mathcal{C}} \cap \boldsymbol{L}_{\mathcal{C}}$ of a CSTA GIPNS 2-vector *plane* $\boldsymbol{\Pi}_{\mathcal{C}}$ and CSTA GIPNS 3-vector *line* $\boldsymbol{L}_{\mathcal{C}}$ can be defined as

$$\boldsymbol{\Pi}_{\mathcal{C}} \cap \boldsymbol{L}_{\mathcal{C}} = \begin{cases} \mathbf{e}_{\infty\gamma} \text{ or } \mathbf{e}^{\star}_{\infty\gamma} \;:\; \boldsymbol{\Pi}_{\mathcal{C}} \wedge \boldsymbol{L}_{\mathcal{C}} \neq 0 \\ \mathbb{P}_{\mathcal{C}_1} = \pm\mathbb{P}_{\mathcal{C}_2} \;:\; \boldsymbol{\Pi}_{\mathcal{C}} \wedge \boldsymbol{L}_{\mathcal{C}} = 0, \boldsymbol{\Pi}_{\mathcal{C}} \vee_{\mathcal{M}} \boldsymbol{L}_{\mathcal{C}} \neq 0 \\ \boldsymbol{L}_{\mathcal{C}} \;:\; \boldsymbol{\Pi}_{\mathcal{C}} \wedge \boldsymbol{L}_{\mathcal{C}} = 0, \boldsymbol{\Pi}_{\mathcal{C}} \vee_{\mathcal{M}} \boldsymbol{L}_{\mathcal{C}} = 0. \end{cases}$$

The point $\boldsymbol{P}_{\mathcal{C}}$ of a flat point $\mathbb{P}^{*}_{\mathcal{C}} = \mathbb{P}_{\mathcal{C}}\mathbf{I}^{-1}_{\mathcal{C}}$ is projected [3] as

$$\boldsymbol{p}_{\mathcal{M}} = \mathcal{C}^{-1}(\boldsymbol{P}_{\mathcal{C}}) \;=\; \frac{(\mathbf{e}_{o\gamma} \wedge \mathbf{e}_{\infty\gamma}) \cdot (\mathbf{e}_{o\gamma} \wedge \mathbb{P}^{*}_{\mathcal{C}})}{-(\mathbf{e}_{o\gamma} \wedge \mathbf{e}_{\infty\gamma}) \cdot \mathbb{P}^{*}_{\mathcal{C}}} = \frac{-\mathbb{P}^{*}_{\mathcal{C}}}{(\mathbf{e}_{o\gamma} \wedge \mathbf{e}_{\infty\gamma}) \cdot \mathbb{P}^{*}_{\mathcal{C}}} \cdot \mathbf{e}_{o\gamma} - \mathbf{e}_{o\gamma}.$$

4.5.6 CSTA GOPNS 3-vector line

The CSTA GOPNS line $\boldsymbol{L}^{*}_{\mathcal{C}}$ is similar to the CGA GOPNS line $\mathbf{L}^{*}$ discussed in [6]. In general, any line in spacetime can be represented as the wedge of three well-chosen points on the line. A CSTA GOPNS *2-vector lightlike line* $\mathcal{L}^{*}_{\mathcal{C}}$ (§4.5.4) is represented by the wedge of just two points but it does not include the point at infinity $\mathbf{e}_{\infty\gamma}$ on the line.

A CSTA GOPNS 3-vector **lightlike** *line* $\boldsymbol{L}^*_{\mathcal{C}}$ is the wedge of *any two relatively lightlike points* $\boldsymbol{P}_{\mathcal{L}_i}$ on the line and the CSTA point at infinity $\mathbf{e}_{\infty\gamma}$

$$\boldsymbol{L}^*_{\mathcal{C}} \;=\; \boldsymbol{P}_{\mathcal{L}_1} \wedge \boldsymbol{P}_{\mathcal{L}_2} \wedge \mathbf{e}_{\infty\gamma} = \mathcal{L}^*_{\mathcal{C}} \wedge \mathbf{e}_{\infty\gamma}.$$

A CSTA GOPNS 3-vector **timelike or spacelike** *line* $\boldsymbol{L}^*_{\mathcal{C}}$ can be the wedge of *any two points* $\boldsymbol{P}_{\mathcal{C}_i}$ on the line and the CSTA point at infinity $\mathbf{e}_{\infty\gamma}$ *or* the wedge of *any three collinear points* $\boldsymbol{P}_{\mathcal{C}_i}$ on the line

$$\begin{aligned}\boldsymbol{L}^*_{\mathcal{C}} \;&=\; \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \mathbf{e}_{\infty\gamma} \\ &\simeq\; \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \\ &\simeq\; \boldsymbol{L}_{\mathcal{C}} \mathbf{I}^{-1}_{\mathcal{C}}\end{aligned}$$

and is equal to the CSTA dual of the CSTA GIPNS 3-vector *line* $\boldsymbol{L}_{\mathcal{C}}$ up to a homogeneous scalar factor.

4.5.7 CSTA GOPNS 3-vector spatial circle

The system of implicit surface equations for a *spatial circle* with radius r_0 centered at (p_x, p_y, p_z) in the xy-plane at $z = p_z$ is

$$\begin{aligned}(x-p_x)^2 + (y-p_y)^2 - r_0^2 \;&=\; 0 \\ z - p_z \;&=\; 0 \\ w - p_w \;&=\; 0.\end{aligned}$$

The CSTA spatial circle entity $\boldsymbol{C}_{\mathcal{C}}$ represents a system of implicit surface equations of this form for the intersection of a circular cylinder and plane. The center position of the circle

$$\boldsymbol{p}_{\mathcal{M}} \;=\; p_w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3$$

includes a time component $p_w\boldsymbol{\gamma}_0$ that indicates *when* the circle exists.

The CSTA GOPNS spatial circle $\boldsymbol{C}^*_{\mathcal{C}}$ is similar to the CGA GOPNS circle $\mathbf{C}^*$ discussed in [6] and is the wedge of *any three points* on the circle in space at the same time. Three points are always *coplanar cocircular* points. Three *collinear* points are on a circle of infinite radius, which is a line.

The CSTA GOPNS 3-vector *spatial circle* $\boldsymbol{C}^*_{\mathcal{C}}$ is the wedge of *any three* CSTA *points* $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(p_w\boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}_i})$ *at the same time* p_w on the circle

$$\begin{aligned}\boldsymbol{C}^*_{\mathcal{C}} \;&=\; \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \\ &\simeq\; \boldsymbol{C}_{\mathcal{C}} \mathbf{I}^{-1}_{\mathcal{C}}\end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 3-vector *spatial circle* $\boldsymbol{C}_{\mathcal{C}}$ up to a homogeneous scalar factor.

The wedge of three points that are not all at the same time may produce a *spacetime hyperbola* (§4.5.8). The circle is produced for three points at the same time.

4.5.8 CSTA GOPNS 3-vector spacetime hyperbola (pseudocircle)

The system of implicit surface equations for a *spacetime circular hyperbola* in the xw-plane centered at (p_w, p_x, p_y, p_z), with central radius r, opening up and down the w-axis is

$$\begin{aligned}(x-p_x)^2 + r^2 - (w-p_w)^2 \;&=\; 0 \\ y - p_y \;&=\; 0 \\ z - p_z \;&=\; 0.\end{aligned}$$

The spacetime circular hyperbola can also be called a *pseudocircle*. The CSTA pseudocircle entity represents a system of implicit surface equations of this form. This hyperbola is not general, but circular. To get the expected shape, the points have to be chosen carefully. At $x = p_x$, $w = p_w \pm r$. At $w = p_w + \sqrt{2}r$, $x = p_x \pm r$. The axes may be transposed. Spatial rotations, spacetime translations, and spacetime isotropic dilations permit the pseudocircle to be in any Minkowski space-time plane, at any spacetime center point, and with any central radius. The hyperbola is generally a conic section of a related circular hyperboloid (§4.5.10) cut through a spacetime plane and has lightlike asymptotes. By cutting the related hyperboloid in different spacetime planes, it is possible to get hyperbolas that open up and down the time or space axis. The hyperboloids are w-axis (time-axis) aligned with circles in the xy-planes.

The CSTA GOPNS 3-vector *spacetime circular hyperbola* $\boldsymbol{C}^*_{\mathcal{C}}$ is the wedge of *any three non-collinear* CSTA *spacetime points* $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}_i})$ on the spacetime circular hyperbola

$$\begin{aligned} \boldsymbol{C}^*_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \\ &\simeq \boldsymbol{C}_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}^{-1} \end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 3-vector *spacetime hyperbola* $\boldsymbol{C}_{\mathcal{C}}$ up to a homogeneous scalar factor. Similar to a circle, the point at infinity $\mathbf{e}_{\infty\gamma}$ is *not* a point on the pseudocircle.

The spacetime hyperbola $\boldsymbol{C}^*_{\mathcal{C}}$ becomes a *light-cone light-line pair* when the three non-collinear points are *relatively lightlike* points $\boldsymbol{P}_{\mathcal{L}_i}$ (§4.5.4). The points are relatively lightlike when any two points are relatively lightlike, forming one of the light-lines. The perpendicular line through the third point is the other light-line. The light-cone vertex is the point of intersection of the two light-lines, which could be one of the points.

In general, the wedge of three non-collinear CSTA *spatial points* $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(\mathbf{p}_{\mathcal{S}_i})$ produces a *spatial circle* (§4.5.7) that holds $w = ct = 0$.

4.5.9 CSTA GOPNS 4-vector spatial sphere

The CSTA GOPNS spatial sphere $\boldsymbol{S}^*_{\mathcal{C}}$ is similar to the CGA GOPNS sphere $\mathbf{S}^*$ discussed in [6].

The CSTA GOPNS 4-vector *spatial sphere* $\boldsymbol{S}^*_{\mathcal{C}}$ is the wedge of four CSTA *spatial points* $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(\mathbf{p}_{\mathcal{S}_i})$ on the sphere surface that span the sphere

$$\begin{aligned} \boldsymbol{S}^*_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \wedge \boldsymbol{P}_{\mathcal{C}_4} \\ &\simeq \boldsymbol{S}_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}^{-1} \end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 2-vector *spatial sphere* $\boldsymbol{S}_{\mathcal{C}}$ up to a homogeneous scalar factor. To span the sphere, the points cannot be all coplanar. The spatial sphere $\boldsymbol{S}^*_{\mathcal{C}}$ holds $w = ct = 0$ and is a sphere in space at time $t = 0$.

4.5.10 CSTA GOPNS 4-vector spacetime hyperboloid (pseudosphere)

The implicit quadric surface equation of a *spacetime circular hyperboloid* of one sheet with circular sections in the xy-plane and central radius r is

$$(w - p_w)^2 + r^2 - (x - p_x)^2 - (y - p_y) = 0.$$

The spacetime circular hyperboloid can also be called a *pseudosphere*. Spatial rotations, spacetime translations, and spacetime isotropic dilations permit the pseudosphere to be in any spatial plane, at any spacetime center point, and with any central radius.

The CSTA GOPNS 4-vector *spacetime circular hyperboloid of one sheet (pseudosphere)* $\boldsymbol{S}^*_{\mathcal{C}}$ is the wedge of four CSTA *spacetime points* $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}_i})$ on the surface that span the surface

$$\begin{aligned}\boldsymbol{S}^*_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \wedge \boldsymbol{P}_{\mathcal{C}_4} \\ &\simeq \mathcal{L}^*_{\mathcal{C}_1} \wedge \mathcal{L}^*_{\mathcal{C}_2} \\ &\simeq \boldsymbol{S}_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}^{-1}\end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 2-vector *spacetime circular hyperboloid of one sheet* $\boldsymbol{S}_{\mathcal{C}}$ up to a homogeneous scalar factor. Similar to a sphere, the point at infinity $\mathbf{e}_{\infty\gamma}$ is *not* a point on the pseudosphere. Two *non-coplanar* light-lines $\mathcal{L}^*_{\mathcal{C}_1}$ and $\mathcal{L}^*_{\mathcal{C}_2}$ span a pseudosphere with radius equal to half the distance between the light-lines.

The pseudosphere $\boldsymbol{S}^*_{\mathcal{C}}$ becomes a *light-cone*, also called a null cone, when the four points are *relatively lightlike* points $\boldsymbol{P}_{\mathcal{L}_i}$. The points are relatively lightlike when any three points are relatively lightlike to the fourth point, which is the vertex center point of the light-cone. The wedge of the light-cone vertex and another point is a light-line $\mathcal{L}^*_{\mathcal{C}}$, and the light-cone is spanned by three light-lines sharing the vertex.

In general, the wedge of four non-coplanar CSTA *spatial points* $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(\mathbf{p}_{\mathcal{S}_i})$ produces a *spatial sphere* that holds $w = ct = 0$.

It is also possible to produce the CSTA GOPNS 4-vector *spacetime hyperboloid of two sheets (imaginary pseudosphere)* as the wedge of four well-chosen points that span the surface.

4.5.11 CSTA GOPNS 4-vector plane

The CSTA GOPNS plane $\boldsymbol{\Pi}^*_{\mathcal{C}}$ is similar to the CGA GOPNS plane $\boldsymbol{\Pi}^*$ discussed in [6]. In CGA, a plane $\boldsymbol{\Pi}^*$ is the wedge of *any four coplanar non-collinear non-cocircular points* on the plane. The four well-chosen points that define a plane in CGA are nearly the same for $\boldsymbol{\Pi}^*_{\mathcal{C}}$, but light-lines $\mathcal{L}^*_{\mathcal{C}}$ (§4.5.4) introduce an additional constraint on the choice of the four coplanar points in spacetime.

Three non-collinear finite points $\boldsymbol{P}_{\mathcal{C}_i}$ are co(pseudo)circular and define a finite (pseudo)circle $\boldsymbol{C}^*_{\mathcal{C}}$. The fourth point can be the point at infinity $\mathbf{e}_{\infty\gamma}$ or some other coplanar non-co(pseudo)circular finite point $\boldsymbol{P}_{\mathcal{C}_4}$.

Three collinear, not relatively lightlike, finite points $\boldsymbol{P}_{\mathcal{C}_i}$ define a line $\boldsymbol{L}^*_{\mathcal{C}}$. The fourth point *cannot* be the point at infinity $\mathbf{e}_{\infty\gamma}$ since it is collinear. The fourth point can be some other non-collinear finite point $\boldsymbol{P}_{\mathcal{C}_4}$.

Two relatively lightlike finite points $\boldsymbol{P}_{\mathcal{L}_i}$ define a light-line $\mathcal{L}^*_{\mathcal{C}}$. The other two points can be $\mathbf{e}_{\infty\gamma}$ and a coplanar non-collinear finite point $\boldsymbol{P}_{\mathcal{C}_4}$. The other two points can also be not relatively lightlike finite points $\boldsymbol{P}_{\mathcal{C}_3}$ and $\boldsymbol{P}_{\mathcal{C}_4}$ that are coplanar non-collinear points to $\mathcal{L}^*_{\mathcal{C}}$.

The CSTA GOPNS 4-vector *plane* $\boldsymbol{\Pi}^*_{\mathcal{C}}$ is the wedge of *four well-chosen points* $\boldsymbol{P}_{\mathcal{C}_i}$ on the plane in space or spacetime

$$\begin{aligned}\boldsymbol{\Pi}^*_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \wedge \boldsymbol{P}_{\mathcal{C}_4} = \boldsymbol{C}_{\mathcal{C}} \wedge \boldsymbol{P}_{\mathcal{C}_4} \\ &\simeq \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \wedge \mathbf{e}_{\infty\gamma} = \boldsymbol{C}_{\mathcal{C}} \wedge \mathbf{e}_{\infty\gamma} \\ &\simeq \boldsymbol{L}^*_{\mathcal{C}} \wedge \boldsymbol{P}_{\mathcal{C}_4} \\ &\simeq \mathcal{L}^*_{\mathcal{C}} \wedge \mathbf{e}_{\infty\gamma} \wedge \boldsymbol{P}_{\mathcal{C}_4} \\ &\simeq \mathcal{L}^*_{\mathcal{C}} \wedge \boldsymbol{P}_{\mathcal{C}_3} \wedge \boldsymbol{P}_{\mathcal{C}_4} \\ &\simeq \boldsymbol{\Pi}_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}^{-1}\end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 2-vector *plane* $\boldsymbol{\Pi}_{\mathcal{C}}$ up to a homogeneous scalar factor. The four points must be well-chosen as explained above.

The entity $\mathbf{\Pi}^*_{\mathcal{C}}$ is a *plane in space* that holds $w = ct = 0$ when its points $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(\mathbf{p}_{\mathcal{S}_i})$ are the embeddings of spatial points $\mathbf{p}_{\mathcal{S}_i}$ in 3-D SA space. In the general case of points $\boldsymbol{P}_{\mathcal{C}_i} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}_i})$ in spacetime, the entity $\mathbf{\Pi}^*_{\mathcal{C}}$ is a *plane in spacetime*. The plane entity is generally valid in both space and spacetime.

4.5.12 CSTA GOPNS 5-vector hyperhyperboloid

The implicit quadric surface equation for a circular *hyperhyperboloid of one sheet* is

$$r_0^2 + (w - p_w)^2 - (x - p_x)^2 - (y - p_y)^2 - (z - p_x)^2 \;=\; 0$$

where r_0 is the initial radius of the expanding sphere in space with time-varying radius

$$r \;=\; \sqrt{r_0^2 + (w - p_w)^2} = \sqrt{r_0^2 + (ct - p_w)^2}$$

and center position

$$\boldsymbol{p}_{\mathcal{M}} \;=\; p_w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3$$

in spacetime.

The hyperhyperboloid can be spanned by five surface points that do not form entities for any (pseudo)sphere, plane, line, or (pseudo)circle. Planes and lines are avoided by excluding the point at infinity. Spheres and circles are avoided by using only one or two points in any circle on the surface. The choice of points is otherwise arbitrary. For example, using an arbitrary scalar $l \neq 0$, three values of time

$$w \;\in\; \{p_w + l, p_w + 2l, p_w - 3l\}$$

and corresponding values of radius

$$r \;\in\; \left\{\sqrt{r_0^2 + l^2}, \sqrt{r_0^2 + 4l^2}, \sqrt{r_0^2 + 9l^2}\right\}$$

can be chosen. Then, use at most two surface points per value of w. The hyperhyperboloid, a sphere that expands with time, has the five surface points that span the surface

$$\begin{aligned}
\boldsymbol{P}_{\mathcal{C}_1} &= \mathcal{C}\Big(\boldsymbol{p}_{\mathcal{M}} + l\boldsymbol{\gamma}_0 + \sqrt{r_0^2 + l^2}\boldsymbol{\gamma}_1\Big) \\
\boldsymbol{P}_{\mathcal{C}_2} &= \mathcal{C}\Big(\boldsymbol{p}_{\mathcal{M}} + 2l\boldsymbol{\gamma}_0 - \sqrt{r_0^2 + 4l^2}\boldsymbol{\gamma}_2\Big) \\
\boldsymbol{P}_{\mathcal{C}_3} &= \mathcal{C}\Big(\boldsymbol{p}_{\mathcal{M}} + 2l\boldsymbol{\gamma}_0 - \sqrt{r_0^2 + 4l^2}\boldsymbol{\gamma}_3\Big) \\
\boldsymbol{P}_{\mathcal{C}_4} &= \mathcal{C}\Big(\boldsymbol{p}_{\mathcal{M}} - 3l\boldsymbol{\gamma}_0 + \sqrt{r_0^2 + 9l^2}\boldsymbol{\gamma}_1\Big) \\
\boldsymbol{P}_{\mathcal{C}_5} &= \mathcal{C}\Big(\boldsymbol{p}_{\mathcal{M}} - 3l\boldsymbol{\gamma}_0 + \sqrt{r_0^2 + 9l^2}\boldsymbol{\gamma}_2\Big).
\end{aligned}$$

The CSTA GOPNS 5-vector *hyperhyperboloid of one sheet (hyperpseudosphere)* $\mathbf{\Sigma}^*_{\mathcal{C}}$ is the wedge of five CSTA1 points $\boldsymbol{P}_{\mathcal{C}_i}$ on the surface that span the surface

$$\begin{aligned}
\mathbf{\Sigma}^*_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \wedge \boldsymbol{P}_{\mathcal{C}_4} \wedge \boldsymbol{P}_{\mathcal{C}_5} \\
&\simeq \mathbf{\Sigma}_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}^{-1}
\end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 1-vector *hyperhyperboloid* $\mathbf{\Sigma}_{\mathcal{C}}$ up to a homogeneous scalar factor.

