G8,2 Geometric Algebra, DCGA

by Robert Benjamin Easter

Abstract

This paper introduces the Double Conformal / Darboux Cyclide Geometric Algebra (DCGA), based in the $\mathcal{G}_{8,2}$ Clifford geometric algebra. DCGA is an extension of CGA and has entities representing points and general Darboux cyclide surfaces in Euclidean 3D space. The general Darboux cyclide is a quartic surface. Darboux cyclides include circular tori and all quadrics, and also all surfaces formed by their inversions in spheres. Dupin cyclide surfaces can be formed as inversions in spheres of circular toroid, cylinder, and cone surfaces. Parabolic cyclides are cubic surfaces formed by inversion spheres centered on other surfaces. All DCGA entities can be conformally transformed in 3D space by rotors, dilators, translators, and motors, which are all types of versors. All entities can be inversed in general spheres and reflected in general planes. Entities representing the intersections of surfaces can be created by wedge products. All entities can be intersected with spheres, planes, lines, and circles. DCGA provides a higher-level algebra for working with 3D geometry in an object/entity-oriented system of mathematics above the level of the underlying implicit surface equations of algebraic geometry. DCGA could be used in the study of geometry in 3D, and also for some applications.

Keywords: conformal geometric algebra, Darboux Dupin cyclide, quadric surface

A.M.S. subject classification: 15A66, 53A30, 14J26, 53A05, 51N20, 51K05

1 Introduction

This paper[1] introduces[2] an application of the $\mathcal{G}_{8,2}$ geometric algebra [7][8], tentatively named in this paper the *Double Conformal / Darboux Cyclide Geometric Algebra* (DCGA). The organization of this paper is not entirely front to back and is organized to be useful as a quick reference text. The reader should skip around as needed.

The $\mathcal{G}_{8,2}$ geometric algebra contains two subspaces of the $\mathcal{G}_{4,1}$ *Conformal Geometric Algebra* (CGA) [3][9][12][14][18]. The first CGA subspace, called CGA1, is

$$\mathbf{e}_i \cdot \mathbf{e}_j = \begin{cases} 1 & : \ i = j, 1 \le i \le 4 \\ -1 & : \ i = j = 5 \\ 0 & : \ i \ne j. \end{cases} \tag{1}$$

The second CGA subspace, called CGA2, is

$$\mathbf{e}_i \cdot \mathbf{e}_j = \begin{cases} 1 & : \ i = j, 6 \le i \le 9 \\ -1 & : \ i = j = 10 \\ 0 & : \ i \ne j. \end{cases} \tag{2}$$

1. Revised version vA, *October 2, 2015*, with some minor corrections and improvements.

2. DCGA is the work of an independent research by this author. No prior published work on DCGA was consulted or known to this author at the time of original research and publication of this paper (v1) on *August 11, 2015* on viXra.org.

The metric we use for $\mathcal{G}_{8,2}$ is $[1,1,1,1,-1,1,1,1,1,-1]$. This metric makes it very simple to use the CGA subalgebras of the CGA1 and CGA2 subspaces in a way fully compatible with CGA. These two CGAs are used as mirror copies or doubles to create *bivector-valued entities* for points and surfaces. As such, the $\mathcal{G}_{8,2}$ geometric algebra of these new bivector-valued point and surface entities could be called *Double-Conformal Geometric Algebra* (DCGA), or even *Bi-conformal Geometric Algebra* (Bi-CGA or 2-CGA).

The $\mathcal{G}_{8,2}$ DCGA surface entities include all of the types of surface entities available in $\mathcal{G}_{4,1}$ CGA and in the $\mathcal{G}_{6,3}$ geometric algebra [19] known as $\mathcal{G}_{6,3}$ *Quadric Geometric Algebra* (QGA) [4][10]. DCGA also includes a new toroid (torus) surface entity. More generally, DCGA has entities for quartic cyclide surfaces, including Darboux cyclides and Dupin cyclides based on torus and quadric surface inversions in spheres.

All DCGA entities, both GIPNS and their GOPNS duals, can be rotated, isometrically dilated, and translated by versor operations on the entities. The DCGA rotor, dilator, and translator are each a new form of bi-versor or double-versor of multivector-grade four.

The *DCGA entities* will refer to *all* entities of the DCGA algebra, and the *bi-CGA entities* will refer specifically to the CGA-subset of entities (points, lines, circles, planes, and spheres) as they exist in DCGA. The CGA1 and CGA2 entities also exist within their CGA subalgebras, offering the flexibility to use them separately, or also to double them to form bi-CGA entities when it is desired to bring them into the larger DCGA space.

Compared to QGA, DCGA extends CGA in a new way and has a new toroid entity and general cyclide entities. While QGA can only rotate the CGA 6,3D entities, DCGA provides a rotor that can rotate all DCGA entities around any axis by any angle. While QGA has an isotropic dilator and methods for anisotropic dilation, DCGA has only an isotropic dilator. Both QGA and DCGA have a translator for translations of all entities. While QGA supports intersecting all QGA GIPNS entities, DCGA can intersect all DCGA GIPNS entities only with the subset of bi-CGA GIPNS entities including spheres, planes, lines, and circles.

Depending on the needs of a particular application, DCGA may provide a larger set of operations and entities than QGA. As with QGA, there may be performance issues when working with a high-dimensional Clifford algebra such as DCGA. For applications where anisotropic dilation is not required, but where rotation of all surfaces is required and intersecting them with CGA entities is sufficient, then DCGA provides a powerful geometric algebra with all of the standard operations as versors.

2 CGA1 and CGA2

The CGA1 and CGA2 entities follow the ordinary $\mathcal{G}_{4,1}$ *Conformal Geometric Algebra* (CGA), and these subalgebras would be easily available to an application that is also using the $\mathcal{G}_{8,2}$ algebra. This subsection gives a quick review of CGA.