The hyperhyperboloid with $r_0 = 0$ degenerates into the dual CSTA GOPNS null 5-vector *hypercone* $\boldsymbol{P}^*_{\mathcal{C}} = \mathbf{\Sigma}^*_{\mathcal{C}}(\boldsymbol{p}_{\mathcal{M}}, r_0 = 0)$. The CSTA GIPNS null 1-vector hypercone $\boldsymbol{P}_{\mathcal{C}}$ at $\boldsymbol{p}_{\mathcal{M}}$ is the *point embedding* $\boldsymbol{P}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{p}_{\mathcal{M}})$. The undual $\boldsymbol{P}^\star_{\mathcal{C}} = \boldsymbol{P}_{\mathcal{C}}\mathbf{I}_{\mathcal{C}} = \boldsymbol{P}^*_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}\mathbf{I}_{\mathcal{C}} = -\boldsymbol{P}^*_{\mathcal{C}}$ is the CSTA GIPNS null 5-vector *point* $\boldsymbol{P}^\star_{\mathcal{C}}$.

It is also possible to produce the CSTA GOPNS 5-vector *hyperhyperboloid of two sheets (imaginary hyperpseudosphere)* $\boldsymbol{\Xi}^*_{\mathcal{C}}$ as the wedge of five well-chosen points that span the surface.

4.5.13 CSTA GOPNS 5-vector hyperplane

A hyperplane is a subspace of dimension $(n-1)$ in a space of dimension n. In 4-D spacetime, a hyperplane is a 3-D subspace. The signature of the hyperplane space can be $(2,1)$ or $(3,0)$.

The implicit linear surface equation for a hyperplane in spacetime is

$$\begin{aligned} \boldsymbol{t}_{\mathcal{M}} \cdot \boldsymbol{n}^{\dagger}_{\mathcal{M}} - d &= \\ \boldsymbol{t}_{\mathcal{M}} \cdot (\boldsymbol{\gamma}_0 \boldsymbol{n}_{\mathcal{M}} \boldsymbol{\gamma}_0) - d &= \\ \boldsymbol{t}_{\mathcal{M}} \cdot (\boldsymbol{n}_{\mathcal{M}} - 2(\boldsymbol{n}_{\mathcal{M}} \cdot \mathbf{I}_{\mathcal{S}})\mathbf{I}_{\mathcal{S}}^{-1}) - d &= \\ n_w w + n_x x + n_y y + n_z z - d &= 0 \end{aligned}$$

where d is the distance of the hyperplane from the origin in the direction $\boldsymbol{n}_{\mathcal{M}}$, which is the unit-norm STA vector that is perpendicular, or *normal*, to the hyperplane

$$\boldsymbol{n}_{\mathcal{M}} = \frac{\boldsymbol{n}_{\mathcal{M}}}{\|\boldsymbol{n}_{\mathcal{M}}\|} = \frac{\boldsymbol{n}_{\mathcal{M}}}{\sqrt{\boldsymbol{n}_{\mathcal{M}} \cdot \boldsymbol{n}^{\dagger}_{\mathcal{M}}}} = \frac{\boldsymbol{n}_{\mathcal{M}}}{\sqrt{n_w^2 + n_x^2 + n_y^2 + n_z^2}} = n_w \boldsymbol{\gamma}_0 + n_x \boldsymbol{\gamma}_1 + n_y \boldsymbol{\gamma}_2 + n_z \boldsymbol{\gamma}_3.$$

The STA test vector $\boldsymbol{t}_{\mathcal{M}}$ is

$$\boldsymbol{t}_{\mathcal{M}} = w\boldsymbol{\gamma}_0 + x\boldsymbol{\gamma}_1 + y\boldsymbol{\gamma}_2 + z\boldsymbol{\gamma}_3.$$

The position of the hyperplane is

$$\boldsymbol{p}_{\mathcal{M}} = d\boldsymbol{n}_{\mathcal{M}}.$$

The hyperplane can be spanned by the point at infinity $\mathbf{e}_{\infty\gamma}$ and four finite points $\boldsymbol{P}_{\mathcal{C}_i}$ that are perpendicular to $\boldsymbol{n}_{\mathcal{M}}$ and centered relative to the position $d\boldsymbol{n}_{\mathcal{M}}$. Three of the four points can span a plane, and the fourth point must not be coplanar with the other three points. The points are cohyperplanar. The 3-D space of the hyperplane is parallel to the dual space

$$\boldsymbol{n}^*_{\mathcal{M}} = \boldsymbol{n}_{\mathcal{M}} \mathbf{I}^{-1}_{\mathcal{M}}.$$

Four unit vectors in this space can be obtained by contractions and conjugations and then spanned around $\boldsymbol{p}_{\mathcal{M}}$ to obtain the four finite points that span the hyperplane.

For a non-null vector $\boldsymbol{n}_{\mathcal{M}} \in \mathcal{G}^{\varnothing 1}_{2,4}$ [11], where $\boldsymbol{n}^2_{\mathcal{M}} \neq 0$, the hyperplane has the five surface points that span the surface

$$\begin{aligned} \boldsymbol{P}_{\mathcal{C}_1} &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + \boldsymbol{\gamma}_0((\boldsymbol{n}_{\mathcal{M}} \cdot (\boldsymbol{\gamma}_0\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2)) \cdot \boldsymbol{n}^*_{\mathcal{M}})\boldsymbol{\gamma}_0) \\ \boldsymbol{P}_{\mathcal{C}_2} &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + \boldsymbol{\gamma}_0((\boldsymbol{n}_{\mathcal{M}} \cdot (\boldsymbol{\gamma}_1\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3)) \cdot \boldsymbol{n}^*_{\mathcal{M}})\boldsymbol{\gamma}_0) \\ \boldsymbol{P}_{\mathcal{C}_3} &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + \boldsymbol{\gamma}_0((\boldsymbol{n}_{\mathcal{M}} \cdot (\boldsymbol{\gamma}_2\boldsymbol{\gamma}_3\boldsymbol{\gamma}_0)) \cdot \boldsymbol{n}^*_{\mathcal{M}})\boldsymbol{\gamma}_0) \\ \boldsymbol{P}_{\mathcal{C}_4} &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}} + \boldsymbol{\gamma}_0((\boldsymbol{n}_{\mathcal{M}} \cdot (\boldsymbol{\gamma}_3\boldsymbol{\gamma}_0\boldsymbol{\gamma}_1)) \cdot \boldsymbol{n}^*_{\mathcal{M}})\boldsymbol{\gamma}_0) \\ \boldsymbol{P}_{\mathcal{C}_5} &= \mathbf{e}_{\infty\gamma} \end{aligned}$$

For a null vector $\boldsymbol{n}_{\mathcal{M}} \in \mathcal{G}^{\circ 1}_{2,4}$ [11], where $\boldsymbol{n}^2_{\mathcal{M}} = 0$, the points $\boldsymbol{P}_{\mathcal{C}_i}$ as given above do *not* span the surface and their wedge product is zero. A null vector $\boldsymbol{n}_{\mathcal{M}}$ can be used to create the GIPNS entity $\boldsymbol{E}_{\mathcal{C}}$, and its dual GOPNS entity $\boldsymbol{E}^*_{\mathcal{C}} = \boldsymbol{E}_{\mathcal{C}} \mathbf{I}^{-1}_{\mathcal{C}}$ represents the same surface. In general, the implicit surface function

The CSTA GOPNS 5-vector *hyperplane* $\boldsymbol{E}^*_{\mathcal{C}}$ is the wedge the CSTA point at infinity $\mathbf{e}_{\infty\gamma}$ and four CSTA points $\boldsymbol{P}_{\mathcal{C}_i}$ on the surface that span the hyperplane

$$\begin{aligned}\boldsymbol{E}^*_{\mathcal{C}} &= \boldsymbol{P}_{\mathcal{C}_1} \wedge \boldsymbol{P}_{\mathcal{C}_2} \wedge \boldsymbol{P}_{\mathcal{C}_3} \wedge \boldsymbol{P}_{\mathcal{C}_4} \wedge \mathbf{e}_{\infty\gamma} \\ &= \boldsymbol{E}_{\mathcal{C}} \mathbf{I}^{-1}_{\mathcal{C}}\end{aligned}$$

and is the CSTA dual of the CSTA GIPNS 1-vector *hyperplane* $\boldsymbol{E}_{\mathcal{C}}$ up to a homogeneous scalar factor, since the points $\boldsymbol{P}_{\mathcal{C}^1}$ are arbitrary. Straightforward substitutions produce $\boldsymbol{E}^*_{\mathcal{C}^1}$ and $\boldsymbol{E}^*_{\mathcal{C}^2}$ in CSTA1 and CSTA2, respectively.

The symbolic CSTA test point $\boldsymbol{T}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{t}_{\mathcal{M}})$, or an actual point $\boldsymbol{P}_{\mathcal{C}}$, that satisfies the hyperplane implicit surface equation as $\boldsymbol{T}_{\mathcal{C}} \wedge \boldsymbol{E}^*_{\mathcal{C}} = 0$ is a point on the CSTA GOPNS hyperplane $\boldsymbol{E}^*_{\mathcal{C}}$.

4.6 CSTA operations

4.6.1 CSTA dualizaton

The CSTA *dual* $A^{*\mathcal{C}}_{\mathcal{C}}$ of a CSTA multivector $A_{\mathcal{C}}$ is

$$A^{*\mathcal{C}}_{\mathcal{C}} = A_{\mathcal{C}^1} \mathbf{I}^{-1}_{\mathcal{C}^1} = A_{\mathcal{C}^1} \mathbf{I}_{\widetilde{\mathcal{C}}^1}.$$

The CSTA *undual* $A_{\mathcal{C}}$ of a CSTA multivector $A^{*\mathcal{C}}_{\mathcal{C}}$ is

$$A_{\mathcal{C}} = A^{*\mathcal{C}}_{\mathcal{C}} \mathbf{I}_{\mathcal{C}} = A_{\mathcal{C}} \mathbf{I}^{-1}_{\mathcal{C}} \mathbf{I}_{\mathcal{C}}.$$

4.6.2 CSTA spatial projection

The $\mathcal{G}_{1,4}$ CSA1 *spatial projection* $A_{\mathcal{CS}^1}$ of a $\mathcal{G}_{2,4}$ CSTA1 multivector $A_{\mathcal{C}^1}$ is

$$A_{\mathcal{CS}^1} = (A_{\mathcal{C}^1} \cdot \mathbf{I}_{\mathcal{CS}^1}) \mathbf{I}^{-1}_{\mathcal{CS}^1}$$

where the $\mathcal{G}_{1,4}$ Conformal Space-like Algebra 1 (CSA1) *unit pseudoscalar* $\mathbf{I}_{\mathcal{CS}^1}$ is

$$\mathbf{I}_{\mathcal{CS}^1} = \mathbf{e}_1 \cdot \mathbf{I}_{\mathcal{C}^1} = \mathbf{I}_{\mathcal{S}^1} \mathbf{e}_5 \mathbf{e}_6.$$

The $\mathcal{G}_{1,4}$ CSA2 *spatial projection* $A_{\mathcal{CS}^2}$ of a $\mathcal{G}_{2,4}$ CSTA2 multivector $A_{\mathcal{C}^2}$ is

$$A_{\mathcal{CS}^2} = (A_{\mathcal{C}^2} \cdot \mathbf{I}_{\mathcal{CS}^2}) \mathbf{I}^{-1}_{\mathcal{CS}^2}$$

where the $\mathcal{G}_{1,4}$ Conformal Space-like Algebra 2 (CSA2) *unit pseudoscalar* $\mathbf{I}_{\mathcal{CS}^2}$ is

$$\mathbf{I}_{\mathcal{CS}^2} = \mathbf{e}_7 \cdot \mathbf{I}_{\mathcal{C}^2} = \mathbf{I}_{\mathcal{S}^2} \mathbf{e}_{11} \mathbf{e}_{12}.$$

The spatial projections $A_{\mathcal{CS}}$ drop the time-like components of $A_{\mathcal{C}}$, and may be useful for extracting geometrical results in space.

4.6.3 CSTA spatial rotor

The spatial rotor R is the same in SA, STA, and CSTA.

The CSTA *spatial rotor* $R_{\mathcal{C}}$ is equal to the SA *rotor* $R_{\mathcal{S}}$

$$\begin{aligned}R_{\mathcal{C}} &= R_{\mathcal{S}} = e^{\frac{1}{2}\theta \hat{\mathbf{n}}^{*\mathcal{S}}_{\mathcal{S}}} \\ &= \cos\left(\frac{1}{2}\theta\right) + \sin\left(\frac{1}{2}\theta\right) \hat{\mathbf{n}}_{\mathcal{S}} \mathbf{I}_{\widetilde{\mathcal{S}}}\end{aligned}$$

where SA unit direction vector $\hat{\mathbf{n}}_{\mathcal{S}}$ is the axis of rotation, and θ is the angle of rotation. The *rotor operation* $R_{\mathcal{C}} A_{\mathcal{C}} R_{\widetilde{\mathcal{C}}}$ on a CSTA GIPNS entity $A_{\mathcal{C}}$ spatially rotates the entity in the usual way in space, leaving any time component unchanged.

The CSTA *rotor operation* that spatially rotates CSTA entity $A_{\mathcal{C}}$ by angle θ around SA axis $\hat{\mathbf{n}}_{\mathcal{S}}$ is defined as

$$A'_{\mathcal{C}} \;=\; R_{\mathcal{C}} A_{\mathcal{C}} \widetilde{R_{\mathcal{C}}}.$$

4.6.4 CSTA translator

The CSTA *translator* $T_{\mathcal{C}}$, adapted from the CGA translator, is defined as

$$\begin{aligned} T_{\mathcal{C}} &= e^{-\frac{1}{2}\boldsymbol{d}_{\mathcal{M}}\mathbf{e}_{\infty\gamma}} \\ &= 1 - \frac{1}{2}\boldsymbol{d}_{\mathcal{M}} \wedge \mathbf{e}_{\infty\gamma}. \end{aligned}$$

The translation vector $\boldsymbol{d}_{\mathcal{M}}$ is an STA *spacetime displacement* vector in STA1 or STA2.

The CSTA *translator operation* that translates CSTA entity $A_{\mathcal{C}}$ in spacetime by STA spacetime displacement $\boldsymbol{d}_{\mathcal{M}}$ is the two-sided versor "sandwich" operation

$$A'_{\mathcal{C}} \;=\; T_{\mathcal{C}} A_{\mathcal{C}} \widetilde{T_{\mathcal{C}}}.$$

4.6.5 CSTA spatial rotor around a line

The CSTA 2-versor *line rotor* $L_{\mathcal{C}}$ is defined as

$$\begin{aligned} L_{\mathcal{C}} &= e^{-\frac{1}{2}\theta\boldsymbol{\gamma}_0 \cdot \boldsymbol{L}_{\mathcal{C}}} \\ &= \cos\left(\frac{1}{2}\theta\right) + \sin\left(\frac{1}{2}\theta\right)(-\boldsymbol{\gamma}_0 \cdot \boldsymbol{L}_{\mathcal{C}}). \end{aligned}$$

The line rotor $L_{\mathcal{C}}$ rotates around the line $\boldsymbol{L}_{\mathcal{C}}$ in space by the angle θ. The direction of rotation follows the right-hand rule such that if the line is rotated and translated into the z-axis, then the rotation is counter-clockwise around the z-axis by angle θ. If the line has been rotated by π or has been scaled by -1, then the rotation is negative or left-handed.

The time component of the line $\boldsymbol{L}_{\mathcal{C}}$ has no effect on the spatial rotation operation $A'_{\mathcal{C}} = L_{\mathcal{C}} A_{\mathcal{C}} \widetilde{L_{\mathcal{C}}}$ on CSTA entity$A_{\mathcal{C}}$. The time component of $A_{\mathcal{C}}$ is unaffected by the rotation operation into $A'_{\mathcal{C}}$.

4.6.6 CSTA isotropic dilator

The CSTA 2-versor *isotropic dilator* $D_{\mathcal{C}}$, adapted from the CGA dilator, is defined as

$$D_{\mathcal{C}} \;=\; \frac{1}{2}(1+d) + \frac{1}{2}(1-d)\mathbf{e}_{\infty\gamma} \wedge \mathbf{e}_{o\gamma}.$$

The scalar d is the *dilation factor*. The CSTA isotropic dilator $D_{\mathcal{C}}$ is a *spacetime dilator*, which includes the dilation of the time and space components of an entity by the factor d.

The CSTA *isotropic dilation operation* that isotropically dilates CSTA entity $A_{\mathcal{C}}$ by factor d in spacetime is the two-sided versor "sandwich" operation

$$A'_{\mathcal{C}} \;=\; T_{\mathcal{C}} A_{\mathcal{C}} \widetilde{T_{\mathcal{C}}}.$$

4.6.7 CSTA spacetime boost

The STA boost $B_{\mathcal{M}}$ operator can also be applied to CSTA points and entities. The effect of a boost on a CSTA point mirrors the effect on the STA point it embeds. A boosted CSTA point can be projected back to a boosted STA point that can be renormalized. The effect of boosts on other CSTA surface entities will not be considered in this paper. Boosts on DCSTA quadric surface entities will be considered later in this paper.