2.1 CGA point

In $\mathcal{G}_{4,1}$ *Conformal Geometric Algebra* (CGA), the embedding of a 3D point $\mathbf{p} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$ in $\mathbb{R}^3$ starts with a stereographic embedding of $\mathbf{p}$ onto a hypersphere or 3-sphere $\mathbb{S}^3$ using $\mathbf{e}_4$ as the stereographic 3-sphere pole. As shown in Figure 1, this requires finding the intersection of the line through $\mathbf{e}_4$ and $\mathbf{p}$ with the 3-sphere. The vectors $\mathbf{e}_4$ and $\mathbf{p}$ are perpendicular, and we can treat the embedding of $\mathbf{p}$ similarly to a 1D axis embedding into a stereographic 1-sphere or circle.

The identities

$$\begin{aligned} |\mathbf{p}| &= \sqrt{x^2+y^2+z^2} \\ \hat{\mathbf{p}} &= \frac{\mathbf{p}}{|\mathbf{p}|} \\ \mathbf{p} &= |\mathbf{p}|\hat{\mathbf{p}} \\ \mathbf{p}^2 &= |\mathbf{p}|^2 = x^2+y^2+z^2 \end{aligned}$$

are used in the following.

The stereographic embedding of $|\mathbf{p}|\hat{\mathbf{p}}$ is the intersection $\alpha\hat{\mathbf{p}}+\beta\mathbf{e}_4$ of the unit 3-circle on the $\hat{\mathbf{p}}\mathbf{e}_4$-hyperplane with the line through $\mathbf{e}_4$ and $|\mathbf{p}|\hat{\mathbf{p}}$. The Minkowski homogenization is $\alpha\hat{\mathbf{p}}+\beta\mathbf{e}_4+\mathbf{e}_5$. The point at the origin embeds to $\mathbf{e}_o=-\mathbf{e}_4+\mathbf{e}_5$ and the point at infinity embeds to $\mathbf{e}_\infty=\mathbf{e}_4+\mathbf{e}_5$. It is convenient to scale $\mathbf{e}_o$ as $\mathbf{e}_o=\frac{1}{2}(-\mathbf{e}_4+\mathbf{e}_5)$ such that $\mathbf{e}_o\cdot\mathbf{e}_\infty=-1$. The values for α and β are solved as follows.

The initial relations are the unit circle $\alpha^2+\beta^2=1$ and, by similar triangles, the line $\frac{1-\beta}{\alpha}=\frac{1}{|\mathbf{p}|}$.

$$\begin{aligned} \alpha^2 &= 1-\beta^2=(1+\beta)(1-\beta)=((1-\beta)|\mathbf{p}|)^2 \\ (1+\beta) &= (1-\beta)|\mathbf{p}|^2 \\ \beta|\mathbf{p}|^2+\beta &= |\mathbf{p}|^2-1 \\ \beta &= \frac{|\mathbf{p}|^2-1}{|\mathbf{p}|^2+1} \end{aligned}$$

$$\begin{aligned} \alpha &= (1-\beta)|\mathbf{p}| \\ &= \left(1-\frac{|\mathbf{p}|^2-1}{|\mathbf{p}|^2+1}\right)|\mathbf{p}| \\ &= \left(\frac{|\mathbf{p}|^2+1}{|\mathbf{p}|^2+1}-\frac{|\mathbf{p}|^2-1}{|\mathbf{p}|^2+1}\right)|\mathbf{p}| \\ &= \frac{2|\mathbf{p}|}{|\mathbf{p}|^2+1} \end{aligned}$$

The stereographic embedding of $|\mathbf{p}|\hat{\mathbf{p}}$, denoted $\mathcal{S}(|\mathbf{p}|\hat{\mathbf{p}})$, can now be written as

$$\begin{aligned} \mathcal{S}(|\mathbf{p}|\hat{\mathbf{p}}) &= \alpha\hat{\mathbf{p}}+\beta\mathbf{e}_4 \\ &= \left(\frac{2|\mathbf{p}|}{|\mathbf{p}|^2+1}\right)\hat{\mathbf{p}}+\left(\frac{|\mathbf{p}|^2-1}{|\mathbf{p}|^2+1}\right)\mathbf{e}_4. \end{aligned} \tag{3}$$

The homogenization of $\mathcal{S}(|\mathbf{p}|\hat{\mathbf{p}})$, denoted $\mathcal{H}_M(\mathcal{S}(|\mathbf{p}|\hat{\mathbf{p}}))$, can be written as

$$\mathbf{P}=\mathcal{H}_M(\mathcal{S}(|\mathbf{p}|\hat{\mathbf{p}})) = \left(\frac{2|\mathbf{p}|}{|\mathbf{p}|^2+1}\right)\hat{\mathbf{p}}+\left(\frac{|\mathbf{p}|^2-1}{|\mathbf{p}|^2+1}\right)\mathbf{e}_4+\mathbf{e}_5. \tag{4}$$

Since this point entity $\mathbf{P}$ is homogeneous, and $|\mathbf{p}|^2+1$ is never zero, it can be scaled by $\frac{|\mathbf{p}|^2+1}{2}$ to define $\mathbf{P}$ as

$$\begin{aligned} \mathbf{P} &= \mathcal{H}_M(\mathcal{S}(|\mathbf{p}|\hat{\mathbf{p}}))\simeq\mathcal{C}_{4,1}(\mathbf{p}) \\ &= |\mathbf{p}|\hat{\mathbf{p}}+\frac{|\mathbf{p}|^2-1}{2}\mathbf{e}_4+\frac{|\mathbf{p}|^2+1}{2}\mathbf{e}_5 \\ &= |\mathbf{p}|\hat{\mathbf{p}}+\frac{|\mathbf{p}|^2}{2}(\mathbf{e}_4+\mathbf{e}_5)+\frac{1}{2}(-\mathbf{e}_4+\mathbf{e}_5) \\ &= \mathbf{p}+\frac{1}{2}\mathbf{p}^2(\mathbf{e}_4+\mathbf{e}_5)+\frac{1}{2}(-\mathbf{e}_4+\mathbf{e}_5). \end{aligned} \tag{5}$$