The CSTA *boost operator* $B_{\mathcal{C}}$ for a natural speed $\beta_{\mathbf{v}}$ in the SA *unit* direction $\hat{\mathbf{v}}_{\mathcal{S}}$ is defined as

$$\begin{aligned} B_{\mathcal{C}} = B_{\mathcal{M}} &= e^{\frac{1}{2}\varphi_{\mathbf{v}}\hat{\mathbf{v}}_{\mathcal{S}}\boldsymbol{\gamma}_0} \\ &= \cosh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right) + \sinh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right)\hat{\mathbf{v}}_{\mathcal{S}} \wedge \boldsymbol{\gamma}_0 \end{aligned}$$

where the boost velocity is

$$\mathbf{v}_{\mathcal{S}} = \|\mathbf{v}_{\mathcal{S}}\|\hat{\mathbf{v}}_{\mathcal{S}} = \beta_{\mathbf{v}} c \hat{\mathbf{v}}_{\mathcal{S}}$$

and the rapidity is

$$\varphi_{\mathbf{v}} = \operatorname{atanh}(\beta_{\mathbf{v}}) = \operatorname{atanh}\left(\frac{\|\mathbf{v}_{\mathcal{S}}\|}{c}\right).$$

The SA velocity $\mathbf{v}_{\mathcal{S}}$ of the CSTA boost $B_{\mathcal{C}} = B_{\mathcal{M}}$ is relative to the STA observer velocity

$$\boldsymbol{o}_{\mathcal{M}} = c\boldsymbol{\gamma}_0$$

and has the STA velocity

$$\boldsymbol{v}_{\mathcal{M}} = \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}}.$$

The boost and renormalization of the STA observer velocity $\boldsymbol{o}_{\mathcal{M}}$ produces a particle velocity

$$\begin{aligned} \boldsymbol{o}'_{\mathcal{M}} = \boldsymbol{v}_{\mathcal{M}} &= c\frac{B_{\mathcal{C}}\boldsymbol{o}_{\mathcal{M}}B_{\mathcal{C}}^{\sim}}{(B_{\mathcal{C}}\boldsymbol{o}_{\mathcal{M}}B_{\mathcal{C}}^{\sim})\cdot\boldsymbol{\gamma}_0} \\ &= \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}} = c\boldsymbol{\gamma}_0 + \mathbf{v}_{\mathcal{S}} \end{aligned}$$

moving relative to the same observer $\boldsymbol{o}_{\mathcal{M}}$.

The STA observer velocity $\boldsymbol{o}_{\mathcal{M}} = c\boldsymbol{\gamma}_0$ embedded as CSTA velocity $\boldsymbol{O}_{\mathcal{C}} = \mathcal{C}(\boldsymbol{o}_{\mathcal{M}})$ is boosted as

$$\boldsymbol{O}'_{\mathcal{C}} = B_{\mathcal{C}}\boldsymbol{O}_{\mathcal{C}}B_{\mathcal{C}}^{\sim}.$$

Then $\boldsymbol{O}'_{\mathcal{C}}$ can be projected and renormalized as the STA particle velocity $\boldsymbol{v}_{\mathcal{M}}$

$$\boldsymbol{o}'_{\mathcal{M}} = c\frac{\mathcal{C}^{-1}(\boldsymbol{O}'_{\mathcal{C}})}{\mathcal{C}^{-1}(\boldsymbol{O}'_{\mathcal{C}})\cdot\boldsymbol{\gamma}_0} = \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}} = \boldsymbol{v}_{\mathcal{M}}.$$

4.6.8 CSTA spacetime reframe (reverse boost)

This operation changes the reference frame on which time, distances, and velocities are measured. Each observer carries a local reference frame, where the observer is at the spatial origin of the frame. Each observer measures time t and coordinates $w = ct$, x, y, and z on its local frame. The time t and other coordinates are generally not shared in common with any other observer and are unique variables to each observer. The time and other coordinates of one observer can be transformed into those of another observer via the Lorentz transformations, which is implemented as hyperbolic rotations in spacetime. The hyperbolic rotations are called boosts. As the previous section discussed, a boost can increase the velocity of a particle consistent with Special Relativity, but without changing the frame or observer. As a type of rotation, the boost can be operated in reverse to change the spacetime basis or frame, which is very similar to how a reverse circular rotation changes the coordinate axes in space.

The CSTA *reframe (reverse boost) operation* on a CSTA velocity

$$\boldsymbol{A}_{\mathcal{M}} \;=\; \mathcal{C}(\boldsymbol{a}_{\mathcal{M}}) = \mathcal{C}(\boldsymbol{o}_{\mathcal{M}} + \mathbf{a}_{\mathcal{S}})$$

is

$$\boldsymbol{A}'_{\mathcal{M}} \;=\; B\widetilde{_{\mathcal{M}}}\boldsymbol{A}_{\mathcal{M}}B_{\mathcal{M}}$$

where $\boldsymbol{A}_{\mathcal{M}}$, which is relative to the initial observer $\boldsymbol{o}_{\mathcal{M}}$, becomes $\boldsymbol{A}'_{\mathcal{M}}$ that is relative to the new observer $\boldsymbol{o}'_{\mathcal{M}}$. The new observer $\boldsymbol{o}'_{\mathcal{M}}$ is moving at natural speed $\beta_{\mathbf{v}}$ in unit direction $\hat{\mathbf{v}}_{\mathcal{S}}$ relative to the initial observer $\boldsymbol{o}_{\mathcal{M}}$. Relative to the initial observer $\boldsymbol{o}_{\mathcal{M}}$, the new observer $\boldsymbol{o}'_{\mathcal{M}}$ was

$$\begin{aligned}\boldsymbol{v}_{\mathcal{M}} &= \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}} = \boldsymbol{o}_{\mathcal{M}} + \beta_{\mathbf{v}} c\hat{\mathbf{v}}_{\mathcal{S}} \\ &= \boldsymbol{o}_{\mathcal{M}} + \frac{\|\mathbf{v}_{\mathcal{S}}\|}{c} c\hat{\mathbf{v}}_{\mathcal{S}}.\end{aligned}$$

If $\boldsymbol{A}_{\mathcal{M}} = \boldsymbol{O}_{\mathcal{M}}$ is the CSTA embedding $\boldsymbol{O}_{\mathcal{M}} = \mathcal{C}(\boldsymbol{o}_{\mathcal{M}})$ of the STA initial observer velocity $\boldsymbol{o}_{\mathcal{M}}$, then

$$\begin{aligned}\boldsymbol{a}'_{\mathcal{M}} &= c\frac{\mathcal{C}^{-1}(\boldsymbol{O}'_{\mathcal{C}})}{\mathcal{C}^{-1}(\boldsymbol{O}'_{\mathcal{C}}) \cdot \boldsymbol{\gamma}_0} \\ &= \boldsymbol{o}'_{\mathcal{M}} - \mathbf{v}_{\mathcal{S}} = \boldsymbol{o}'_{\mathcal{M}} - \beta_{\mathbf{v}} c\hat{\mathbf{v}}_{\mathcal{S}}\end{aligned}$$

where $\boldsymbol{o}'_{\mathcal{M}}$ is the new observer that was called $\boldsymbol{v}_{\mathcal{M}}$ and which is moving relative to the initial observer $\boldsymbol{o}_{\mathcal{M}}$. The new velocity $\boldsymbol{a}'_{\mathcal{M}}$ is the initial observer that is now relative to the new observer $\boldsymbol{o}'_{\mathcal{M}}$. The new position is $\boldsymbol{a}'_{\mathcal{M}}t$, where time t is the proper time measured by the new observer at position $\boldsymbol{o}'_{\mathcal{M}}t = ct\boldsymbol{\gamma}_0$. The current observer, after proper normalization, always has the spacetime velocity form

$$\boldsymbol{o}_{\mathcal{M}} \;=\; c\boldsymbol{\gamma}_0$$

and spacetime position form

$$\boldsymbol{o}_{\mathcal{M}}t \;=\; ct\boldsymbol{\gamma}_0$$

where time t is the current observer's clock time or proper time. The observer is considered to be stationary in its local frame of reference, and the time-like axis $\boldsymbol{\gamma}_0$ is basically the observer's clock, although it holds a distance coordinate $w = ct$ with time t.

The boost, reframe, and observer normalizations are potentially confusing operations. Added to the complexity is that these operations can be performed either directly in STA or on embedded values in CSTA. Using a computer algebra system (CAS) for symbolic calculations, such as the Geometric Algebra Module for Sympy [1], can assist in testing these calculations.

5 Double Conformal Space-Time Algebra (DCSTA)

The $\mathcal{G}_{4,8}$ Double Conformal Space-Time Algebra (DCSTA) is a straightforward extension of the $\mathcal{G}_{8,2}$ Double Conformal / Darboux Cyclide Geometric Algebra (DCGA) that is introduced in the paper [6] and discussed further in the papers [4] and [5].

DCSTA includes many operations on quadric surface entities, including

- Rotation in space
- Translation in spacetime
- Isotropic dilation in spacetime
- Anisotropic dilation (directed length dilation) in space
- Spacetime boost of velocity relative to observer, with length contraction effect
- Spacetime reframe (reverse boost) relative to a new observer
- Intersection with standard entities that are doubled CSTA entities.

The general DCSTA GIPNS 2-vector surface entity $\mathbf{\Omega}$ has the general form of a Darboux cyclide in spacetime, which has degenerate forms that include Dupin cyclides, horned Dupin cyclides, parabolic cyclides, and the quadric surfaces. The DCSTA quadric surfaces support anisotropic length contraction and dilation since these forms can be written in terms of the DCSTA value-extraction elements. On the other hand, the higher-degree surfaces, which include cubic parabolic cyclides and quartic Darboux and Dupin cyclides, do not support anisotropic length contraction and dilation forms. Any DCSTA GIPNS 2-vector surface entity $\mathbf{\Omega}$ represents an implicit surface function in spacetime (w, x, y, z) and supports function differentiation using the differential operations for

- Differentiation with respect to $w = ct$, t, x, y, or z
- Directional derivative with respect to a unit-norm direction $\boldsymbol{n}$ in spacetime.

The DCSTA forms of conic sections can also support the operations for

- Orthographic projection
- Perspective projection

as discussed in the paper [4].

5.1 DCSTA unit pseudoscalar

The DCSTA 12-vector *unit pseudoscalar* $\mathbf{I}_{\mathcal{D}}$ with signature $(+---+-+---+-)$ is

$$\begin{aligned} \mathbf{I}_{\mathcal{D}} &= \mathbf{I}_{\mathcal{C}^1} \wedge \mathbf{I}_{\mathcal{C}^2} = \bigwedge_{i=1}^{12} \mathbf{e}_i \\ &= \mathbf{I}_{\mathcal{D}}^{\sim} = \mathbf{I}_{\mathcal{D}}^{-1} \\ \mathbf{I}_{\mathcal{D}}^2 &= 1. \end{aligned}$$

5.2 DCSTA point

5.2.1 DCSTA point embedding

The STA position vector

$$\boldsymbol{p}_{\mathcal{M}} = w\boldsymbol{\gamma}_0 + p_x\boldsymbol{\gamma}_1 + p_y\boldsymbol{\gamma}_2 + p_z\boldsymbol{\gamma}_3 = p_w\boldsymbol{\gamma}_0 + \mathbf{p}_{\mathcal{S}}$$

is embedded into the DCSTA null 2-vector *point embedding* $\boldsymbol{P}_{\mathcal{D}}$ as

$$\begin{aligned}\boldsymbol{P}_{\mathcal{D}} &= \mathcal{C}(\boldsymbol{p}_{\mathcal{M}^1}) \wedge \mathcal{C}(\boldsymbol{p}_{\mathcal{M}^2}) \\ &= \boldsymbol{P}_{\mathcal{C}^1} \wedge \boldsymbol{P}_{\mathcal{C}^2}\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{p}_{\mathcal{M}^1} &= (\boldsymbol{o}_{\mathcal{M}^1} + \mathbf{v}_{\mathcal{S}^1})t = \boldsymbol{v}_{\mathcal{M}^1}t = p_w\mathbf{e}_1 + p_x\mathbf{e}_2 + p_y\mathbf{e}_3 + p_z\mathbf{e}_4 \\ \boldsymbol{p}_{\mathcal{M}^2} &= (\boldsymbol{o}_{\mathcal{M}^2} + \mathbf{v}_{\mathcal{S}^2})t = \boldsymbol{v}_{\mathcal{M}^2}t = p_w\mathbf{e}_7 + p_x\mathbf{e}_8 + p_y\mathbf{e}_9 + p_z\mathbf{e}_{10}.\end{aligned}$$

The DCSTA null 2-vector *point at the origin* is

$$\mathbf{e}_o = \mathbf{e}_{o1} \wedge \mathbf{e}_{o2}.$$

The DCSTA null 2-vector *point at infinity* is

$$\mathbf{e}_\infty = \mathbf{e}_{\infty 1} \wedge \mathbf{e}_{\infty 2}.$$

The STA vector $\boldsymbol{x}_{\mathcal{M}}$ can be an STA position vector or an STA velocity vector, but it must be handled appropriately in calculations for each type of vector. For spacetime calculations with embedded STA position vectors, then $w = ct$, $\boldsymbol{x}_{\mathcal{M}} = \boldsymbol{v}_{\mathcal{M}}t = (\boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}})t$, and $\boldsymbol{x}_{\mathcal{M}} = 0$ at $t = 0$.

5.2.2 DCSTA point projection (inverse embedding)

The projection of DCSTA point $\boldsymbol{P}_{\mathcal{D}}$ back to STA1 vector $\boldsymbol{p}_{\mathcal{M}^1}$ is

$$\begin{aligned}\boldsymbol{p}_{\mathcal{M}^1} &= \mathcal{C}^{-1}(\boldsymbol{P}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 2}) \\ &= \left(\frac{\boldsymbol{P}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 2}}{-(\boldsymbol{P}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 2}) \cdot \mathbf{e}_{\infty 1}} \cdot \mathbf{I}_{\mathcal{M}^1} \right) \cdot \mathbf{I}_{\mathcal{M}^1}^{-1}.\end{aligned}$$

The projection of DCSTA point $\boldsymbol{P}_{\mathcal{D}}$ back to STA2 vector $\boldsymbol{p}_{\mathcal{M}^2}$ is

$$\begin{aligned}\boldsymbol{p}_{\mathcal{M}^2} &= \mathcal{C}^{-1}(\boldsymbol{P}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 1}) \\ &= \left(\frac{\boldsymbol{P}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 1}}{-(\boldsymbol{P}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 1}) \cdot \mathbf{e}_{\infty 2}} \cdot \mathbf{I}_{\mathcal{M}^2} \right) \cdot \mathbf{I}_{\mathcal{M}^2}^{-1}.\end{aligned}$$

A projected STA vector $\boldsymbol{p}_{\mathcal{M}}$ can be normalized as appropriate for an STA position vector or an STA velocity vector.

5.2.3 DCSTA point value-extraction elements

Let the STA test vector $\boldsymbol{t}$ and its square $\boldsymbol{t}^2$ be

$$\begin{aligned}\boldsymbol{t} &= w\boldsymbol{\gamma}_0 + x\boldsymbol{\gamma}_1 + y\boldsymbol{\gamma}_2 + z\boldsymbol{\gamma}_3 \\ &= ct\boldsymbol{\gamma}_0 + v_x t\boldsymbol{\gamma}_1 + v_y t\boldsymbol{\gamma}_2 + v_z t\boldsymbol{\gamma}_3 \\ &= \boldsymbol{o}_{\mathcal{M}}t + \mathbf{v}_{\mathcal{S}}t = \boldsymbol{o}_{\mathcal{M}}t + \|\mathbf{v}_{\mathcal{S}}\| t\hat{\mathbf{v}}_{\mathcal{S}} = \boldsymbol{o}_{\mathcal{M}}t + vt\hat{\mathbf{v}}_{\mathcal{S}} \\ \boldsymbol{t}^2 &= w^2 - x^2 - y^2 - z^2 \\ &= (ct)^2 - x^2 - y^2 - z^2 \\ &= (ct)^2 - (vt)^2 = t^2(c^2 - v^2) = t^2(c^2 - \beta^2 c^2) \\ &= c^2t^2(1 - \beta^2).\end{aligned}$$

The DCSTA test point $\boldsymbol{T}_{\mathcal{D}} = \mathcal{D}(\boldsymbol{t})$ value-extraction elements T_s are defined as

$$\begin{aligned}
T_w &= \frac{1}{2}(\mathbf{e}_1 \wedge \mathbf{e}_{\infty 2} + \mathbf{e}_{\infty 1} \wedge \mathbf{e}_7) \\
T_t &= \frac{1}{c} T_w \\
T_x &= \frac{1}{2}(\mathbf{e}_{\infty 2} \wedge \mathbf{e}_2 + \mathbf{e}_8 \wedge \mathbf{e}_{\infty 1}) \\
T_y &= \frac{1}{2}(\mathbf{e}_{\infty 2} \wedge \mathbf{e}_3 + \mathbf{e}_9 \wedge \mathbf{e}_{\infty 1}) \\
T_z &= \frac{1}{2}(\mathbf{e}_{\infty 2} \wedge \mathbf{e}_4 + \mathbf{e}_{10} \wedge \mathbf{e}_{\infty 1}) \\
T_{w^2} &= \mathbf{e}_7 \wedge \mathbf{e}_1 \\
T_{t^2} &= \frac{1}{c^2} T_{w^2} \\
T_{x^2} &= \mathbf{e}_8 \wedge \mathbf{e}_2 \\
T_{y^2} &= \mathbf{e}_9 \wedge \mathbf{e}_3 \\
T_{z^2} &= \mathbf{e}_{10} \wedge \mathbf{e}_4 \\
T_{wx} &= \frac{1}{2}(\mathbf{e}_1 \wedge \mathbf{e}_8 + \mathbf{e}_2 \wedge \mathbf{e}_7) \\
T_{wy} &= \frac{1}{2}(\mathbf{e}_1 \wedge \mathbf{e}_9 + \mathbf{e}_3 \wedge \mathbf{e}_7) \\
T_{wz} &= \frac{1}{2}(\mathbf{e}_1 \wedge \mathbf{e}_{10} + \mathbf{e}_4 \wedge \mathbf{e}_7) \\
T_{tx} &= \frac{1}{c} T_{wx} \\
T_{ty} &= \frac{1}{c} T_{wy} \\
T_{tz} &= \frac{1}{c} T_{wz} \\
T_{xy} &= \frac{1}{2}(\mathbf{e}_9 \wedge \mathbf{e}_2 + \mathbf{e}_8 \wedge \mathbf{e}_3) \\
T_{yz} &= \frac{1}{2}(\mathbf{e}_{10} \wedge \mathbf{e}_3 + \mathbf{e}_9 \wedge \mathbf{e}_4) \\
T_{zx} &= \frac{1}{2}(\mathbf{e}_8 \wedge \mathbf{e}_4 + \mathbf{e}_{10} \wedge \mathbf{e}_2) \\
T_{w\boldsymbol{t}^2} &= \mathbf{e}_1 \wedge \mathbf{e}_{o2} + \mathbf{e}_{o1} \wedge \mathbf{e}_7 \\
T_{t\boldsymbol{t}^2} &= \frac{1}{c} T_{w\boldsymbol{t}^2} \\
T_{x\boldsymbol{t}^2} &= \mathbf{e}_{o2} \wedge \mathbf{e}_2 + \mathbf{e}_8 \wedge \mathbf{e}_{o1} \\
T_{y\boldsymbol{t}^2} &= \mathbf{e}_{o2} \wedge \mathbf{e}_3 + \mathbf{e}_9 \wedge \mathbf{e}_{o1} \\
T_{z\boldsymbol{t}^2} &= \mathbf{e}_{o2} \wedge \mathbf{e}_4 + \mathbf{e}_{10} \wedge \mathbf{e}_{o1} \\
T_1 &= -\mathbf{e}_\infty \\
T_{\boldsymbol{t}^2} &= \mathbf{e}_{o2} \wedge \mathbf{e}_{\infty 1} + \mathbf{e}_{\infty 2} \wedge \mathbf{e}_{o1} \\
T_{\boldsymbol{t}^4} &= -4\mathbf{e}_o.
\end{aligned}$$

The value s is extracted from $\boldsymbol{T}_{\mathcal{D}}$ as

$$s = \boldsymbol{T}_{\mathcal{D}} \cdot T_s.$$

The value-extraction elements are used to define DCSTA GIPNS entities that are similar to those that can be defined in DCGA. The DCSTA GIPNS quadric surface entities can be boosted in spacetime and display length contraction.