When $|\mathbf{p}| = 0$,

$$\mathbf{P}_{|\mathbf{p}|=0} = \frac{1}{2}(-\mathbf{e}_4 + \mathbf{e}_5) = \mathbf{e}_o \tag{6}$$

representing the point at the origin. In the limit as $|\mathbf{p}| \to \pm\infty$, we find that

$$\mathbf{P}_{|\mathbf{p}|\to\infty} = \mathbf{e}_4 + \mathbf{e}_5 = \mathbf{e}_\infty \tag{7}$$

represents the point at infinity. By taking inner products, it can be shown that these points are all null vectors on a null 4-cone, and the inner product $\mathbf{e}_o \cdot \mathbf{e}_\infty = -1$. The CGA embedding of vector $\mathbf{p}$ as CGA point $\mathbf{P}$ can now be defined as

$$\mathbf{P} = \mathbf{P}_\mathcal{C} = \mathcal{C}(\mathbf{p}) = \mathcal{C}_{4,1}(\mathbf{p}) = \mathbf{p} + \frac{1}{2}\mathbf{p}^2\mathbf{e}_\infty + \mathbf{e}_o. \tag{8}$$

The projection of a CGA point $\mathbf{P}_\mathcal{C}$ back to a vector $\mathbf{p}$ is

$$\mathbf{p} = \frac{(\mathbf{P}_\mathcal{C} \cdot \mathbf{I}_3)\mathbf{I}_3^\sim}{-\mathbf{P}_\mathcal{C} \cdot \mathbf{e}_\infty} \tag{9}$$

where $\mathbf{I}_3 = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$ is the Euclidean 3D unit pseudoscalar.

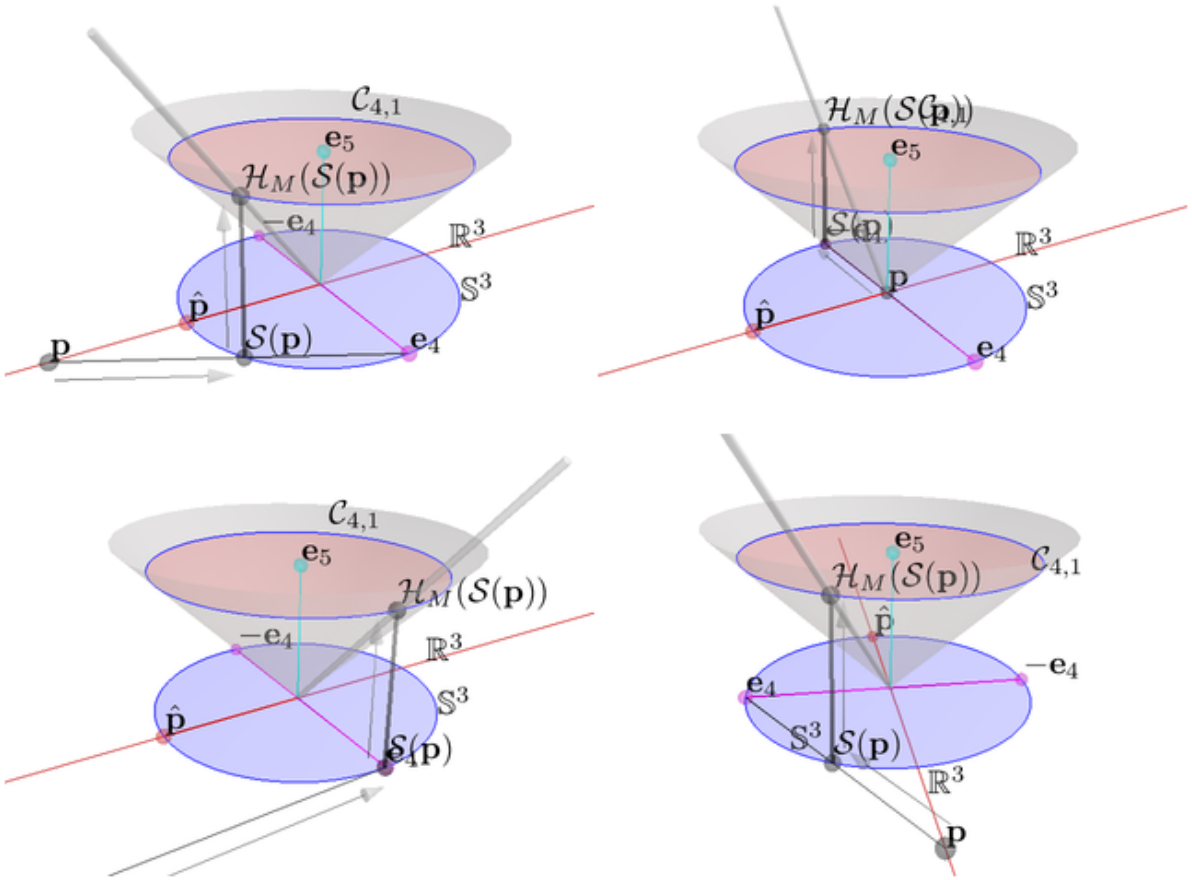

Figure 1. CGA point embedding

Figure 1 shows the CGA embedding procedure. The upper-left image shows a 3D vector $\mathbf{p}$ in $\mathbb{R}^3 = \mathcal{G}_3^1$ that is intersected with the 3-sphere $\mathbb{S}^3$, similar to an ordinary circle, at $\mathcal{S}(\mathbf{p})$ and then raised or homogenized to $\mathcal{H}_M(\mathcal{S}(\mathbf{p})) = \mathcal{C}(\mathbf{p})$ as the CGA embedding of $\mathbf{p}$. The *null cone* is the space of all homogeneous CGA points. A CGA point may be arbitrarily scaled along a line in the null cone without affecting the point being represented. A normalized CGA point has its $\mathbf{e}_o$ component scaled to 1. The upper-right image shows $\mathbf{p}$ at the origin where it is embedded as $2\mathbf{e}_o = -\mathbf{e}_4 + \mathbf{e}_5$; but after scaling this by $\frac{1}{2}$ to our preferred normalization, then it embeds to $\mathcal{C}(\mathbf{p}) = \mathbf{e}_o = \frac{1}{2}(-\mathbf{e}_4 + \mathbf{e}_5)$. The lower-left image shows $\mathbf{p}$ moved very far from the origin off screen, and it approaches the embedding $\mathcal{C}(\mathbf{p}) = \mathbf{e}_\infty = \mathbf{e}_4 + \mathbf{e}_5$ as it moves to an infinite distance from the origin in any direction. The lower-right image shows $\mathbf{p}$ at a relatively negative position where it embeds into a different quadrant of the null cone. The figure was generated using the program *CLUCalc* written by CHRISTIAN PERWASS, and is a modification of Fig 4.14 in [12].

2.2 CGA GIPNS surfaces

A CGA point $\mathbf{T}_\mathcal{C} = \mathcal{C}(\mathbf{t}) = \mathcal{C}_{4,1}(\mathbf{t})$ is on a CGA *geometric inner product null space* (GIPNS) surface $\mathbf{S}$ if $\mathbf{T}_\mathcal{C} \cdot \mathbf{S} = 0$ [12].

2.2.1 CGA GIPNS sphere

The CGA GIPNS 1-vector *sphere* $\mathbf{S}$, centered at CGA point $\mathbf{P}_\mathcal{C}$ with radius r, is defined as

$$\mathbf{S} \;=\; \mathbf{P}_\mathcal{C} - \frac{1}{2} r^2 \mathbf{e}_\infty. \tag{10}$$

2.2.2 CGA GIPNS plane

The CGA GIPNS 1-vector *plane* $\mathbf{\Pi}$, normal to unit vector $\mathbf{n}$ at distance d from the origin, is defined as

$$\mathbf{\Pi} \;=\; \mathbf{n} + d\mathbf{e}_\infty. \tag{11}$$

2.2.3 CGA GIPNS line

The CGA GIPNS 2-vector *line* $\mathbf{L}$, in the direction of the unit vector $\mathbf{d}$, perpendicular to $\mathbf{D} = \mathbf{d}^{*\mathcal{E}} = \mathbf{d}/\mathbf{I}_\mathcal{E}$, and through 3D point $\mathbf{p}$, is defined as

$$\mathbf{L} \;=\; \mathbf{D} - (\mathbf{p} \cdot \mathbf{D})\mathbf{e}_\infty. \tag{12}$$

The Euclidean 3D pseudoscalar is $\mathbf{I}_\mathcal{E} = \mathbf{I}_3 = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$, and the Euclidean 3D dual of any multivector $\mathbf{d}$ in this space is defined as $\mathbf{d}^{*\mathcal{E}} = \mathbf{d}/\mathbf{I}_\mathcal{E}$.

2.2.4 CGA GIPNS circle

The CGA GIPNS 2-vector *circle* $\mathbf{C}$ is defined as

$$\mathbf{C} \;=\; \mathbf{S} \wedge \mathbf{\Pi} \tag{13}$$

which is the intersection of a sphere $\mathbf{S}$ and plane $\mathbf{\Pi}$.

2.3 CGA GOPNS surfaces

A CGA point $\mathbf{T}_{\mathcal{C}} = \mathcal{C}(\mathbf{t}) = \mathcal{C}_{4,1}(\mathbf{t})$ is on a CGA *geometric outer product null space* (GOPNS) surface $\mathbf{S}^{*\mathcal{C}}$ if $\mathbf{T}_{\mathcal{C}} \wedge \mathbf{S}^{*\mathcal{C}} = 0$ [12].

2.3.1 CGA GOPNS sphere

The CGA GOPNS 4-vector *sphere* $\mathbf{S}^{*\mathcal{C}}$ is the wedge of four CGA points $\mathbf{P}_{\mathcal{C}_i}$ on the sphere

$$\begin{aligned} \mathbf{S}^{*\mathcal{C}} &= \mathbf{P}_{\mathcal{C}_1} \wedge \mathbf{P}_{\mathcal{C}_2} \wedge \mathbf{P}_{\mathcal{C}_3} \wedge \mathbf{P}_{\mathcal{C}_4} \\ &= \mathbf{S}/\mathbf{I}_{\mathcal{C}} \end{aligned} \tag{14}$$

and is the CGA dual of the CGA GIPNS 1-vector sphere $\mathbf{S}$.

The CGA unit pseudoscalar is $\mathbf{I}_{\mathcal{C}} = \mathbf{I}_5 = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4\mathbf{e}_5$, and the CGA dual of any multivector $\mathbf{S}$ in this space is defined as $\mathbf{S}^{*\mathcal{C}} = \mathbf{S}/\mathbf{I}_{\mathcal{C}}$. The CGA GOPNS and CGA GIPNS entities are CGA duals of each other.

2.3.2 CGA GOPNS plane

The CGA GOPNS 4-vector *plane* $\mathbf{\Pi}^{*\mathcal{C}}$ is the wedge of three CGA points $\mathbf{P}_{\mathcal{C}_i}$ on the plane and the point $\mathbf{e}_\infty$

$$\begin{aligned} \mathbf{\Pi}^{*\mathcal{C}} &= \mathbf{P}_{\mathcal{C}_1} \wedge \mathbf{P}_{\mathcal{C}_2} \wedge \mathbf{P}_{\mathcal{C}_3} \wedge \mathbf{e}_\infty \\ &= \mathbf{\Pi}/\mathbf{I}_{\mathcal{C}} \end{aligned} \tag{15}$$

and is the CGA dual of CGA GIPNS 1-vector plane $\mathbf{\Pi}$.