5.2.4 DCSTA point value-extraction pseudo-inverse elements

The pseudo-inverse of A is denoted A^+ and has the relation

$$A \cdot A^+ \;=\; 1.$$

If A^{-1} exists, it may be equal to A^+. The inverse or pseudo-inverse of an extraction element T_s can be useful for formulating certain other elements and operators. The pseudo-inverses of some of the extraction elements are

$$\begin{aligned}
T_{w^2}^{-1} = T_{w^2}^{+} &= -T_{w^2} \\
T_{t^2}^{-1} = T_{t^2}^{+} &= -c^2 T_{w^2} \\
T_{x^2}^{-1} = T_{x^2}^{+} &= -T_{x^2} \\
T_{y^2}^{-1} = T_{y^2}^{+} &= -T_{y^2} \\
T_{z^2}^{-1} = T_{z^2}^{+} &= -T_{z^2}
\end{aligned}$$

$$\begin{aligned}
T_{w}^{+} &= T_{w\boldsymbol{t}^2} \\
T_{t}^{+} &= c^2 T_{t\boldsymbol{t}^2} \\
T_{x}^{+} &= -T_{x\boldsymbol{t}^2} \\
T_{y}^{+} &= -T_{y\boldsymbol{t}^2} \\
T_{z}^{+} &= -T_{z\boldsymbol{t}^2}
\end{aligned}$$

$$\begin{aligned}
T_{wx}^{+} &= 2T_{wx} \\
T_{wy}^{+} &= 2T_{wy} \\
T_{wz}^{+} &= 2T_{wz} \\
T_{tx}^{+} &= 2c^2 T_{tx} \\
T_{ty}^{+} &= 2c^2 T_{ty} \\
T_{tz}^{+} &= 2c^2 T_{tz} \\
T_{xy}^{+} &= -2T_{xy} \\
T_{yz}^{+} &= -2T_{yz} \\
T_{zx}^{+} &= -2T_{zx}
\end{aligned}$$

$$\begin{aligned}
T_{1}^{+} &= -\frac{1}{4} T_{\boldsymbol{t}^4} \\
T_{\boldsymbol{t}^2}^{+} &= -\frac{1}{2} T_{\boldsymbol{t}^2} \\
T_{\boldsymbol{t}^4}^{+} &= -\frac{1}{4} T_{1}.
\end{aligned}$$

5.3 DCSTA GIPNS standard entities

The *standard* surface entities are similar to the entities available in CGA. These entities have special properties and operations that include reflection and intersection. All DCSTA entities can be reflected in the standard entities. The reflection in a standard sphere is called inversion in a sphere. All DCSTA entities can be intersected with standard entities. A DCSTA *intersection* entity is a wedge of entities, similar to a DCGA intersection entity and with similar limitations on what combinations of entities can be wedged to form a valid intersection entity. The basic examples of intersection entities are the DCSTA 4-vector standard line $\boldsymbol{L}_{\mathcal{D}} = \boldsymbol{\Pi}_{\mathcal{D}_1} \wedge \boldsymbol{\Pi}_{\mathcal{D}_2}$ and DCSTA 4-vector standard circle or pseudocircle $\boldsymbol{C}_{\mathcal{D}} = \boldsymbol{S}_{\mathcal{D}_1} \wedge \boldsymbol{S}_{\mathcal{D}_2}$ or $\boldsymbol{C}_{\mathcal{D}} = \boldsymbol{S}_{\mathcal{D}} \wedge \boldsymbol{\Pi}_{\mathcal{D}}$.

5.3.1 DCSTA GIPNS null 2-vector hypercone

The DCSTA GIPNS null 2-vector *standard hypercone* $\boldsymbol{K}_{\mathcal{D}}$ is defined as

$$\boldsymbol{K}_{\mathcal{D}} \;=\; \boldsymbol{K}_{\mathcal{C}^1} \wedge \boldsymbol{K}_{\mathcal{C}^2} = \boldsymbol{P}_{\mathcal{C}^1} \wedge \boldsymbol{P}_{\mathcal{C}^2}$$

which is the wedge of the same point embedding in CSTA1 and CSTA2.

5.3.2 DCSTA GIPNS 2-vector standard hyperhyperboloid of one sheet

The DCSTA GIPNS 2-vector *standard hyperhyperboloid of one sheet (hyperpseudosphere)* $\boldsymbol{\Sigma}_{\mathcal{D}}$ is defined as

$$\boldsymbol{\Sigma}_{\mathcal{D}} \;=\; \boldsymbol{\Sigma}_{\mathcal{C}^1} \wedge \boldsymbol{\Sigma}_{\mathcal{C}^2}$$

which is the wedge of the same hyperpseudosphere embedded in CSTA1 and CSTA2.

5.3.3 DCSTA GIPNS 2-vector standard hyperhyperboloid of two sheets

The DCSTA GIPNS 2-vector *standard hyperhyperboloid of two sheets (imaginary hyperpseudosphere)* $\boldsymbol{\Xi}_{\mathcal{D}}$ is defined as

$$\boldsymbol{\Xi}_{\mathcal{D}} \;=\; \boldsymbol{\Xi}_{\mathcal{C}^1} \wedge \boldsymbol{\Xi}_{\mathcal{C}^2}$$

which is the wedge of the same imaginary hyperpseudosphere embedded in CSTA1 and CSTA2.

5.3.4 DCSTA GIPNS 4-vector standard sphere or pseudosphere

The DCSTA GIPNS 4-vector *standard sphere or pseudosphere* $\boldsymbol{S}_{\mathcal{D}}$ is defined as

$$\boldsymbol{S}_{\mathcal{D}} \;=\; \boldsymbol{S}_{\mathcal{C}^1} \wedge \boldsymbol{S}_{\mathcal{C}^2}$$

which is the wedge of the same sphere or pseudosphere embedded in CSTA1 and CSTA2.

5.3.5 DCSTA GIPNS 4-vector standard plane

The DCSTA GIPNS 4-vector *standard plane* $\boldsymbol{\Pi}_{\mathcal{D}}$ is defined as

$$\boldsymbol{\Pi}_{\mathcal{D}} \;=\; \boldsymbol{\Pi}_{\mathcal{C}^1} \wedge \boldsymbol{\Pi}_{\mathcal{C}^2}$$

which is the wedge of the same plane embedded in CSTA1 and CSTA2.

5.3.6 DCSTA GIPNS 6-vector standard line

The DCSTA GIPNS 6-vector *standard line* $\boldsymbol{L}_{\mathcal{D}}$ is defined as

$$\boldsymbol{L}_{\mathcal{D}} \;=\; \boldsymbol{L}_{\mathcal{C}^1} \wedge \boldsymbol{L}_{\mathcal{C}^2}$$

which is the wedge of the same line embedded in CSTA1 and CSTA2.

5.3.7 DCSTA GIPNS 6-vector standard circle or pseudocircle

The DCSTA GIPNS 6-vector *standard circle or pseudocircle* $\boldsymbol{C}_{\mathcal{D}}$ is defined as

$$\boldsymbol{C}_{\mathcal{D}} \;=\; \boldsymbol{C}_{\mathcal{C}^1} \wedge \boldsymbol{C}_{\mathcal{C}^2}$$

which is the wedge of the same circle or pseudocircle embedded in CSTA1 and CSTA2.

5.3.8 DCSTA GIPNS 8-vector standard point pair

The DCSTA GIPNS 8-vector *standard point pair* $\mathbf{2}_{\mathcal{D}}$ is defined as

$$\mathbf{2}_{\mathcal{D}} \;=\; \mathbf{2}_{\mathcal{C}^1} \wedge \mathbf{2}_{\mathcal{C}^2}$$

which is the wedge of the same point pair embedded in CSTA1 and CSTA2.

5.3.9 DCSTA GIPNS null 10-vector standard point

The DCSTA GIPNS null 10-vector *standard point* $\boldsymbol{P}^{\star}_{\mathcal{D}}$ is defined as

$$\boldsymbol{P}^{\star}_{\mathcal{D}} \;=\; \boldsymbol{P}^{\star}_{\mathcal{C}^1} \wedge \boldsymbol{P}^{\star}_{\mathcal{C}^2}$$

which is the wedge of the same point embedded in CSTA1 and CSTA2.

5.4 DCSTA GOPNS standard entities

The DCSTA GOPNS *standard entities* are the DCSTA duals of the DCSTA GIPNS *standard entities*. The DCSTA GOPNS *standard entities* can also be formed as the wedges of DCSTA points by the same formulas as in CSTA. Only the DCSTA GOPNS *standard entities* can be formed as wedges of DCSTA null 2-vector *points*.

The DCSTA GIPNS 2-vector *non-standard entities* that are formed as linear combinations of the DCSTA 2-vector *value-extraction elements* T_s have DCSTA dual forms as DCSTA GOPNS 10-vector *non-standard entities*, but these GOPNS entities cannot be formed as wedges of DCSTA points. All DCSTA GIPNS entities have a DCSTA dual that is the DCSTA GOPNS entity.

5.5 DCSTA GIPNS 2-vector non-standard surface entities

The DCSTA GIPNS 2-vector *non-standard surface entities* are defined as linear combinations of the DCSTA 2-vector *value-extraction elements* T_s. In a straightforward way, the entities are formulated in terms of the value-extraction elements to represent implicit surface functions. In terms of the value-extraction elements, these entities are defined exactly as they are in the $\mathcal{G}_{8,2}$ Double Conformal / Darboux Cyclide Geometric Algebra (DCGA) that is introduced in [6]. The reader should refer to [6] for details, which will not be repeated here.

The DCSTA GIPNS 2-vector or GOPNS 10-vector non-standard surface entities can be translated in spacetime using the DCSTA *translator*, spatially rotated in space using the DCSTA spatial *rotor*, and isotropically dilated in spacetime using the DCSTA *isotropic dilator*.

The DCSTA GIPNS 2-vector or GOPNS 10-vector non-standard quadric surface entities can also be boosted in spacetime using the DCSTA boost operation. The quadric surfaces can also be anisotropically dilated in a specific direction in space using the DCSTA boost operation with an imaginary natural boost speed that is followed by a DCSTA spatial projection. The spatial projection discards all time components, but the translator operation can move an entity in spacetime if it is to be at a certain time other than $w = 0$.

5.6 DCSTA conic section entities

It should be straightforward to adapt the DCGA conic sections into DCSTA. The reader is referred to the paper [4] for details on conic sections in DCGA and possible applications that include orthographic and perspective projection of conic sections.

5.7 DCSTA operations

5.7.1 DCSTA spacetime boost

The DCSTA boost operator is defined as

$$B_{\mathcal{D}} \;=\; B_{\mathcal{C}^1} \wedge B_{\mathcal{C}^2}.$$

A DCSTA GIPNS 2-vector quadric surface entity $\mathbf{Q}$ is boosted as

$$\mathbf{Q}' \;=\; B_{\mathcal{D}} \mathbf{Q} B_{\mathcal{D}}^{\sim}.$$

The quadric surface $\mathbf{Q}$ should be initially centered at the spacetime origin before it is boosted the first time. The boosted quadric surface $\mathbf{Q}'$ can be evaluated at any time t and graphed in space. The graph of $\mathbf{Q}'$ should show it to be centered in space at a position consistent with the elapsed time t in the frame of the observer, and the shape of the quadric surface should show a length contraction effect consistent with the speed of the boosting that has been applied.

Other surfaces than quadrics can also be boosted, but they may not display the length contraction effect. The boosting of surfaces other than quadrics could be the subject of further studies to determine their properties and possible applications.

5.7.2 DCSTA spacetime reframe (reverse boost)

The DCSTA reframe operation is simply the application of the boost operation in reverse.

A DCSTA GIPNS 2-vector quadric surface entity $\mathbf{Q}_{\mathcal{D}}$ is reframed as

$$\mathbf{Q}' \;=\; B_{\mathcal{D}}^{\sim} \mathbf{Q} B_{\mathcal{D}}.$$

The quadric surface $\mathbf{Q}$, relative to an initial observer $\boldsymbol{o}_{\mathcal{M}}$, is transformed into $\mathbf{Q}'$ relative to a new observer $\boldsymbol{o}'_{\mathcal{M}}$ that moves with velocity $\mathbf{v}_{\mathcal{S}}$ relative to $\boldsymbol{o}_{\mathcal{M}}$ with spacetime velocity $\boldsymbol{v}_{\mathcal{M}} = \boldsymbol{o}_{\mathcal{M}} + \mathbf{v}_{\mathcal{S}}$. After a renormalization of the new observer, the new observer always has the form

$$\boldsymbol{o}_{\mathcal{M}} \;=\; c\boldsymbol{\gamma}_0$$

in the new reference frame. The reframe of other surfaces than quadrics could be the subject of future research by any interested researcher.

5.7.3 DCSTA spatial projection

In some situations, it is desired to discard time-like components of a DCSTA surface entity, leaving only a geometrical entity at it appears at $t = 0$.

The DCSTA spatial projection of a DCSTA surface entity $\mathbf{Q}$ is defined as

$$\mathbf{Q}_{\mathcal{DS}} = (\mathbf{Q} \cdot \mathbf{I}_{\mathcal{DS}})\mathbf{I}_{\mathcal{DS}}^{-1}$$

where

$$\begin{aligned} \mathbf{I}_{\mathcal{DS}} &= \mathbf{I}_{\mathcal{D}} \cdot (\mathbf{e}_1 \wedge \mathbf{e}_7) \\ &= -\mathbf{I}_{\mathcal{DS}}^{\sim} = -\mathbf{I}_{\mathcal{DS}}^{-1}. \end{aligned}$$

5.7.4 DCSTA dualization

The dual DCSTA GOPNS $(12-k)$-vector surface entity $\mathbf{Q}^{*\mathcal{D}}$ of any DCSTA GIPNS k-vector surface entity $\mathbf{Q}$ is obtained by dualization as

$$\mathbf{Q}^{*\mathcal{D}} = \mathbf{Q}\mathbf{I}_{\mathcal{D}} = \mathbf{Q} \cdot \mathbf{I}_{\mathcal{D}}.$$

The undual operation is

$$\mathbf{Q} = \mathbf{Q}^{*\mathcal{D}} \cdot \mathbf{I}_{\mathcal{D}}.$$

The dual and undual operations are the repeated application of the same dualization operation. Therefore, DCSTA dualization is an involution.

5.7.5 DCSTA rotor

The DCSTA GIPNS 4-versor *rotor* $R_{\mathcal{D}}$ is defined as

$$R_{\mathcal{D}} = R_{\mathcal{C}^1} \wedge R_{\mathcal{C}^2}$$

which is the wedge of the same rotor in CSTA1 and CSTA2.

A DCSTA entity $\mathbf{Q}$ is rotated by the rotor operation

$$\mathbf{Q}' = R_{\mathcal{D}}\mathbf{Q}R_{\mathcal{D}}^{\sim}.$$

5.7.6 DCSTA translator

The DCSTA GIPNS 4-versor *translator* $T_{\mathcal{D}}$ is defined as

$$T_{\mathcal{D}} = T_{\mathcal{C}^1} \wedge T_{\mathcal{C}^2}$$

which is the wedge of the same translator in CSTA1 and CSTA2.

A DCSTA entity $\mathbf{Q}$ is translated by the translator operation

$$\mathbf{Q}' = T_{\mathcal{D}}\mathbf{Q}T_{\mathcal{D}}^{\sim}.$$

5.7.7 DCSTA isotropic dilator

The DCSTA GIPNS 4-versor *isotropic dilator* $D_{\mathcal{D}}$ is defined as

$$D_{\mathcal{D}} = D_{\mathcal{C}^1} \wedge D_{\mathcal{C}^2}$$

which is the wedge of the same isotropic dilator in CSTA1 and CSTA2.

A DCSTA entity $\mathbf{Q}$ is isotropically dilated by the dilator operation

$$\mathbf{Q}' = D_{\mathcal{D}}\mathbf{Q}D_{\mathcal{D}}^{\sim}.$$

5.7.8 DCSTA anisotropic dilator

A DCSTA GIPNS 2-vector *quadric surface* entity $\mathbf{Q}$ can be anisotropically dilated by factor d in a unit direction $\hat{\mathbf{v}}_{\mathcal{S}}$ as a boost that is followed by a DCSTA spatial projection. The natural speed $\beta_{\mathbf{v}}$ of the boost, for the dilation factor d, is

$$\beta_{\mathbf{v}} = \sqrt{1-d^2}$$

which is an imaginary number for $1 < d$.

The DCSTA 4-versor *anisotropic dilator* is defined as

$$\begin{aligned} A &= e^{\frac{1}{2}\varphi_{\mathbf{v}}\hat{\mathbf{v}}_{\mathcal{S}^1}\mathbf{e}_1} \wedge e^{\frac{1}{2}\varphi_{\mathbf{v}}\hat{\mathbf{v}}_{\mathcal{S}^2}\mathbf{e}_7} = B_{\mathcal{C}^1} \wedge B_{\mathcal{C}^2} = B_{\mathcal{D}} \\ &= \left(\cosh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right) + \sinh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right)\hat{\mathbf{v}}_{\mathcal{S}^1}\mathbf{e}_1\right) \wedge \left(\cosh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right) + \sinh\left(\frac{1}{2}\varphi_{\mathbf{v}}\right)\hat{\mathbf{v}}_{\mathcal{S}^2}\mathbf{e}_7\right) \end{aligned}$$

where the rapidity is

$$\varphi_{\mathbf{v}} = \operatorname{atanh}(\beta_{\mathbf{v}}) = \operatorname{atanh}\left(\sqrt{1-d^2}\right).$$

The anisotropic dilator A follows from the Special Relativity (SR) length contraction formula

$$L = \frac{L_0}{\gamma_{\mathbf{v}}} = L_0\sqrt{1-\beta_{\mathbf{v}}^2} = L_0 d.$$

In SR, it is required that $d \leq 1$ for any real speed that is at most light speed c. A dilation factor $1 < d$ requires an imaginary natural speed $\beta_{\mathbf{v}}$.