2.3.3 CGA GOPNS line

The CGA GOPNS 3-vector *line* $\mathbf{L}^{*\mathcal{C}}$ is the wedge of two CGA points $\mathbf{P}_{\mathcal{C}_i}$ on the line and the point $\mathbf{e}_\infty$

$$\begin{aligned} \mathbf{L}^{*\mathcal{C}} &= \mathbf{P}_{\mathcal{C}_1} \wedge \mathbf{P}_{\mathcal{C}_2} \wedge \mathbf{e}_\infty \\ &= \mathbf{L}/\mathbf{I}_{\mathcal{C}} \end{aligned} \tag{16}$$

and is the CGA dual of the CGA GIPNS 2-vector line $\mathbf{L}$.

2.3.4 CGA GOPNS circle

The CGA GOPNS 3-vector *circle* $\mathbf{C}^{*\mathcal{C}}$ is the wedge of three CGA points $\mathbf{P}_{\mathcal{C}_i}$ on the circle

$$\begin{aligned} \mathbf{C}^{*\mathcal{C}} &= \mathbf{P}_{\mathcal{C}_1} \wedge \mathbf{P}_{\mathcal{C}_2} \wedge \mathbf{P}_{\mathcal{C}_3} \\ &= \mathbf{C}/\mathbf{I}_{\mathcal{C}} \end{aligned} \tag{17}$$

and is the CGA dual of the CGA GIPNS 2-vector circle $\mathbf{C}$.

2.4 CGA operations

The rotor R, dilator D, and translator T are called *versors*. Their operation on a CGA entity $\mathbf{X}$ has the form $\mathbf{X}' = O\mathbf{X}O^{-1}$, called a *versor operation*. The versor O of the operation often has an exponential form which can be expanded by Taylor series into circular trigonometric, hyperbolic trigonometric, or dual number form.

2.4.1 CGA rotor

A *rotor* is a *rot*ation operat*or*, or vers*or*. The CGA rotor R, for rotation around unit vector axis $\mathbf{n}$ by θ radians, is defined as

$$\begin{aligned} R &= \cos\left(\frac{1}{2}\theta\right) + \sin\left(\frac{1}{2}\theta\right)\mathbf{n}^{*\mathcal{E}} \\ &= e^{\frac{1}{2}\theta\mathbf{n}^{*\mathcal{E}}} = e^{\frac{1}{2}\theta\mathbf{N}}. \end{aligned} \tag{18}$$

The unit bivector $\mathbf{N} = \mathbf{n}^{*\mathcal{E}}$ represents the plane of rotation. Any multivector in $\mathcal{G}_3$ or any CGA entity $\mathbf{X}$ in $\mathcal{G}_{4,1}$ is rotated as

$$\begin{aligned} \mathbf{X}' &= R\mathbf{X}R^{\sim} \\ &= R\mathbf{X}R^{-1} \end{aligned} \tag{19}$$

where $R^{\sim}$ is the reverse of R, and is also equal to the inverse R^{-1}. Rotation is also defined by reflection in two planes as

$$\mathbf{X}' = \mathbf{\Pi}_2\mathbf{\Pi}_1\mathbf{X}\mathbf{\Pi}_1\mathbf{\Pi}_2 \tag{20}$$

which rotates $\mathbf{X}$ by *twice* the angle between the planes from $\mathbf{\Pi}_1$ to $\mathbf{\Pi}_2$.

2.4.2 CGA dilator

A *dilator* is a *dilat*ion operat*or*. The CGA isotropic dilator D by factor d is defined as

$$\begin{aligned} D &= \frac{1}{2}(1+d) + \frac{1}{2}(1-d)\mathbf{e}_\infty \wedge \mathbf{e}_o \\ &\simeq 1 + \frac{(1-d)}{(1+d)}\mathbf{e}_\infty \wedge \mathbf{e}_o \\ &\simeq e^{\operatorname{atanh}\left(\frac{(1-d)}{(1+d)}\right)\mathbf{e}_\infty \wedge \mathbf{e}_o} = e^{-\frac{1}{2}\ln(d)\mathbf{e}_\infty \wedge \mathbf{e}_o}. \end{aligned} \tag{21}$$

Any CGA entity $\mathbf{X}$ in $\mathcal{G}_{4,1}$ is dilated by the factor d as

$$\mathbf{X}' = D\mathbf{X}D^{\sim}. \tag{22}$$

The first form of D is the most applicable since it allows $d < 0$ and usually gives the expected results in that case. The forms of D using atanh or ln cannot accept $d \leq 0$ since those functions would return an infinite result for $d = 0$ or complex numbers which are not valid in a geometric algebra over real numbers.

It should be noted that a zero dilation factor $d = 0$ is *generally not valid*. Entities having $\mathbf{e}_o$ or its dual $\mathbf{e}_o^{*\mathcal{C}}$ as a term will dilate by factor 0 into $\mathbf{e}_o$ or its dual (up to scale), which is a *valid* result. All other entities dilate by factor 0 into the scalar 0, which is an *invalid* result. Dilation by factor 0 is valid on the CGA GIPNS *sphere* and CGA *point* and their duals. Using $d = 0$ in the first form of D gives

$$D = \frac{1}{2} + \frac{1}{2}\mathbf{e}_\infty \wedge \mathbf{e}_o$$

which, as can be checked, dilates any normalized CGA GIPNS sphere $\mathbf{S}$ or CGA point $\mathbf{P}_\mathcal{C}$ into

$$D\mathbf{S}D^{\sim} = D\mathbf{P}_\mathcal{C}D^{\sim} = D\mathbf{e}_oD^{\sim} = \mathbf{e}_o.$$

A DCGA entity must have the DCGA origin point $\mathbf{e}_o$ or its dual $\mathbf{e}_o^{*\mathcal{D}}$ as a term for DCGA dilation by factor $d=0$ to be valid. This is explained further in the section on the DCGA GIPNS cyclide.

Dilation is also defined by inversions in two concentric spheres as

$$\mathbf{X}' = \mathbf{S}_2\mathbf{S}_1\mathbf{X}\mathbf{S}_1\mathbf{S}_2 \tag{23}$$

which dilates by $d=\frac{r_2^2}{r_1^2}$, with radius r_1 of $\mathbf{S}_1$ and radius r_2 of $\mathbf{S}_2$. The dilator D is derived from successive inversions in two spheres centered at the origin, but it is also possible for the spheres to be centered at any point and to dilate relative to that point. The inversion of an entity $\mathbf{X}$ in just one sphere $\mathbf{S}$ as $\mathbf{X}'=\mathbf{S}\mathbf{X}\mathbf{S}$ produces the entity $\mathbf{X}'$ that is $\mathbf{X}$ reflected in the sphere. The sphere $\mathbf{S}$ can be visualized as a mirrored surface and $\mathbf{X}'$ is the image of $\mathbf{X}$ in $\mathbf{S}$. The inversion of a surface in a sphere also turns the surface inside-out and is sometimes called the inverse surface, especially if the sphere is the unit sphere at the origin.

2.4.3 CGA translator

A *translator* is a *translat*ion operat*or*. The CGA translator T by a vector $\mathbf{d} = d_x\mathbf{e}_1 + d_y\mathbf{e}_2 + d_z\mathbf{e}_3$ is defined as

$$\begin{aligned} T &= 1-\frac{1}{2}\mathbf{d}\mathbf{e}_\infty \\ &= e^{-\frac{1}{2}\mathbf{d}\mathbf{e}_\infty}. \end{aligned} \tag{24}$$

Any CGA entity $\mathbf{X}$ is translated by the vector $\mathbf{d}$ as

$$\mathbf{X} = T\mathbf{X}T^{\sim}. \tag{25}$$

Translation is also defined by reflection in two parallel planes as

$$\mathbf{X}' = \mathbf{\Pi}_2\mathbf{\Pi}_1\mathbf{X}\mathbf{\Pi}_1\mathbf{\Pi}_2 \tag{26}$$

which translates by *twice* the vector $\mathbf{d}=(d_2-d_1)\mathbf{n}$, with common normal unit vector $\mathbf{n}$ of each plane (they are parallel) and plane distances from origin d_1 and d_2 of planes $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$, respectively.

2.4.4 CGA motor

A *motor* is a *mot*ion operat*or*. A rotation around a unit vector axis $\mathbf{n}$, followed by a translation parallel to $\mathbf{n}$ are commutative operations. Either the translation or the rotation can be done first, and the other second, to reach the same final position. This commutative operation, being a screw or helical motion, can be seen physically without mathematics. The motor is a special case where the commutative rotor and translator can be composed into a single versor M with an *exponential form* as

$$\begin{aligned} M &= RT = TR \\ &= e^{\frac{1}{2}\theta\mathbf{n}^{*\mathcal{E}}}e^{-\frac{1}{2}d\mathbf{n}\mathbf{e}_\infty} = e^{-\frac{1}{2}d\mathbf{n}\mathbf{e}_\infty}e^{\frac{1}{2}\theta\mathbf{n}^{*\mathcal{E}}} \\ &= e^{\frac{1}{2}\theta\mathbf{n}^{*\mathcal{E}}-\frac{1}{2}d\mathbf{n}\mathbf{e}_\infty} \\ &= e^{-\frac{1}{2}\mathbf{n}(\theta\mathbf{I}_\mathcal{E}+d\mathbf{e}_\infty)}. \end{aligned} \tag{27}$$

The exponents or *logarithms* of commutative exponentials can be added. A motor can be used to model smoothly-interpolated *screw*, *twistor*, or helical motions, performed in n steps using the nth root of M

$$M^{\frac{1}{n}} = e^{-\frac{1}{2n}\mathbf{n}(\theta\mathbf{I}_{\mathcal{E}}+d\mathbf{e}_{\infty})} \tag{28}$$

applied at each step.

2.4.5 CGA intersection

CGA GIPNS *intersection* entities which represent the surface intersections of two or more CGA GIPNS entities are formed by the wedge of the CGA GIPNS entities. The CGA GIPNS circle is defined as a CGA GIPNS intersection entity $\mathbf{C}=\mathbf{S}\wedge\mathbf{\Pi}$.

Almost any combination of CGA GIPNS entities may be wedged to form a CGA GIPNS intersection entity up to grade 4, except that the CGA GIPNS 2-vector line and circle entities that are coplanar cannot be intersected unless their common plane is first contracted out of each of them, then the common plane is wedged back onto their intersection entity.

Like any CGA GIPNS entity, a CGA GIPNS intersection entity $\mathbf{X}$ can be taken dual as $\mathbf{X}^{*\mathcal{C}}=\mathbf{X}/\mathbf{I}_{\mathcal{C}}$ into its CGA GOPNS intersection entity $\mathbf{X}^{*\mathcal{C}}$.