The DCSTA *anisotropic dilator operation* on a DCSTA GIPNS 2-vector *quadric surface* $\mathbf{Q}$ is defined as

$$\mathbf{Q}'_{\mathcal{DS}} = ((A\mathbf{Q}A^{\sim}) \cdot \mathbf{I}_{\mathcal{DS}})\mathbf{I}_{\mathcal{DS}}^{-1}.$$

The spatial projection discards imaginary time-like components, leaving only a geometrical surface entity at $w = ct = 0$ in spacetime. The dilated entity $\mathbf{Q}'_{\mathcal{DS}}$ can be evaluated with time-like coordinate fixed to $w = ct = 0$, and the entity should appear dilated in the unit direction $\hat{\mathbf{v}}_{\mathcal{S}}$ by factor d. The center point of the entity $\mathbf{Q}$ is also dilated, which causes a translation of the center point unless $\mathbf{Q}$ is at the origin. If $\mathbf{Q}'_{\mathcal{DS}}$ is to be at a time $p_w \neq 0$, then the translator operation can be used to translate $\mathbf{Q}'_{\mathcal{DS}}$ in spacetime.

5.8 DCSTA differential calculus

The DCSTA differential calculus is a straightforward extension of the DCGA differential calculus that is introduced in the paper [5].

5.8.1 DCSTA differential elements

Some of the DCSTA point *value-extraction elements* T_s have inverses. These inverses allow the following DCSTA 2-vector *differential elements* to be defined as

$$\begin{aligned} D_w &= 2T_w T_{w^2}^{-1} \\ D_t &= 2T_t T_{t^2}^{-1} \\ D_x &= 2T_x T_{x^2}^{-1} \\ D_y &= 2T_y T_{y^2}^{-1} \\ D_z &= 2T_z T_{z^2}^{-1}. \end{aligned}$$

5.8.2 DCSTA antisymmetric differential operators

The DCSTA antisymmetric *differential operators* are defined as

$$\begin{aligned}\partial_w = \frac{\partial}{\partial w} &= D_w \times \\ \partial_t = \frac{\partial}{\partial t} &= D_t \times \\ \partial_x = \frac{\partial}{\partial x} &= D_x \times \\ \partial_y = \frac{\partial}{\partial y} &= D_y \times \\ \partial_z = \frac{\partial}{\partial z} &= D_z \times\end{aligned}$$

where the symbol $\times$ is the antisymmetric *commutator product*. For any multivectors A and B, the commutator product is

$$A \times B = \frac{1}{2}(AB - BA).$$

Any DCSTA GIPNS 2-vector surface entity $\mathbf{\Omega}$, in terms of the extraction elements T_s, can be differentiated as $D_{\boldsymbol{n}} \times \mathbf{\Omega}$, where $D_{\boldsymbol{n}}$ is one of the differential elements or is a linear combination of differential elements. Higher-order mixed partial derivatives can also be computed, as for example

$$\frac{d^2\mathbf{\Omega}}{\partial x \partial y} = D_x \times (D_y \times \mathbf{\Omega}) = D_y \times (D_x \times \mathbf{\Omega}).$$

As required of partial differential operators, the sequence in which the derivatives are computed does not affect the result.

5.8.3 DCSTA directional derivative operator

The DCSTA $\boldsymbol{n}$-directional derivative operator is defined as

$$\partial_{\boldsymbol{n}} = \frac{\partial}{\partial \boldsymbol{n}} = (n_w D_w + n_x D_x + n_y D_y + n_z D_z) \times$$

where $\boldsymbol{n}$ is a unit norm spacetime direction

$$\boldsymbol{n} = \frac{\boldsymbol{n}}{\|\boldsymbol{n}\|} = \frac{\boldsymbol{n}}{\sqrt{\boldsymbol{n} \cdot \boldsymbol{n}^\dagger}} = n_w \boldsymbol{\gamma}_0 + n_x \boldsymbol{\gamma}_1 + n_y \boldsymbol{\gamma}_2 + n_z \boldsymbol{\gamma}_3.$$

5.8.4 DCSTA time derivative operator

The DCSTA time t derivative operator is

$$\partial_t = \frac{\partial}{\partial t} = D_t \times .$$

The time t derivative of any DCSTA GIPNS 2-vector *spacetime entity* $\mathbf{\Omega}$ is

$$\dot{\mathbf{\Omega}} = \partial_t \mathbf{\Omega} = \frac{\partial \mathbf{\Omega}}{\partial t} = D_t \times \mathbf{\Omega}.$$

The DCSTA 2-vector *spacetime entity* $\mathbf{\Omega}$ is the most general DCSTA GIPNS 2-vector *non-standard surface entity* that is formed as a linear combination of the DCSTA 2-vector *extraction elements* T_s.

5.9 DCSTA pseudo-integral calculus

In the paper [5], the DCGA pseudo-integral calculus is introduced. A straightforward adaptation and extension into DCSTA is possible but will not be explored here.

6 DCSTA computing with SymPy

DCSTA computing with *SymPy* (http://sympy.org) [13] is possible by using the *Geometric Algebra Module for Sympy* ($\mathcal{GA}$lgebra) by Alan Bromborsky (https://github.com/brombo/galgebra) [1]. This section provides sample code listings and example computations in DCSTA using $\mathcal{GA}$lgebra. The *Anaconda* and *SciPy* python distributions both include SymPy and the *Mayavi* [12] data visualization package. The current version of the $\mathcal{GA}$lgebra module for SymPy can be downloaded and installed from **GitHub**. The *Jupyter Notebook* web application (http://jupyter.org) is recommended for running the sample code and example computations.

6.1 Sample code

The sample code that is listed in the following subsections can be inserted into cells of a Jupyter notebook file and executed in the order listed. The sample code initializes the $\mathcal{GA}$lgebra modules and defines functions and symbols for DCSTA computing. The example computations use the functions and symbols that are defined in the sample code. The sample code is provided as is for experimental testing and educational purposes only!

6.1.1 Imports

Import the SymPy and $\mathcal{GA}$lgebra modules:

```
from sympy import *
from sympy.printing import *
from galgebra.ga import *
from galgebra.mv import *
from galgebra.lt import *
from galgebra.metric import *
from galgebra.printer import *
init_printing()
```

6.1.2 Basis vectors

$\mathcal{G}_{4,8}$ DCSTA requires twelve unit vectors, which can be setup as follows:

```
(e1,e2,e3,e4,e5,e6,e7,e8,e9,e10,e11,e12) = MV.setup(
    'e*1|2|3|4|5|6|7|8|9|10|11|12',
    metric=[1,-1,-1,-1,1,-1, 1,-1,-1,-1,1,-1]
)
```

6.1.3 Points at the origin and at infinity

The CSTA1, CSTA2, and DCSTA points at the origin and at infinity are defined as follows:

```
(eo1,ei1,eo2,ei2,eo,ei) = symbols(
    'e_o1 e_i1 e_o2 e_i2 e_o e_i'
)
# CSTA1 points
eo1 = Pow(2,-1)*(-e5+e6)
ei1 = (e5+e6)
# CSTA2 points
eo2 = Pow(2,-1)*(-e11+e12)
ei2 = (e11+e12)
# DCSTA points
eo = eo1^eo2
ei = ei1^ei2
```

6.1.4 Unit pseudoscalars

The unit pseudoscalars in $\mathcal{G}_{0,3}$ SA1, $\mathcal{G}_{1,3}$ STA1, $\mathcal{G}_{2,4}$ CSTA1, $\mathcal{G}_{0,3}$ SA2, $\mathcal{G}_{1,3}$ STA2, $\mathcal{G}_{2,4}$ CSTA2, and $\mathcal{G}_{4,8}$ DCSTA, respectively, are defined as follows:

```
(I31,I41,I61,I32,I42,I62,ID IDS) = symbols(
    'I_31 I_41 I_61 I_32 I_42 I_62 I_D I_DS'
)
# SA1 unit pseudoscalar
I31 = e2^e3^e4
# STA1 unit pseudoscalar
I41 = e1^I31
# CSTA1 unit pseudoscalar
I61 = I41^ei1^eo1
# SA2 unit pseudoscalar
I32 = e8^e9^e10
# STA2 unit pseudoscalar
I42 = e7^I32
# CSTA2 unit pseudoscalar
I62 = I42^ei2^eo2
# DCSTA unit pseudoscalar
ID = I61^I62
# G2,8 anti-DCGA (spatial) unit pseudoscalar
IDS = (e1^e7)|ID
```

The last value, IDS, is the $\mathcal{G}_{2,8}$ anti-DCGA unit pseudoscalar for a space that is very similar to the $\mathcal{G}_{8,2}$ DCGA. The IDS unit pseudoscalar is used to project entities into a purely spatial algebra that drops the two time dimensions $\mathbf{e}_1$ and $\mathbf{e}_7$. When these time dimensions are dropped, or rejected, by a projection of an entity onto IDS, then the entity is effectively located at $t=0$ in spacetime. The projection onto IDS is useful after a directed scaling, or anisotropic dilation, of a quadric surface.

6.1.5 Point embeddings

CSTA1 point embedding:

```
def EV1(v):
    # Embed STA1 vector v as CSTA1 point.
    v1 = v
    return ( v1 + Pow(2,-1)*(v1*v1)*ei1 + eo1 )
```

CSTA2 point embedding:

```
def EV2(v):
    # Embed STA1 vector v as CSTA2 point.
    # STA1 vector v is converted to an STA2 vector v2.
    v2 = (v|e1)*e7 - ( (v|e2)*e8 + (v|e3)*e9 + (v|e4)*e10 )
    return ( v2 + Pow(2,-1)*(v2*v2)*ei2 + eo2 )
```

DCSTA point embedding:

```
def EV(v):
    # Embed STA1 vector v as DCSTA point.
    return ( EV1(v)^EV2(v) )
```

6.1.6 Point projections

CSTA1 point projection to an STA1 vector:

```
def PV1(V1):
    # Project CSTA1 point to STA1 vector.
    # 1) Normalize point.
    # 2) Use multivector projection to project vector part.
    return Pow(scalar(-V1|ei1),-1)*(V1|I41)*I41.inv()
```

CSTA2 point projection to an STA1 vector:

```
def PV2(V2):
    # Project CSTA2 point to STA1 vector.
    # 1) Normalize point.
    # 2) Use multivector projection to project vector part.
    # 3) Convert into main STA1 space.
    v2 = Pow(scalar(-V2|ei2),-1)*(V2|I42)*I42.inv()
    return ( (v2|e7)*e1 + (-v2|e8)*e2 + (-v2|e9)*e3 + (-v2|e10)*e4 )
```

DCSTA point projection to an STA1 vector:

```
def PV(V):
    # Project DCSTA point V to an STA1 vector.
    # 1) Contract DCSTA point into CSTA1 point using ei2.
    # 2) Project CSTA1 point V1 to an STA1 vector.
    V1 = V|ei2
    return PV1(V1)
```

6.1.7 Symbolic vectors and points

Symbols for coordinates, parameters, and vectors:

```
w,x,y,z,c,t,g,b = symbols('w x y z c t g b')
pw,px,py,pz = symbols('p_w p_x p_y p_z')
rx,ry,rz = symbols('r_x r_y r_z')
nw,nx,ny,nz = symbols('n_w n_x n_y n_z')
vx,vy,vz = symbols('v_x v_y v_z')
v,v1,v2,V,V1,V2 = symbols('v v1 v2 V V1 V2')
```

The `pw,px,py,pz` are used as symbolic position coordinates for the center position of surface entities. The `rx,ry,rz` are used as symbolic radii parameters of implicit quadric surface functions. The `nw,nx,ny,nz` may be used as symbolic coordinates of a normalized unit vector $\mathbf{n}$. The `vx,vy,vz` may be used to hold the velocity components of a velocity vector $\mathbf{v}$. The symbol `c` is used as the symbolic speed of light, and symbol `t` is time.

Symbolic values, vectors, and points:

```
w = c*t
v = w*e1 + x*e2 + y*e3 + z*e4
v1 = v
v2 = w*e7 + x*e8 + y*e9 + z*e10
V1 = EV1(v)
V2 = EV2(v)
V = EV(v)
```

The embedding of the symbolic STA1 and STA2 vectors `v1` and `v2` are symbolic CSTA1 and CSTA2 points `V1` and `V2`, respectively. The DCSTA embedding of a symbolic STA1 vector `v` is the symbolic DCSTA point `V`. In symbolic calculations, these symbolic point embeddings `V1`, `V2`, and `V` are useful to check results.

6.1.8 Extraction elements

The DCSTA point value-extraction elements T_s are used to extract the value s from a DCSTA point $\boldsymbol{T}_{\mathcal{D}}$ as

$$s \;=\; \boldsymbol{T}_{\mathcal{D}} \cdot T_s.$$

The extraction elements are defined in code as:

```
(
    Tw,Tx,Ty,Tz,
    Tww,Txx,Tyy,Tzz,
    Txy,Tyz,Tzx,Twx,Twy,Twz,
    Twt2,Txt2,Tyt2,Tzt2,
    Tt,Ttt,Ttx,Tty,Ttz,Ttt2,
    T0,T1,Tt2,Tt4
) = symbols(
        'Tw Tx Ty Tz '
        'Tww Txx Tyy Tzz '
        'Txy Tyz Tzx Twx Twy Twz '
        'Twt2 Txt2 Tyt2 Tzt2 '
        'Tt Ttt Ttx Tty Ttz Ttt2 '
        'T0 T1 Tt2 Tt4'
)
# Coordinates; linear extractions
Tw = Pow(2,-1)*((e1^ei2)+(ei1^e7))
Tt = Pow(c,-1)*Tw
Tx = -Pow(2,-1)*((e2^ei2)+(ei1^e8))
Ty = -Pow(2,-1)*((e3^ei2)+(ei1^e9))
Tz = -Pow(2,-1)*((e4^ei2)+(ei1^e10))
# Squares; quadratic extractons
Tww = e7^e1
Ttt = Pow(c,-2)*Tw
Txx = e8^e2
Tyy = e9^e3
Tzz = e10^e4
# Cross terms; quadratic extractions
Twx = Pow(2,-1)*((e1^e8)+(e2^e7))
Twy = Pow(2,-1)*((e1^e9)+(e3^e7))
Twz = Pow(2,-1)*((e1^e10)+(e4^e7))
Ttx = Pow(c,-1)*Twx
Tty = Pow(c,-1)*Twy
Ttz = Pow(c,-1)*Twz
Txy = Pow(2,-1)*((e8^e3)+(e9^e2))
Tyz = Pow(2,-1)*((e10^e3)+(e9^e4))
Tzx = Pow(2,-1)*((e10^e2)+(e8^e4))
# Coordinates * squared test vector; cubic extractions
Twt2 = (e1^eo2)+(eo1^e7)
Ttt2 = Pow(c,-1)*Twt2
Txt2 = (eo2^e2)+(e8^eo1)
Tyt2 = (eo2^e3)+(e9^eo1)
Tzt2 = (eo2^e4)+(e10^eo1)
# Unit scalar extraction
T1 = -ei
# Squared test vector; quadratic extraction
Tt2 = (eo2^ei1)+(ei2^eo1)
# Squared squared test vector; quartic extraction
Tt4 = -4*eo
```

6.1.9 Extraction pseudo-inverse elements

An extraction pseudo-inverse element has the property

$$T_s \cdot T_s^+ \;=\; 1.$$

The extraction pseudo-inverse elements are defined in code as:

```
(
    iTw,iTx,iTy,iTz,iTt,
    iTww,iTxx,iTyy,iTzz,iTtt,
    iTxy,iTyz,iTzx,iTwx,iTwy,iTwz,iTtx,iTty,iTtz,
    iT1,iTt2,iTt4 ) = symbols(
    'i_Tww i_Txx i_Tyy i_Tzz i_Ttt '
    'i_Tw i_Tx i_Ty i_Tz iTt '
    'i_Txy i_Tyz i_Tzx i_Twx i_Twy i_Twz i_Ttx i_Tty i_Ttz '
    'i_T1 i_Tt2 i_Tt4'
)
iTw = Twt2
iTx = -Txt2
iTy = -Tyt2
iTz = -Tzt2
iTt = Pow(c,2)*Ttt2
iTww = -Tww
iTxx = -Txx
iTyy = -Tyy
iTzz = -Tzz
iTtt = -Pow(c,2)*Tww
iTxy = -2*Txy
iTyz = -2*Tyz
iTzx = -2*Tzx
iTwx = 2*Twx
iTwy = 2*Twy
iTwz = 2*Twz
iTtx = 2*Pow(c,2)*Ttx
iTty = 2*Pow(c,2)*Tty
iTtz = 2*Pow(c,2)*Ttz
iT1 = -Pow(4,-1)*Tt4
iTt2 = -Pow(2,-1)*Tt2
iTt4 = -Pow(4,-1)*T1
```

6.1.10 Differential elements

The differential and pseudo-integral elements are:

```
(
    Dw,Dx,Dy,Dz,Dt,
    Iw,Ix,Iy,Iz,It
) = symbols(
    'D_w D_x D_y D_z D_t '
    'I_w I_x I_y I_z I_t'
)
# Differential elements
Dw = 2*Tw*Tww.inv()
Dx = 2*Tx*Txx.inv()
Dy = 2*Ty*Tyy.inv()
Dz = 2*Tz*Tzz.inv()
Dt = 2*Tt*Ttt.inv()
# Pseudo-integral elements
Iw = Pow(2,-1)*Tww*iTw
Ix = Pow(2,-1)*Txx*iTx
Iy = Pow(2,-1)*Tyy*iTy
Iz = Pow(2,-1)*Tzz*iTz
It = Pow(2,-1)*Ttt*iTt
```

In recent versions of the $\mathcal{GA}$lgebra module, the commutator product $A \times B$ is coded as `(A>>B)`, and the anti-commutator product $A \bar{\times} B$ is coded as `(A<<B)`. The parentheses are required to ensure that the precedence rules for Python operators do not interfere. For example, the derivative of a DCSTA GIPNS 2-vector surface entity $\mathbf{\Omega}$ with respect to t is written $\partial_t \mathbf{\Omega} = \dot{\mathbf{\Omega}} = D_t \times \mathbf{\Omega}$, and if $\mathbf{\Omega}$ is assigned to variable `E`, then the derivative is coded as `(Dt>>E)` and evaluated symbolically as `(V|(Dt>>E))`. The operation `(A|B)` is the inner product.

6.1.11 Directional derivative operator

The $\boldsymbol{n}$-directional derivative operator is defined in code as:

```
def Dn(w,x,y,z):
    n = sqrt(w**2 + x**2 + y**2 + z**2)
    return Pow(n,-1)*(w*Dw + x*Dx + y*Dy + z*Dz)
```

Only the direction of the spacetime vector

$$\boldsymbol{n} \;=\; w\mathbf{e}_1 + x\mathbf{e}_2 + y\mathbf{e}_3 + z\mathbf{e}_3$$

is significant. The $\boldsymbol{n}$-directional derivative uses the norm-unit of $\boldsymbol{n}$, which is

$$\frac{\boldsymbol{n}}{\|\boldsymbol{n}\|} \;=\; \frac{\boldsymbol{n}}{\sqrt{n_w^2 + n_x^2 + n_y^2 + n_z^2}}.$$

The directional derivative of a DCSTA GIPNS 2-vector surface entity `E` is coded as `(Dn(w,x,y,z)>>E)`.