De Morgan's law for the intersection $\mathbf{X}$ of two objects $\mathbf{A}$ and $\mathbf{B}$ is

$$\mathbf{X} = \text{not}((\text{not}\,\mathbf{A})\,\text{and}\,(\text{not}\,\mathbf{B}))$$

and translates into the CGA intersection

$$\mathbf{X}^{*\mathcal{C}} = (\mathbf{A}^{*\mathcal{C}}\wedge\mathbf{B}^{*\mathcal{C}})^{*\mathcal{C}}. \tag{29}$$

This is just the creation of the CGA GOPNS intersection entity $\mathbf{X}^{*\mathcal{C}}$ of two *CGA GOPNS entities* $\mathbf{A}$ and $\mathbf{B}$. In this case, $\mathbf{A}^{*\mathcal{C}}$ and $\mathbf{B}^{*\mathcal{C}}$ are the *undual* CGA GIPNS entities, which can then be intersected by wedge product. The CGA GIPNS intersection $\mathbf{X}$ is then dualized as the CGA GOPNS entity $\mathbf{X}^{*\mathcal{C}}$. The classical view of intersections is by working with spanning objects, which are the CGA GOPNS entities.

2.4.6 CGA dualization

The Euclidean 3D or $\mathcal{G}_3$ unit pseudoscalar $\mathbf{I}_{\mathcal{E}}$ is defined as

$$\begin{aligned}
\mathbf{I}_{\mathcal{E}}=\mathbf{I}_3 &= \mathbf{e}_1\wedge\mathbf{e}_2\wedge\mathbf{e}_3=\mathbf{e}_1\mathbf{e}_2\mathbf{e}_3 \\
\mathbf{I}_{\mathcal{E}}^{\sim} &= (-1)^{3(3-1)/2}\mathbf{I}_{\mathcal{E}}=-\mathbf{I}_{\mathcal{E}} \\
\mathbf{I}_{\mathcal{E}}^{2} &= -\mathbf{I}_{\mathcal{E}}\mathbf{I}_{\mathcal{E}}^{\sim}=-1 \\
\mathbf{I}_{\mathcal{E}}^{-1} &= \mathbf{I}_{\mathcal{E}}^{\sim}=-\mathbf{I}_{\mathcal{E}}
\end{aligned} \tag{30}$$

and is the dualization operator on multivectors in $\mathcal{G}_3$. A blade $\mathbf{B}\in\mathcal{G}_3^k$ of grade k is taken to its Euclidean 3D or $\mathcal{G}_3$ dual $\mathbf{B}^{*\mathcal{E}}\in\mathcal{G}_3^{3-k}$ of grade $3-k$ as

$$\mathbf{B}^{*\mathcal{E}} = \mathbf{B}/\mathbf{I}_{\mathcal{E}}=-\mathbf{B}\cdot\mathbf{I}_{\mathcal{E}}. \tag{31}$$

Duals represent the same objects from two converse spatial spans, and the duals have different behavior as operators or algebraic factors on other multivectors. The dual of a unit vector is a unit bivector that can act as the unit of a rotor around the vector, but a unit vector can only operate as a reflector through the vector.

The CGA or $\mathcal{G}_{4,1}$ unit pseudoscalar $\mathbf{I}_{\mathcal{C}}$ is defined as

$$\begin{aligned} \mathbf{I}_{\mathcal{C}} = \mathbf{I}_5 &= \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \wedge \mathbf{e}_4 \wedge \mathbf{e}_5 = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4\mathbf{e}_5 \\ \mathbf{I}_{\mathcal{C}}^{\sim} &= (-1)^{5(5-1)/2}\mathbf{I}_{\mathcal{C}} = \mathbf{I}_{\mathcal{C}} \\ \mathbf{I}_{\mathcal{C}}^2 &= \mathbf{I}_{\mathcal{C}}\mathbf{I}_{\mathcal{C}}^{\sim} = -1 \\ \mathbf{I}_{\mathcal{C}}^{-1} &= -\mathbf{I}_{\mathcal{C}}^{\sim} = -\mathbf{I}_{\mathcal{C}} \end{aligned} \tag{32}$$

and is the dualization operator on CGA entities that takes CGA GIPNS entities to or from CGA GOPNS entities. A CGA entity $\mathbf{X} \in \mathcal{G}_{4,1}^k$ of grade k is taken to its CGA dual entity $\mathbf{X}^{*\mathcal{C}} \in \mathcal{G}_{4,1}^{5-k}$ of grade $5-k$ as

$$\mathbf{X}^{*\mathcal{C}} = \mathbf{X}/\mathbf{I}_{\mathcal{C}} = -\mathbf{X} \cdot \mathbf{I}_{\mathcal{C}}. \tag{33}$$

The pseudoscalar $\mathbf{I}_{\mathcal{C}}$ does not represent any CGA entity, so no CGA GOPNS entity nor CGA GIPNS intersection entity can have grade 5. The max grade of a CGA entity is grade 4.

2.5 CGA1 and CGA2 notations

The CGA1 and CGA2 spaces are used as exact copies of CGA. All that is needed is a little notation to separate the two spaces.

Multivectors in the $\mathcal{G}_3$ subspace of the CGA1 space will use the subscript $\mathcal{E}^1$. For example, a Euclidean 3D vector $\mathbf{p}$ in the CGA1 space is denoted in the form

$$\mathbf{p}_{\mathcal{E}^1} = p_x\mathbf{e}_1 + p_y\mathbf{e}_2 + p_z\mathbf{e}_3. \tag{34}$$

A CGA entity in the CGA1 space will use the subscript $\mathcal{C}^1$. For example, the embedding of $\mathbf{p}_{\mathcal{E}^1}$ as a CGA1 point $\mathbf{P}_{\mathcal{C}^1}$ is denoted

$$\begin{aligned} \mathbf{P}_{\mathcal{C}^1} &= \mathcal{C}^1(\mathbf{p}_{\mathcal{E}^1}) \\ &= \mathbf{p}_{\mathcal{E}^1} + \frac{1}{2}\mathbf{p}_{\mathcal{E}^1}^2\mathbf{e}_{\infty 1} + \mathbf{e}_{o1} \end{aligned} \tag{35}$$

where

$$\mathbf{e}_{\infty 1} = (\mathbf{e}_4 + \mathbf{e}_5) \tag{36}$$

$$\mathbf{e}_{o1} = \frac{1}{2}(-\mathbf{e}_4 + \mathbf{e}_5). \tag{37}$$

The CGA1 point embedding function has been named $\mathcal{C}^1$. Likewise, a CGA1 surface entity is named $\mathbf{X}_{\mathcal{C}^1}$. The CGA1 point at the origin $\mathbf{e}_{o1}$ and point at infinity $\mathbf{e}_{\infty 1}$ are named with suffix 1 to indicate their version as being the CGA1 versions.