6.1.12 Pseudo-integral operator

The $\boldsymbol{n}$-directional pseudo-integral operator is defined in code as:

```
def In(w,x,y,z):
    n = sqrt(w**2 + x**2 + y**2 + z**2)
    return Pow(n,-1)*(w*Iw + x*Ix + y*Iy + z*Iz)
```

The directional pseudo-integral of a DCSTA GIPNS 2-vector surface entity `E` is coded as `(In(w,x,y,z)>>E)`.

6.1.13 DCSTA GIPNS 2-vector surface entities

The following subsections define, in code, many of the same surface entities that are discussed in the paper on $\mathcal{G}_{8,2}$ DCGA [6]. The most general DCSTA GIPNS 2-vector surface entity $\boldsymbol{\Omega}$ is the linear combination of the value-extraction elements (§6.1.8). The value-extraction elements can form a general DCSTA GIPNS 2-vector *quadric surface* entity $\mathbf{Q}$ that supports anisotropic dilations. The value-extraction elements can form particular cubic surfaces known as parabolic cyclides and particular quartic surfaces known as Dupin and Darboux cyclides that do not support anisotropic dilations. All of the DCSTA GIPNS 2-vector surfaces $\boldsymbol{\Omega}$ can be boosted into a velocity in spacetime, but only the quadric surface entities can correctly display length contraction or dilation effects.

6.1.14 DCSTA GIPNS 2-vector toroid

The DCSTA GIPNS 2-vector *toroid* is coded as:

```
def GIPNS_Toroid(R,r):
    # Torus centered at the origin circling the z-axis.
    # R is the major radius
    # r is the minor radius
    # R=0 degenerates into exactly -4*Sphere(0,r)
    # R=r=0 degenerates into exactly -4*eo
    # r=0 degenerates into non-standard circle radius R
    # Note, -Tt2 since signatures are negative
    return (
        Tt4 +
        -Tt2*2*(R**2 - r**2) +
        T1*(R**2 - r**2)**2 +
        (Txx + Tyy)*(-4)*R**2
    )
```

The `Toroid` is evaluated at $t=0$ to obtain the same toroid as in $\mathcal{G}_{8,2}$ DCGA:

```
EV(c*0*e1+x*e2+y*e3+z*e4)|Toroid(R,r)
```

The cyclide surface that is formed from the toroid at other times $t\neq 0$ could be researched

further. Most surfaces are evaluated at $t=0$ to obtain a surface similar to those in DCGA.

6.1.15 DCSTA GIPNS 2-vector Dupin cyclide

The DCSTA GIPNS 2-vector *Dupin cyclide* is coded as:

```
def GIPNS_DupinCyclide(R,r1,r2):
    # DupinCyclide generalizes the torus.
    # Types of cyclide:
    #  Ring cyclide when (r1+r2)<2R
    #  Spindle cyclide when (r1+r2)>2R
    # Types of torus:
    #  Horn torus when (r1=r2)=R
    #  Ring torus when (r1=r2)<R
    #  Spindle torus when (r1=r2)>R
    #
    # R is major radius in the xy-plane.
    # r1 and r2 are minor radii.
    # r1 is the radius of sphere centered at x=+R.
    # r2 is the radius of sphere centered at x=-R.
    # When r1=r2, we get exactly a Toroid(R,r=r1=r2).
    # When r1+r2=2R, we get the union of two spheres
    # that touch in a tangent point, exactly.
    #
    # Note: -Tt2 since signatures are negative.
    a = R
    u = (r1+r2)*Pow(2,-1)
    c = (r1-r2)*Pow(2,-1)
    b = sqrt(a**2-c**2)
    return (
        Tt4 +
        -2*(b**2-u**2)*Tt2 +
        (b**2-u**2)**2*T1 +
        -4*(a**2*Txx - 2*a*c*u*Tx + c**2*u**2*T1) +
        -4*b**2*Tyy
    )
```

The `DupinCyclide` is evaluated at $t=0$ to obtain the same Dupin cyclide as in $\mathcal{G}_{8,2}$ DCGA:

```
EV(c*0*e1+x*e2+y*e3+z*e4)|DupinCyclide(R,r1,r2)
```

6.1.16 DCSTA GIPNS 2-vector horned Dupin cyclide

The DCSTA GIPNS 2-vector *horned Dupin cyclide* is coded as:

```
def GIPNS_HornedDupinCyclide(R,r1,r2):
    # Compared to DupinCyclide, just exchange values of
    # u and c to get horned Dupin cyclide.
    # For r1=r2: symmetrical, with horn points on y-axis.
    # For (r1+r2)<2R: horned ring cyclide.
    # For (r1+r2)>2R: horned spindle cyclide.
    # For (r1+r2)=2R: union of two spheres exactly.
    a = R
    u = (r1+r2)*Pow(2,-1)
    c = (r1-r2)*Pow(2,-1)
    b = sqrt(a**2-u**2)
    return (
        Tt4 +
        -2*(b**2-c**2)*Tt2 +
        (b**2-c**2)**2*T1 +
        -4*(a**2*Txx - 2*a*c*u*Tx + c**2*u**2*T1) +
        -4*b**2*Tyy
    )
```

6.1.17 DCSTA GIPNS 2-vector ellipsoid

The DCSTA GIPNS 2-vector *ellipsoid* is coded as:

```
def GIPNS_Ellipsoid(px,py,pz,rx,ry,rz):
    # Axis-aligned ellipsoid.
    return (
        Txx*Pow(rx**2,-1) +
        Tyy*Pow(ry**2,-1) +
        Tzz*Pow(rz**2,-1) +
        -Tx*2*px*Pow(rx**2,-1) +
        -Ty*2*py*Pow(ry**2,-1) +
        -Tz*2*pz*Pow(rz**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        T1*pz**2*Pow(rz**2,-1) +
        -T1
    )
```

An ellipsoid, or any other *quadric surface* entity, that is to be boosted into a spacetime velocity should initially be at the origin position $(p_x, p_y, p_z) = (0, 0, 0)$ before the first boost operation on the entity. After the boost operation(s) on a quadric surface entity, the boosted entity can be evaluated at any time t, where the entity has a moving position and displays a length contraction effect.

6.1.18 DCSTA GIPNS 2-vector elliptic cylinder, x-axis aligned

The DCSTA GIPNS 2-vector *x-axis aligned elliptic cylinder* is coded as:

```
def GIPNS_ECylinderX(px,py,pz,rx,ry,rz):
    # x-axis aligned elliptic cylinder.
    return (
        T1*(py**2*Pow(ry**2,-1)+pz**2*Pow(rz**2,-1)-1) +
        Tyy*Pow(ry**2,-1) +
        Tzz*Pow(rz**2,-1) +
        -2*py*Ty*Pow(ry**2,-1) +
        -2*pz*Tz*Pow(rz**2,-1)
    )
```

6.1.19 DCSTA GIPNS 2-vector elliptic cylinder, y-axis aligned

The DCSTA GIPNS 2-vector *y-axis aligned elliptic cylinder* is coded as:

```
def GIPNS_ECylinderY(px,py,pz,rx,ry,rz):
    # y-axis aligned elliptic cylinder.
    return (
        T1*(px**2*Pow(rx**2,-1)+pz**2*Pow(rz**2,-1)-1) +
        Txx*Pow(rx**2,-1) +
        Tzz*Pow(rz**2,-1) +
        -2*px*Tx*Pow(rx**2,-1) +
        -2*pz*Tz*Pow(rz**2,-1)
    )
```

6.1.20 DCSTA GIPNS 2-vector elliptic cylinder, z-axis aligned

The DCSTA GIPNS 2-vector *z-axis aligned elliptic cylinder* is coded as:

```
def GIPNS_ECylinderZ(px,py,pz,rx,ry,rz):
    # z-axis aligned elliptic cylinder.
    return (
        T1*(px**2*Pow(rx**2,-1)+py**2*Pow(ry**2,-1)-1) +
        Txx*Pow(rx**2,-1) +
        Tyy*Pow(ry**2,-1) +
        -2*px*Tx*Pow(rx**2,-1) +
        -2*py*Ty*Pow(ry**2,-1)
    )
```

6.1.21 DCSTA GIPNS 2-vector elliptic cone, x-axis aligned

The DCSTA GIPNS 2-vector *x-axis aligned elliptic cone* is coded as:

```
def GIPNS_ConeX(px,py,pz,rx,ry,rz):
    # x-axis aligned elliptic cone.
    return (
        -T1*px**2*Pow(rx**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        T1*pz**2*Pow(rz**2,-1) +
        -Txx*Pow(rx**2,-1) +
        Tyy*Pow(ry**2,-1) +
        Tzz*Pow(rz**2,-1) +
        2*px*Tx*Pow(rx**2,-1) +
        -2*py*Ty*Pow(ry**2,-1) +
        -2*pz*Tz*Pow(rz**2,-1)
    )
```

6.1.22 DCSTA GIPNS 2-vector elliptic cone, y-axis aligned

The DCSTA GIPNS 2-vector *y-axis aligned elliptic cone* is coded as:

```
def GIPNS_ConeY(px,py,pz,rx,ry,rz):
    # y-axis aligned elliptic cone.
    return (
        T1*px**2*Pow(rx**2,-1) +
        -T1*py**2*Pow(ry**2,-1) +
        T1*pz**2*Pow(rz**2,-1) +
        Txx*Pow(rx**2,-1) +
        -Tyy*Pow(ry**2,-1) +
        Tzz*Pow(rz**2,-1) +
        -2*px*Tx*Pow(rx**2,-1) +
        2*py*Ty*Pow(ry**2,-1) +
        -2*pz*Tz*Pow(rz**2,-1)
    )
```

6.1.23 DCSTA GIPNS 2-vector elliptic cone, z-axis aligned

The DCSTA GIPNS 2-vector *z-axis aligned elliptic cone* is coded as:

```
def GIPNS_ConeZ(px,py,pz,rx,ry,rz):
    # z-axis aligned elliptic cone.
    return (
        T1*px**2*Pow(rx**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        -T1*pz**2*Pow(rz**2,-1) +
        Txx*Pow(rx**2,-1) +
        Tyy*Pow(ry**2,-1) +
        -Tzz*Pow(rz**2,-1) +
        -2*px*Tx*Pow(rx**2,-1) +
        -2*py*Ty*Pow(ry**2,-1) +
        2*pz*Tz*Pow(rz**2,-1)
    )
```

6.1.24 DCSTA GIPNS 2-vector elliptic paraboloid, x-axis aligned

The DCSTA GIPNS 2-vector *x-axis aligned elliptic paraboloid* is coded as:

```
def GIPNS_ParaboloidX(px,py,pz,rx,ry,rz):
    # x-axis aligned elliptic paraboloid.
    return (
        -2*pz*Tz*Pow(rz**2,-1) +
        -2*py*Ty*Pow(ry**2,-1) +
        -Tx*Pow(rx,-1) +
        Tzz*Pow(rz**2,-1) +
        Tyy*Pow(ry**2,-1) +
        T1*pz**2*Pow(rz**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        T1*px*Pow(rx,-1)
    )
```

6.1.25 DCSTA GIPNS 2-vector elliptic paraboloid, y-axis aligned

The DCSTA GIPNS 2-vector *y-axis aligned elliptic paraboloid* is coded as:

```
def GIPNS_ParaboloidY(px,py,pz,rx,ry,rz):
    # y-axis aligned elliptic paraboloid.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        -2*pz*Tz*Pow(rz**2,-1) +
        -Ty*Pow(ry,-1) +
        Txx*Pow(rx**2,-1) +
        Tzz*Pow(rz**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        T1*pz**2*Pow(rz**2,-1) +
        T1*py*Pow(ry,-1)
    )
```

6.1.26 DCSTA GIPNS 2-vector elliptic paraboloid, z-axis aligned

The DCSTA GIPNS 2-vector *z-axis aligned elliptic paraboloid* is coded as:

```
def GIPNS_ParaboloidZ(px,py,pz,rx,ry,rz):
    # z-axis aligned elliptic paraboloid.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        -2*py*Ty*Pow(ry**2,-1) +
        -Tz*Pow(rz,-1) +
        Txx*Pow(rx**2,-1) +
        Tyy*Pow(ry**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        T1*pz*Pow(rz,-1)
    )
```

6.1.27 DCSTA GIPNS 2-vector hyperbolic paraboloid

The DCSTA GIPNS 2-vector *z-axis aligned hyperbolic paraboloid* is coded as:

```
def GIPNS_HParaboloidZ(px,py,pz,rx,ry,rz):
    # z-axis aligned hyperbolic paraboloid.
    # A saddle-like shape
    #  that "saddles" x-axis
    #  and "straddles" y-axis
    #  with "up" direction as z-axis.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        2*py*Ty*Pow(ry**2,-1) +
        -Tz*Pow(rz,-1) +
        Txx*Pow(rx**2,-1) +
        -Tyy*Pow(ry**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        -T1*py**2*Pow(ry**2,-1) +
        T1*pz*Pow(rz,-1)
    )
```

6.1.28 DCSTA GIPNS 2-vector hyperboloid of one sheet

The DCSTA GIPNS 2-vector *hyperboloid of one sheet* is coded as:

```
def GIPNS_Hyperboloid1(px,py,pz,rx,ry,rz):
    # z-axis aligned hyperboloid of one sheet.
    # An hourglass-like shape that
    #  is elliptic in the xy-plane
    #  and hyperbolic in xz and yz planes.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        -2*py*Ty*Pow(ry**2,-1) +
        2*pz*Tz*Pow(rz**2,-1) +
        Txx*Pow(rx**2,-1) +
        Tyy*Pow(ry**2,-1) +
        -Tzz*Pow(rz**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        -T1*pz**2*Pow(rz**2,-1) +
        -T1
    )
```

6.1.29 DCSTA GIPNS 2-vector hyperboloid of two sheets

The DCSTA GIPNS 2-vector *hyperboloid of two sheets* is coded as:

```
def GIPNS_Hyperboloid2(px,py,pz,rx,ry,rz):
    # z-axis aligned hyperboloid of two sheets.
    # A shape like two dishes that
    #  are elliptic in the xy-plane
    #  and hyperbolic in xz and yz planes.
    return (
        Tx*2*px*Pow(rx**2,-1) +
        Ty*2*py*Pow(ry**2,-1) +
        -Tz*2*pz*Pow(rz**2,-1) +
        -Txx*Pow(rx**2,-1) +
        -Tyy*Pow(ry**2,-1) +
        Tzz*Pow(rz**2,-1) +
        -T1*px**2*Pow(rx**2,-1) +
        -T1*py**2*Pow(ry**2,-1) +
        T1*pz**2*Pow(rz**2,-1) +
        -T1
    )
```

6.1.30 DCSTA GIPNS 2-vector parabolic cylinder, x-axis aligned

The DCSTA GIPNS 2-vector *x-axis aligned parabolic cylinder* is coded as:

```
def GIPNS_PCylinderX(px,py,pz,rx,ry,rz):
    # Cylinder along x-axis with
    # constant parabola cross-section in yz-plane.
    return (
        -2*py*Ty*Pow(ry**2,-1) +
        -Tz*Pow(rz,-1) +
        Tyy*Pow(ry**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        T1*pz*Pow(rz,-1)
    )
```

6.1.31 DCSTA GIPNS 2-vector parabolic cylinder, y-axis aligned

The DCSTA GIPNS 2-vector *y-axis aligned parabolic cylinder* is coded as:

```
def GIPNS_PCylinderY(px,py,pz,rx,ry,rz):
    # Cylinder along y-axis with
    # constant parabola cross-section in xz-plane.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        -Tz*Pow(rz,-1) +
        Txx*Pow(rx**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        T1*pz*Pow(rz,-1)
    )
```

6.1.32 DCSTA GIPNS 2-vector parabolic cylinder, z-axis aligned

The DCSTA GIPNS 2-vector *z-axis aligned parabolic cylinder* is coded as:

```
def GIPNS_PCylinderZ(px,py,pz,rx,ry,rz):
    # Cylinder along z-axis with
    # constant parabola cross-section in xy-plane.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        -Ty*Pow(ry,-1) +
        Txx*Pow(rx**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        T1*py*Pow(ry,-1)
    )
```

6.1.33 DCSTA GIPNS 2-vector hyperbolic cylinder, x-axis aligned

The DCSTA GIPNS 2-vector *x-axis aligned hyperbolic cylinder* is coded as:

```
def GIPNS_HCylinderX(px,py,pz,rx,ry,rz):
    # Cylinder along x-axis with
    # constant hyperbola cross-section in yz-plane
    # opening up and down the y-axis.
    return (
        -Ty*2*py*Pow(ry**2,-1) +
         Tz*2*pz*Pow(rz**2,-1) +
         Tyy*Pow(ry**2,-1) +
        -Tzz*Pow(rz**2,-1) +
         T1*py**2*Pow(ry**2,-1) +
        -T1*pz**2*Pow(rz**2,-1) +
        -T1
    )
```

6.1.34 DCSTA GIPNS 2-vector hyperbolic cylinder, y-axis aligned

The DCSTA GIPNS 2-vector *y-axis aligned hyperbolic cylinder* is coded as:

```
def GIPNS_HCylinderY(px,py,pz,rx,ry,rz):
    # Cylinder along y-axis with
    # constant hyperbola cross-section in xz-plane
    # opening up and down the z-axis.
    return (
        -Tz*2*pz*Pow(rz**2,-1) +
         Tx*2*px*Pow(rx**2,-1) +
         Tzz*Pow(rz**2,-1) +
        -Txx*Pow(rx**2,-1) +
         T1*pz**2*Pow(rz**2,-1) +
        -T1*px**2*Pow(rx**2,-1) +
        -T1
    )
```

6.1.35 DCSTA GIPNS 2-vector hyperbolic cylinder, z-axis aligned

The DCSTA GIPNS 2-vector *z-axis aligned hyperbolic cylinder* is coded as:

```
def GIPNS_HCylinderZ(px,py,pz,rx,ry,rz):
    # Cylinder along z-axis with
    # constant hyperbola cross-section in xy-plane
    # opening up and down the x-axis.
    return (
        -Tx*2*px*Pow(rx**2,-1) +
         Ty*2*py*Pow(ry**2,-1) +
         Txx*Pow(rx**2,-1) +
        -Tyy*Pow(ry**2,-1) +
         T1*px**2*Pow(rx**2,-1) +
        -T1*py**2*Pow(ry**2,-1) +
        -T1
    )
```

6.1.36 DCSTA GIPNS 2-vector parallel planes pair, perpendicular to x-axis

The DCSTA GIPNS 2-vector *parallel planes pair* $\perp x$-axis is coded as:

```
def GIPNS_PPlanesPairX(px1,px2):
    # Parallel planes pair, x=px1 and x=px2.
    return ( Txx - (px1+px2)*Tx + px1*px2*T1 )
```

6.1.37 DCSTA GIPNS 2-vector parallel planes pair, perpendicular to y-axis

The DCSTA GIPNS 2-vector *parallel planes pair* $\perp y$-axis is coded as:

```
def GIPNS_PPlanesPairY(py1,py2):
    # Parallel planes pair, y=py1 and y=py2.
    return ( Tyy - (py1+py2)*Ty + py1*py2*T1 )
```

6.1.38 DCSTA GIPNS 2-vector parallel planes pair, perpendicular to z-axis

The DCSTA GIPNS 2-vector *parallel planes pair* $\perp z$-axis is coded as:

```
def GIPNS_PPlanesPairZ(pz1,pz2):
    # Parallel planes pair, z=pz1 and z=pz2.
    return ( Tzz - (pz1+pz2)*Tz + pz1*pz2*T1 )
```

6.1.39 DCSTA GIPNS 2-vector non-parallel planes pair, x-axis aligned

The DCSTA GIPNS 2-vector *x-axis aligned non-parallel planes pair* is coded as:

```
def GIPNS_XPlanesPairX(py,pz,ry,rz):
    # The non-parallel planes pair aligned with x-axis
    # is a type of cylinder with constant cross section
    # that is a pair of lines in the yz-plane. The lines
    # intersect at (py,pz), and the slopes of the two
    # lines are +rz/ry and -rz/ry.
    return (
        -2*py*Ty*Pow(ry**2,-1) +
        2*pz*Tz*Pow(rz**2,-1) +
        Tyy*Pow(ry**2,-1) +
        -Tzz*Pow(rz**2,-1) +
        T1*py**2*Pow(ry**2,-1) +
        -T1*pz**2*Pow(rz**2,-1)
    )
```

6.1.40 DCSTA GIPNS 2-vector non-parallel planes pair, y-axis aligned

The DCSTA GIPNS 2-vector *y-axis aligned non-parallel planes pair* is coded as:

```
def GIPNS_XPlanesPairY(px,pz,rx,rz):
    # The non-parallel planes pair aligned with y-axis
    # is a type of cylinder with constant cross section
    # that is a pair of lines in the xz-plane. The lines
    # intersect at (px,pz), and the slopes of the two
    # lines are +rz/rx and -rz/rx.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        2*pz*Tz*Pow(rz**2,-1) +
        Txx*Pow(rx**2,-1) +
        -Tzz*Pow(rz**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        -T1*pz**2*Pow(rz**2,-1)
    )
```

6.1.41 DCSTA GIPNS 2-vector non-parallel planes pair, z-axis aligned

The DCSTA GIPNS 2-vector *z-axis aligned non-parallel planes pair* is coded as:

```
def GIPNS_XPlanesPairZ(px,py,rx,ry):
    # The non-parallel planes pair aligned with z-axis
    # is a type of cylinder with constant cross section
    # that is a pair of lines in the xy-plane. The lines
    # intersect at (px,py), and the slopes of the two
    # lines are +ry/rx and -ry/rx.
    return (
        -2*px*Tx*Pow(rx**2,-1) +
        2*py*Ty*Pow(ry**2,-1) +
        Txx*Pow(rx**2,-1) +
        -Tyy*Pow(ry**2,-1) +
        T1*px**2*Pow(rx**2,-1) +
        -T1*py**2*Pow(ry**2,-1)
    )
```

6.1.42 CSTA1 GIPNS 1-vector hyperplane

The CSTA1 GIPNS 1-vector *hyperplane* is coded as:

```
def GIPNS_HPlane1(p,n):
    # CSTA1 1-vector hyperplane
    # p is any point on the hyperplane
    # n is the normal vector
    # the magnitude of n is not significant
    return ( n + (p|n)*ei1 )
```

6.1.43 CSTA2 GIPNS 1-vector hyperplane

The CSTA2 GIPNS 1-vector *hyperplane* is coded as:

```
def GIPNS_HPlane2(p,n):
    # CSTA2 1-vector hyperplane
    # p is any point on the hyperplane
    # n is the normal vector
    # the magnitude of n is not significant
    p2 = (p1|e1)*e7 + (-p1|e2)*e8 + (-p1|e3)*e9 + (-p1|e4)*e10
    n2 = (n1|e1)*e7 + (-n1|e2)*e8 + (-n1|e3)*e9 + (-n1|e4)*e10
    return ( n2 + (p2|n2)*ei2 )
```

6.1.44 DCSTA GIPNS 2-vector hyperplane

The DCSTA GIPNS 2-vector *hyperplane* is coded as:

```
def GIPNS_HPlane(p,n):
    # DCSTA 2-vector hyperplane
    # p is any point on the hyperplane
    # n is the normal vector
    # the magnitude of n is not significant
    return ( GIPNS_HPlane1(p,n)^GIPNS_HPlane2(p,n) )
```

6.1.45 CSTA1 GIPNS 1-vector hyperhyperboloid of one sheet

The CSTA1 GIPNS 1-vector *hyperhyperboloid of one sheet (hyperpseudosphere)* is coded as:

```
def GIPNS_HPSphere1(p,r):
    # CSTA1 1-vector hyperpseudosphere (hyperhyperboloid)
    # p is the STA center point
    # r is the radius
    # negative radius makes imaginary hyperpseudosphere
    f = Pow(abs(r),-1)*r
    return ( EV1(p) + f*Pow(2,-1)*r**2*ei1 )
```

6.1.46 CSTA2 GIPNS 1-vector hyperhyperboloid of one sheet

The CSTA2 GIPNS 1-vector *hyperhyperboloid of one sheet (hyperpseudosphere)* is coded as:

```
def GIPNS_HPSphere2(p,r):
    # CSTA1 1-vector hyperpseudosphere (hyperhyperboloid)
    # p is the STA center point
    # r is the radius
    # negative radius makes imaginary hyperpseudosphere
    f = Pow(abs(r),-1)*r
    return ( EV2(p) + f*Pow(2,-1)*r**2*ei2 )
```

6.1.47 DCSTA GIPNS 2-vector hyperhyperboloid of one sheet

The DCSTA GIPNS 2-vector *hyperhyperboloid of one sheet (hyperpseudosphere)* is coded as:

```
def GIPNS_HPSphere(p,r):
    # DCSTA 2-vector hyperpseudosphere (hyperhyperboloid)
    # p is the STA center point
    # r is the radius
    # negative radius makes imaginary hyperpseudosphere
    return ( GIPNS_HPSphere1(p,r)^GIPNS_HPSphere2(p,r) )
```

6.1.48 CSTA1 GIPNS 2-vector plane

The CSTA1 GIPNS 2-vector *plane* is coded as:

```
def GIPNS_Plane1(p,da,db):
    # p is any STA1 point on the plane
    # da is STA1 direction one of plane
    # db is STA1 direction two of plane
    # The STA1 plane bivector B is da^db
    p1 = p
    B1 = da^db
    N1 = Pow(sqrt(scalar(B1|(e1*B1.rev()*e1))),-1)*B1
    D1 = ((1*N1*1)|I41.inv())
    return ( D1 - ((p1|D1)^ei1) )
```

6.1.49 CSTA2 GIPNS 2-vector plane

The CSTA2 GIPNS 2-vector *plane* is coded as:

```
def GIPNS_Plane2(p,da,db):
    # p is any STA1 point on the plane
    # da is STA1 direction one of plane
    # db is STA1 direction two of plane
    p2 = (p|e1)*e7+(-p|e2)*e8+(-p|e3)*e9+(-p|e4)*e10
    da2 = (da|e1)*e7+(-da|e2)*e8+(-da|e3)*e9+(-da|e4)*e10
    db2 = (db|e1)*e7+(-db|e2)*e8+(-db|e3)*e9+(-db|e4)*e10
    B2 = da2^db2
    N2 = Pow(sqrt(scalar(B2|(e7*B2.rev()*e7))),-1)*B2
    D2 = (N2|I42.inv())
    return ( D2 - ((p2|D2)^ei2) )
```

6.1.50 DCSTA GIPNS 4-vector standard plane

The DCSTA GIPNS 4-vector *standard plane* is coded as:

```
def GIPNS_Plane(p,da,db):
    # p is any STA1 point on the plane
    # da is STA1 direction one of plane
    # db is STA1 direction two of plane
    return ( GIPNS_Plane1(p,da,db)^GIPNS_Plane2(p,da,db) )
```

The standard plane can be intersected with all other DCSTA GIPNS surface entities.

6.1.51 CSTA1 GIPNS 3-vector line

The CSTA1 GIPNS 3-vector *line* is coded as:

```
def GIPNS_Line1(p,d):
    # p is any STA1 point on the line
    # d is the STA1 direction of line
    dc = e1*d*e1
    d1 = Pow(sqrt(scalar(d|dc)),-1)*d
    D1 = d1|(-I41)
    return ( D1 + ((p|D1)^ei1) )
```

6.1.52 CSTA2 GIPNS 3-vector line

The CSTA2 GIPNS 3-vector *line* is coded as:

```
def GIPNS_Line2(p,d):
    # p is any STA1 point on the line
    # d is the STA1 direction of line
    p2 = (p|e1)*e7+(-p|e2)*e8+(-p|e3)*e9+(-p|e4)*e10
    dc = e1*d*e1
    d1 = Pow(sqrt(scalar(d|dc)),-1)*d
    d2 = (d1|e1)*e7+(-d1|e2)*e8+(-d1|e3)*e9+(-d1|e4)*e10
    D2 = d2|(-I42)
    return ( D2 + ((p2|D2)^ei2) )
```

6.1.53 DCSTA GIPNS 6-vector standard line

The DCSTA GIPNS 6-vector *standard line* is coded as:

```
def GIPNS_Line(p,d):
    # p is any STA1 point on the line
    # d is the STA1 direction of line
    return ( GIPNS_Line1(p,d)^GIPNS_Line2(p,d) )
```

The standard line can be intersected with all other DCSTA GIPNS surface entities.

6.1.54 CSTA1 plane-line intersection

The CSTA1 plane-line intersection is coded as:

```
def GIPNS_PlaneLineIntersection1(p,l):
    # Intersect GIPNS_Plane1 p and GIPNS_Line1 l
    plwedge = (p^l)
    if plwedge != 0: return ei1
    plmeet = (((p|I41.inv())^(l|I41.inv()))|I41)
    if plmeet == 0: return l
    return ((e1*plmeet*e1)|p)^l
```

6.1.55 CSTA2 plane-line intersection

The CSTA2 plane-line intersection is coded as:

```
def GIPNS_PlaneLineIntersection2(p,l):
    # Intersect GIPNS_Plane2 p and GIPNS_Line2 l
    plwedge = (p^l)
    if plwedge != 0: return ei2
    plmeet = (((p|I42.inv())^(l|I42.inv()))|I42)
    if plmeet == 0: return l
    return ((e7*plmeet*e7)|p)^l
```

6.1.56 CSTA1 GOPNS 2-vector point pair decomposition

The decomposition of a CSTA1 GOPNS 2-vector *point pair* is coded as:

```
def GOPNS_PointPairDecomp1(pp,pm):
    # pp is a CSTA1 GOPNS 2-vector point pair
    # pm is -1 or 1 to select a point of the pair
    # returns a CSTA1 null 1-vector point entity
    return ( (pp + pm*sqrt(scalar(pp|pp)))*(-ei1|pp).inv() )
```

6.1.57 CSTA2 GOPNS 2-vector point pair decomposition

The decomposition of a CSTA2 GOPNS 2-vector *point pair* is coded as:

```
def GOPNS_PointPairDecomp2(pp,pm):
    # pp is a CSTA2 GOPNS 2-vector point pair
    # pm is -1 or 1 to select a point of the pair
    # returns a CSTA2 null 1-vector point entity
    return ( (pp + pm*sqrt(scalar(pp|pp)))*(-ei2|pp).inv() )
```

6.1.58 CSTA1 GOPNS 2-vector flat point projection

The projection of the point of a CSTA1 GOPNS 2-vector *flat point* is coded as:

```
def GOPNS_FlatPointProj1(fp):
    # fp is a CSTA1 GOPNS 2-vector flat point
    # returns the STA1 vector projection of the point
    E = eo1^ei1
    return ( -(fp|eo1)*Pow(scalar(E|fp),-1) - eo1 )
```

6.1.59 CSTA2 GOPNS 2-vector flat point projection

The projection of the point of a CSTA2 GOPNS 2-vector *flat point* is coded as:

```
def GOPNS_FlatPointProj2(fp):
    # fp is a CSTA2 GOPNS 2-vector flat point
    # returns the STA2 vector projection of the point
    E = eo2^ei2
    return ( -(fp|eo2)*Pow(scalar(E|fp),-1) - eo2 )
```

6.1.60 SA1, STA1, and CSTA1 2-versor rotor

The CSTA1 2-versor spatial *rotor* is coded as:

```
def Rotor1(axis,angle):
    # Spatial rotor in SA1, STA1, and CSTA1, where
    #  axis is SA1 vector axis of rotation and
    #  angle is scalar angle of rotation in degrees.
    ax1 = Pow(norm(axis),-1)*axis
    ang = pi*Pow(180,-1)*angle
    return (
        cos(ang*Pow(2,-1)) +
        sin(ang*Pow(2,-1))*(ax1|(-I31))
    )
```

6.1.61 SA2, STA2, and CSTA2 2-versor rotor

The CSTA2 2-versor spatial *rotor* is coded as:

```
def Rotor2(axis,angle):
    # Spatial rotor in SA2, STA2, and CSTA2, where
    #  axis is SA1 vector axis of rotation and
    #  angle is scalar angle of rotation in degrees.
    ax1 = Pow(norm(axis),-1)*axis
    ax2 = (-ax1|e2)*e8 + (-ax1|e3)*e9 + (-ax1|e4)*e10
    ang = pi*Pow(180,-1)*angle
    return (
        cos(ang*Pow(2,-1)) +
        sin(ang*Pow(2,-1))*(ax2|(-I32))
    )
```

6.1.62 DCSTA 4-versor rotor

The DCSTA 4-versor spatial *rotor* is coded as:

```
def Rotor(axis,angle):
    # Spatial rotor in DCSTA, where
    #  axis is SA1 vector axis of rotation and
    #  angle is scalar angle of rotation in degrees
    return ( Rotor1(axis,angle)^Rotor2(axis,angle) )
```

6.1.63 CSTA1 2-versor line rotor

The CSTA1 2-versor *line rotor* for the rotation around a line is coded as:

```
def LRotor1(p,d,a):
    # Rotor around a line l by angle a in degrees
    # line l is given by STA1 point p and direction d
    l = GIPNS_Line1(p,d)
    t = Rational(1,2)*pi*Pow(180,-1)*a
    return ( cos(t) + sin(t)*(-e1|l) )
```

6.1.64 CSTA2 2-versor line rotor

The CSTA2 2-versor *line rotor* for the rotation around a line is coded as:

```
def LRotor2(p,d,a):
    l = GIPNS_Line2(p,d)
    t = Rational(1,2)*pi*Pow(180,-1)*a
    return ( cos(t) + sin(t)*(-e7|l) )
```

6.1.65 DCSTA 4-versor line rotor

The DCSTA 4-versor *line rotor* for the rotation around a line is coded as:

```
def LRotor(p,d,a):
    return LRotor1(p,d,a)^LRotor2(p,d,a)
```

6.1.66 STA1 and CSTA1 2-versor hyperbolic rotor (boost operator)

The CSTA1 2-versor spacetime *hyperbolic rotor (boost operator)* is coded as:

```
def HRotor1(b,d):
    # STA1 and CSTA1 boost operator, where
    #  0<=b<=1 is scalar natural speed of boost and
    #  d is SA1 direction vector of boost velocity
    v1 = Pow(-scalar(d|d),-1)*d
    r = atanh(b)
    return ( cosh(Pow(2,-1)*r) + sinh(Pow(2,-1)*r)*(v1^e1) )
```

The `b` is the natural speed $\beta_{\mathbf{v}}$. The `d` is the SA1 spatial velocity direction $\hat{\mathbf{v}}$ that is normalized as `v1`. The spatial velocity of the boost is $\mathbf{v} = \beta_{\mathbf{v}} c\hat{\mathbf{v}} = \|\mathbf{v}\|\hat{\mathbf{v}}$ relative to an observer $\boldsymbol{o} = ct\mathbf{e}_1$. The `r` is the rapidity $\varphi_{\mathbf{v}} = \text{atanh}(\beta_{\mathbf{v}})$.

6.1.67 STA2 and CSTA2 2-versor hyperbolic rotor (boost operator)

The CSTA2 2-versor spacetime *hyperbolic rotor (boost operator)* is coded as:

```
def HRotor2(b,d):
    # STA2 and CSTA2 boost operator, where
    #  0<=b<=1 is scalar natural speed of boost and
    #  d is SA1 direction vector of boost velocity
    v1 = Pow(-scalar(d|d),-1)*d
    v2 = (-v1|e2)*e8 + (-v1|e3)*e9 + (-v1|e4)*e10
    r = atanh(b)
    return ( cosh(Pow(2,-1)*r) + sinh(Pow(2,-1)*r)*(v2^e7) )
```

6.1.68 DCSTA 4-versor hyperbolic rotor (boost operator)

The DCSTA 4-versor spacetime *hyperbolic rotor (boost operator)* is coded as:

```
def HRotor(b,d):
    # DCSTA boost operator, where
    #  0<=b<=1 is scalar natural speed of boost and
    #  d is SA1 direction vector of boost velocity
    return ( HRotor1(b,d)^HRotor2(b,d) )
```

For an *anisotropic dilation* of a *quadric surface* `Q` by factor d in direction `d`, then speed `b` should be set to $\beta_{\mathbf{v}} = \sqrt{1-d^2}$, which may be an imaginary number.

The anisotropic dilation of `Q` by a dilation factor `d` in an SA1 direction `v=vx*e2+vy*e3+vz*e4` is coded as:

```
((
    HRotor(sqrt(1-d**2),v)*
    Q*
    HRotor(sqrt(1-d**2),v).rev()
)|IDS)*IDS.inv()
```

The projection using `IDS` is the space projection into the $\mathcal{G}_{2,8}$ anti-DCGA, which discards imaginary components that are by-products of the directed scaling operation. A good example to try can use `Q=Ellipsoid(px,py,pz,rx,ry,rz)`.

6.1.69 CSTA1 2-versor translator

The CSTA1 2-versor spacetime *translator* is coded as:

```
def Translator1(d):
    # CSTA1 spacetime translator, where
    #  d is an STA1 spacetime displacement vector.
    d1 = d
    return ( 1 - Pow(2,-1)*(d1^ei1) )
```

6.1.70 CSTA2 2-versor translator

The CSTA2 2-versor spacetime *translator* is coded as:

```
def Translator2(d):
    # CSTA2 spacetime translator, where
    #  d is an STA1 spacetime displacement vector.
    d2 = (d|e1)*e7 + (-d|e2)*e8 + (-d|e3)*e9 + (-d|e4)*e10
    return ( 1 - Pow(2,-1)*(d2^ei2) )
```

6.1.71 DCSTA 4-versor translator

The DCSTA 4-versor spacetime *translator* is coded as:

```
def Translator(d):
    # DCSTA spacetime translator, where
    #  d is an STA1 spacetime displacement vector.
    return ( Translator1(d)^Translator2(d) )
```

6.1.72 CSTA1 2-versor isotropic dilator

The CSTA1 2-versor spacetime *isotropic dilator* is coded as:

```
def Dilator1(d):
    # CSTA1 isotropic dilator, where
    #  d is the scalar dilation factor.
    # Note: dilation factor d=0 is not generally valid.
    return ( Pow(2,-1)*(1+d) + Pow(2,-1)*(1-d)*(ei1^eo1) )
```

6.1.73 CSTA2 2-versor isotropic dilator

The CSTA2 2-versor spacetime *isotropic dilator* is coded as:

```
def Dilator2(d):
    # CSTA2 isotropic dilator, where
    #  d is the scalar dilation factor.
    # Note: dilation factor d=0 is not generally valid.
    return ( Pow(2,-1)*(1+d) + Pow(2,-1)*(1-d)*(ei2^eo2) )
```

6.1.74 DCSTA 4-versor isotropic dilator

The DCSTA 4-versor spacetime *isotropic dilator* is coded as:

```
def Dilator(d):
    # DCSTA isotropic dilator, where
    #  d is the scalar dilation factor.
    # Note: dilation factor d=0 is not generally valid.
    return ( Dilator1(d)^Dilator2(d) )
```

The *anisotropic dilator* on quadric surface entities is implemented using the *hyperbolic rotor* (§6.1.68).

6.2 Example computations

6.2.1 Reframe to new observer in STA

The observer position is $\boldsymbol{o}t = ct\mathbf{e}_1$, and a particle $\boldsymbol{v}t = (\boldsymbol{o} + \mathbf{v})t$ moves relative to $\boldsymbol{o}$, where

$$\mathbf{v} \;=\; \beta_{\mathbf{v}} c\mathbf{e}_2 = \frac{1}{2} c\mathbf{e}_2.$$

We want to change observer as $\boldsymbol{v} \to \boldsymbol{o}$ to make $\boldsymbol{v}$ the new observer, and to compute the current observer as $\boldsymbol{o} \to \boldsymbol{o}'$ relative to the new observer $\boldsymbol{v} \to \boldsymbol{o}$. Solution: use the reframe operation on $\boldsymbol{o}$, followed by a position renormalization.

```
o_rel_v = (
    HRotor1( Rational(1,2), e2 ).rev()*
    ( c*t*e1 )*
    HRotor1( Rational(1,2), e2 )
)
normalized = c*t*Pow( scalar(o_rel_v|e1), -1 )*o_rel_v
normalized
```

$$\text{normalized} \;=\; ct\mathbf{e}_1 - \frac{ct}{2}\mathbf{e}_2.$$

The old observer is now seen as a particle moving with velocity $\mathbf{v} = -\frac{1}{2}c\mathbf{e}_2$ relative to the new observer $\boldsymbol{o} = ct\mathbf{e}_1$. The time t is the proper time of the new observer $\boldsymbol{o}$, which was $\boldsymbol{v}$.

6.2.2 Collinear velocity addition in STA

A particle moves with velocity $\mathbf{u} = \frac{3}{4}c\mathbf{e}_2$ relative to another particle moving with velocity $\mathbf{v} = \frac{1}{2}c\mathbf{e}_2$ relative to an observer $\boldsymbol{o} = c\mathbf{e}_1$. The two velocities $\mathbf{u}$ and $\mathbf{v}$ are collinear, and if we simply add the velocities, we may conclude that $\mathbf{u}$ relative to $\boldsymbol{o}$ is a velocity $\mathbf{v} + \mathbf{u} = \frac{5}{4}c\mathbf{e}_2$. However, this speed is greater than light speed c, which according to the physical theory of special relativity is an impossible speed. Velocities cannot be simply added, and we must use a reframe operation to reframe $\mathbf{u}$ relative to $\boldsymbol{o}$. Relative to $\boldsymbol{v} = \boldsymbol{o} + \mathbf{v}$, the particle moving with velocity $\mathbf{u}$ is written $\boldsymbol{u} = \boldsymbol{o} + \mathbf{u} = ct\mathbf{e}_1 + \mathbf{u}$, where *this* $\boldsymbol{o}$ *is* $\boldsymbol{v}$ as the observer and *this time* t is its proper time. We want this $\boldsymbol{u}$ reframed relative to observer $\boldsymbol{o}$ of $\boldsymbol{v} = \boldsymbol{o} + \mathbf{v}$. The solution is to apply to $\boldsymbol{u}$ the operation for the reverse of the reframe relative to $\boldsymbol{v}$ that goes back to relative to its $\boldsymbol{o}$, and this reframe is also seen as *boosting* the particle $\boldsymbol{u} = \boldsymbol{o} + \mathbf{u}$ by the velocity $\mathbf{v}$ relative to $\boldsymbol{o}$.

```
u_rel_o = (
    HRotor1( Rational(1,2), e2 )*
    ( c*t*e1 + Rational(3,4)*c*t*e2 )*
    HRotor1( Rational(1,2), e2 ).rev()
)
normalized = c*t*Pow( scalar(u_rel_o|e1), -1 )*u_rel_o
normalized
```

$$\text{normalized} \;=\; ct\mathbf{e}_1 + \frac{10}{11}ct\mathbf{e}_2.$$

The result is relativistic velocity addition, where the boost of velocities does not exceed the speed of light c.

6.2.3 Velocity addition in STA

The velocities $\mathbf{u}$ and $\mathbf{v}$ need not be collinear, and the same operation of the previous section (§6.2.2) for collinear velocities can be applied to reframe any velocity $\mathbf{u}$ relative to $\boldsymbol{v} = \boldsymbol{o} + \mathbf{v}$ into $\mathbf{u}$ relative to $\boldsymbol{o}$. The result is the so-called *velocity-addition formula*, which could also be called the *velocity boost formula*,

$$\begin{aligned}
\boldsymbol{u}_{\text{rel}\,\boldsymbol{v}} \to \boldsymbol{u}_{\text{rel}\,\boldsymbol{o}} &= \boldsymbol{o} + \mathbf{u}_{\text{rel}\,\boldsymbol{o}} \\
&= \boldsymbol{o} + \mathbf{u} \oplus \mathbf{v} \\
&\quad ct\boldsymbol{\gamma}_0 + \frac{\mathbf{u}^{\|\hat{\mathbf{v}}} + \sqrt{1 - \frac{\|\mathbf{v}\|^2}{c^2}}\,\mathbf{u}^{\perp\hat{\mathbf{v}}} + \mathbf{v}}{1 - \frac{\mathbf{u}\cdot\mathbf{v}}{c^2}}
\end{aligned}$$

where $\boldsymbol{o} = ct\boldsymbol{\gamma}_0$, and the $\mathcal{G}_{0,3}$ SA metric gives

$$\begin{aligned}
\hat{\mathbf{v}} &= \frac{\mathbf{v}}{\|\mathbf{v}\|} = \frac{\mathbf{v}}{\sqrt{-\mathbf{v}^2}} \\
\hat{\mathbf{v}}^2 &= -1 \\
\mathbf{u}\cdot\mathbf{v} &= -\|\mathbf{u}\|\|\mathbf{v}\|\cos(\theta_{\mathbf{uv}}).
\end{aligned}$$

The notation $\mathbf{u} \oplus \mathbf{v}$ can be read "$\mathbf{u}$ boosted by $\mathbf{v}$" since this is the actual operation, but this may be backwards compared to some other literature. In general, $\mathbf{u} \oplus \mathbf{v} \neq \mathbf{v} \oplus \mathbf{u}$. Some other identities are

$$\begin{aligned} \mathbf{u} = \mathbf{u}^{\|\hat{\mathbf{v}}} + \mathbf{u}^{\perp\hat{\mathbf{v}}} &= (\mathbf{u} \cdot \hat{\mathbf{v}} + \mathbf{u} \wedge \hat{\mathbf{v}})\hat{\mathbf{v}}^{-1} = (-\mathbf{u} \cdot \hat{\mathbf{v}})\hat{\mathbf{v}} + (\hat{\mathbf{v}} \wedge \mathbf{u}) \cdot \hat{\mathbf{v}} \\ \beta_{\mathbf{v}} &= \frac{\|\mathbf{v}\|}{c} \\ \gamma_{\mathbf{v}} &= \frac{1}{\sqrt{1-\beta_{\mathbf{v}}^2}}. \end{aligned}$$

When the boost velocity approaches light speed $\|\mathbf{v}\| \to c$, we get

$$\mathbf{u} \oplus \mathbf{v} = \frac{\|\mathbf{u}\|\cos(\theta_{\mathbf{uv}})\hat{\mathbf{v}} + c\hat{\mathbf{v}}}{1 + \frac{\|\mathbf{u}\|\cos(\theta_{\mathbf{uv}})}{c}} = \frac{c(\|\mathbf{u}\|\cos(\theta_{\mathbf{uv}}) + c)\hat{\mathbf{v}}}{c + \|\mathbf{u}\|\cos(\theta_{\mathbf{uv}})} = c\hat{\mathbf{v}} = \mathbf{v}.$$

For collinear $\mathbf{u}$ and $\mathbf{v}$, then

$$\mathbf{u} \oplus \mathbf{v} = \alpha\mathbf{v} \oplus \mathbf{v} = \mathbf{v} \oplus \mathbf{u} = \frac{\mathbf{u} + \mathbf{v}}{1 + \frac{\|\mathbf{u}\|\|\mathbf{v}\|}{c^2}}$$

where as the boost velocity approaches light speed $\|\mathbf{v}\| \to c$,

$$\mathbf{u} \oplus \mathbf{v} \to \frac{c(\|\mathbf{u}\| + c)\hat{\mathbf{u}}}{c + \|\mathbf{u}\|} = c\hat{\mathbf{u}} = \mathbf{v}.$$

For perpendicular $\mathbf{u}$ and $\mathbf{v}$, then

$$\mathbf{u} \oplus \mathbf{v} = \sqrt{1 - \frac{\|\mathbf{v}\|^2}{c^2}}\mathbf{u} + \mathbf{v} = \frac{1}{\gamma_{\mathbf{v}}}\mathbf{u} + \mathbf{v}$$

where as the boost velocity approaches light speed $\|\mathbf{v}\| \to c$, $\mathbf{u}/\gamma_{\mathbf{v}} \to 0$ and $\mathbf{u} \oplus \mathbf{v} \to \mathbf{v}$. The velocity-addition formula is derived and discussed more in [7].

6.2.4 Boost of an ellipsoid entity for contraction effect

Any DCSTA GIPNS 2-vector quadric surface entity can be boosted into a velocity in spacetime. Boosting sets the quadric surface into motion at constant velocity and gives the surface a length contraction effect. As an example of the contraction effect, we can boost an ellipsoid to a natural speed $\beta_{\mathbf{v}} = \sqrt{1-d^2}$ for the dilation factor d. A good example is to choose $d = 1/2$ to squeeze the ellipsoid into one-half its length in the direction of the velocity.

```
moving_ellipsoid = (
    HRotor( sqrt(1-Rational(1,2)**2), e2 )*
    Ellipsoid(0,0,0,10,10,10)*
    HRotor( sqrt(1-Rational(1,2)**2), e2 ).rev()
)
print( N(V|moving_ellipsoid) )
```

The `moving_ellipsoid` is evaluated at a symbolic point `V`. The full symbolic output can be long, therefore numeric output has been generated using `N()`. The result is printed in plain text using `print()`. Output of this form can be graphed using *Mayavi*. For graphing, it works well to use natural units, where $c = 1$, so that the graph can be near the origin. Mayavi seems to work best if graphing can be limited to a small cube around the origin that is about ± 20 units on each axis. If Mayavi is installed and working, a small `mayavi.py` python file can be created to graph this output (copied into `surface`) as:

```
from __future__ import division
from numpy import *
from mayavi import mlab

mlab.figure(bgcolor=(1,1,1))
x, y, z = mgrid[-20:20:100j, -20:20:100j, -20:20:100j]

# axes
cylx = y**2 + z**2 - 1/10
cyly = x**2 + z**2 - 1/10
cylz = y**2 + x**2 - 1/10
mlab.contour3d(x,y,z,cylx,contours=[0],opacity=0.25,color=(1,0,0))
mlab.contour3d(x,y,z,cyly,contours=[0],opacity=0.25,color=(0,1,0))
mlab.contour3d(x,y,z,cylz,contours=[0],opacity=0.25,color=(0,0,1))

# function for rendering a dot somewhere
def dotat(px,py,pz):
    blackdot = (x-px)**2 + (y-py)**2 + (z-pz)**2 - 1/sqrt(5)
    mlab.contour3d(
        x, y, z, blackdot, contours=[0],
        opacity=0.5, color=(0,0,0)
    )
    return

# plot some dots
dotat(5,0,0)
dotat(0,10,0)
dotat(0,0,10)

# Set the light speed (units per second)
#  Use a small unit or else boosted moving surfaces move
#  out of graphing range after only a few time units.
c = 1
# Set the time.
#  Boosted surfaces move natural-speed units per time unit.
#  At t=20, a surface at speed c=1 moves out of graphing range.
t = 0
# The numerical printed output, copied into here:
surface = (
    0.03*c**2*t**2 - 0.0692820323027551*c*t*x +
    0.04*x**2 + 0.01*y**2 + 0.01*z**2 - 0.999999999999999
)
# Mayavi rendering function
mlab.contour3d(
    x, y, z, surface, contours=[0], opacity=0.5,
    color=(0.0, 1.0, 1.0)
)
```

The `mayavi.py` file is saved and then run from a command line as:

```
ipython mayavi.py
```

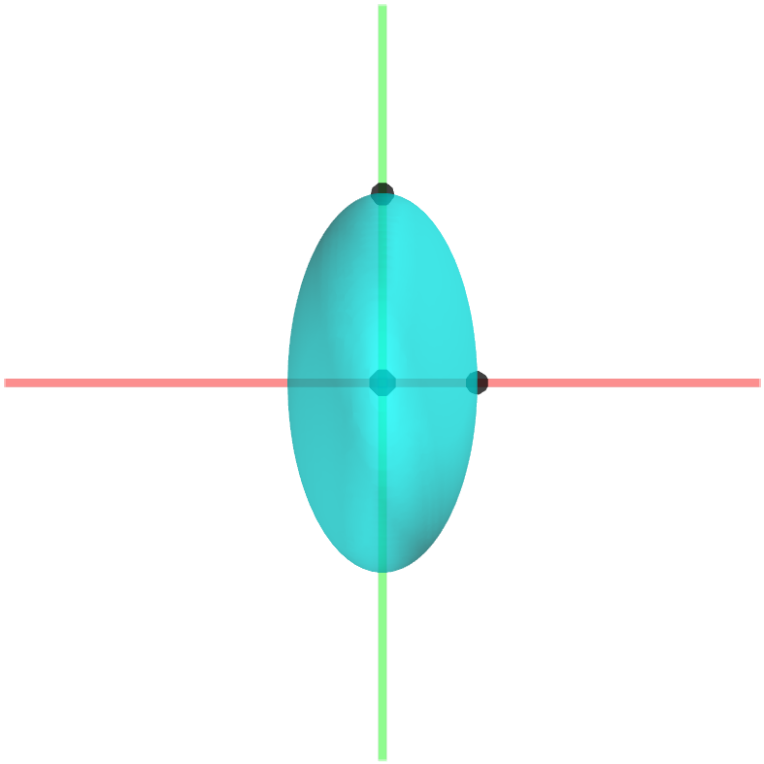

Figure 1. Ellipsoid (sphere $r=10$) boosted to $\beta_{\mathbf{v}}=\sqrt{1-\left(\frac{1}{2}\right)^2}$ in x-direction

Figure 1 shows a boosted ellipsoid at time $t=0$. The ellipsoid was initially at the origin and spherical with radius $r=r_x=r_y=r_z=10$. The spherical ellipsoid was boosted into a natural speed $\beta_{\mathbf{v}}=\sqrt{1-\left(\frac{1}{2}\right)^2}$ for a dilation factor $d=\frac{1}{2}$ in the x-direction $\hat{\mathbf{v}}=\boldsymbol{\gamma}_1=\mathbf{e}_2$. The boosted sphere is squeezed by the boost into an ellipsoid that is length-contracted to half-size in the x-direction with $r_x=5$, while the y and z directions hold their sizes with $r_y=r_z=10$. As the time t is increased, the boosted sphere moves toward the right along the x-axis. For natural units $c=1$, the boosted sphere moves $\beta_{\mathbf{v}}=\sqrt{3}/2\approx 0.866$ units per time unit.

7 Conclusion

The $\mathcal{G}_{4,8}$ Double Conformal Space-Time Algebra (DCSTA) has been presented in this paper as a straightforward extension of the $\mathcal{G}_{8,2}$ Double Conformal / Darboux Cyclide Geometric Algebra (DCGA). DCSTA is a large, complicated algebra and this paper may contain some mistakes and has probably overlooked some things that should have been discussed. Nevertheless, this author feels that this paper substantially conveys the basic ideas and concepts of DCSTA. Certainly, much further research can be done into DCSTA and its applications.

References

[1] Alan Bromborsky. *Geometric Algebra Module for Sympy*. 2015.

[2] Chris Doran and Anthony Lasenby. *Geometric Algebra for Physicists*. Cambridge: Cambridge University Press, Paperback reprint of the 2003 original edition, 2007.

[3] L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science (Revised Edition): An Object-Oriented Approach to Geometry*. The Morgan Kaufmann Series in Computer Graphics. Elsevier Science, 2009.

[4] Robert B. Easter. Conic and Cyclidic Sections in the G8,2 Geometric Algebra, DCGA. viXra.org, 2015.

[5] Robert B. Easter. Differential Operators in the G8,2 Geometric Algebra, DCGA. viXra.org, 2015.

[6] Robert B. Easter. G8,2 Geometric Algebra, DCGA. viXra.org, 2015.

[7] Robert B. Easter. *Quaternions and Clifford Geometric Algebras*. ViXra.org, 2015.

[8] Hamilton, Sir William Rowan. *Lectures on Quaternions: Containing a Systematic Statement of a New Mathematical Method; of which the Principles Were Communicated in 1843 to the Royal Irish Academy; and which Has Since Formed the Subject of Successive Courses of Lectures, Delivered in 1848 and Subsequent Years, in the Halls of Trinity College, Dublin: with Numerous Illustrative Diagrams, and with Some Geometrical and Physical Applications*. Dublin: Hodges and Smith, Grafton-Street, Booksellers to the University. London: Whittaker & Co., Ave-Maria Lane. Cambridge: Macmillan & Co., 1853.

[9] David Hestenes. *Space-Time Algebra*. Springer, Second edition, 2015.

[10] David Hestenes and Garret Sobczyk. *Clifford Algebra to Geometric Calculus, A Unified Language for Mathematics and Physics*, volume 5 of *Fundamental Theories of Physics*. Dordrecht-Boston-Lancaster: D. Reidel Publishing Company, a Member of the Kluwer Academic Publishers Group, 1984.

[11] Christian Perwass. *Geometric Algebra with Applications in Engineering*, volume 4 of *Geometry and Computing*. Springer, 2009.

[12] P. Ramachandran and G. Varoquaux. Mayavi: 3D Visualization of Scientific Data. *Computing in Science & Engineering*, 13(2):40–51, 2011.

[13] SymPy Development Team. *SymPy: Python library for symbolic mathematics*. 2015.

[14] Gerald Sommer, editor. *Geometric Computing with Clifford Algebras, Theoretical Foundations and Applications in Computer Vision and Robotics*. Berlin: Springer, 2001.