Multivectors in the $\mathcal{G}_3$ subspace of the CGA2 space will use the subscript $\mathcal{E}^2$ (e.g., $\mathbf{p}_{\mathcal{E}^2}$). A CGA entity in the CGA2 space will use the subscript $\mathcal{C}^2$ (e.g., $\mathbf{X}_{\mathcal{C}^2}$).

With this notation, the CGA1 unit pseudoscalars are named as

$$\mathbf{I}_{\mathcal{E}^1} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3 \tag{38}$$

$$\mathbf{I}_{\mathcal{C}^1} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4\mathbf{e}_5 \tag{39}$$

and the CGA2 unit pseudoscalars are named as

$$\mathbf{I}_{\mathcal{E}^2} = \mathbf{e}_6\mathbf{e}_7\mathbf{e}_8 \tag{40}$$

$$\mathbf{I}_{\mathcal{C}^2} = \mathbf{e}_6\mathbf{e}_7\mathbf{e}_8\mathbf{e}_9\mathbf{e}_{10}. \tag{41}$$

A Euclidean 3D vector $\mathbf{p}$ in the CGA2 space is denoted in the form

$$\mathbf{p}_{\mathcal{E}^2} = p_x\mathbf{e}_6 + p_y\mathbf{e}_7 + p_z\mathbf{e}_8. \tag{42}$$

The CGA2 point embedding is

$$\begin{aligned}\mathbf{P}_{\mathcal{C}^2} &= \mathcal{C}^2(\mathbf{p}_{\mathcal{E}^2}) \\ &= \mathbf{p}_{\mathcal{E}^2} + \frac{1}{2}\mathbf{p}_{\mathcal{E}^2}^2\mathbf{e}_{\infty 2} + \mathbf{e}_{o2}\end{aligned} \tag{43}$$

where

$$\mathbf{e}_{\infty 2} = (\mathbf{e}_9 + \mathbf{e}_{10}) \tag{44}$$

$$\mathbf{e}_{o2} = \frac{1}{2}(-\mathbf{e}_9 + \mathbf{e}_{10}). \tag{45}$$

The CGA2 point at the origin $\mathbf{e}_{o2}$ and point at infinity $\mathbf{e}_{\infty 2}$ are named with suffix 2 to indicate their version as being the CGA2 versions.

A versor O (rotor, dilator, translator, or motor) that is in the CGA1 space is denoted $O_{\mathcal{C}^1}$, and if it is in the CGA2 space it is denoted $O_{\mathcal{C}^2}$.

3 DCGA point

The standard DCGA *null* 2-vector *point* entity $\mathbf{P}_{\mathcal{D}}$ is the embedding of a vector

$$\mathbf{p}_{\mathcal{E}^1} = \mathbf{p} = p_x\mathbf{e}_1 + p_y\mathbf{e}_2 + p_z\mathbf{e}_3 \tag{46}$$

as

$$\begin{aligned}\mathbf{P}_{\mathcal{D}} &= \mathcal{D}(\mathbf{p}) \\ &= \mathcal{C}^1(\mathbf{p}_{\mathcal{E}^1}) \wedge \mathcal{C}^2(\mathbf{p}_{\mathcal{E}^2}) \\ &= \mathbf{P}_{\mathcal{C}^1} \wedge \mathbf{P}_{\mathcal{C}^2}\end{aligned} \tag{47}$$

where

$$\begin{aligned}\mathbf{p}_{\mathcal{E}^2} &= (\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{e}_1)\mathbf{e}_6 + (\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{e}_2)\mathbf{e}_7 + (\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{e}_3)\mathbf{e}_8 \\ &= p_x\mathbf{e}_6 + p_y\mathbf{e}_7 + p_z\mathbf{e}_8.\end{aligned} \tag{48}$$

The DCGA point $\mathbf{P}_{\mathcal{D}}$, which could be called a *double point*, is the wedge of a CGA1 point $\mathbf{P}_{\mathcal{C}^1}$ with a CGA2 point $\mathbf{P}_{\mathcal{C}^2}$, which are the CGA embeddings of the same Euclidean 3D vector $\mathbf{p}$ into each CGA.

CGA1 and CGA2 points and surface entities can be rotated, translated, and dilated using CGA1 and CGA2 versors for these operations. The wedge of a CGA1 versor with its copy CGA2 versor (rotor, translator, dilator, or motor) creates the DCGA versor on DCGA points and surface entities. The DCGA versors could be called *double versors* or bi-CGA versors.

The DCGA point at the origin $\mathbf{e}_o$ is defined as

$$\mathbf{e}_o = \mathbf{e}_{o1} \wedge \mathbf{e}_{o2}. \tag{49}$$

The DCGA point at infinity $\mathbf{e}_\infty$ is defined as

$$\mathbf{e}_\infty = \mathbf{e}_{\infty 1} \wedge \mathbf{e}_{\infty 2}. \tag{50}$$

As in CGA, these DCGA points also have the inner product

$$\mathbf{e}_\infty \cdot \mathbf{e}_o \;=\; -1. \tag{51}$$

All DCGA points are null 2-vectors, $\mathbf{P}_\mathcal{D}^2 = 0$. However, compared to CGA, all values are squared and this changes the formulas for the metrical results known in CGA. For example, the squared-squared distance d^4 between two DCGA points $\mathbf{P}_{\mathcal{D}_1}$ and $\mathbf{P}_{\mathcal{D}_2}$ is

$$\begin{aligned} d^4 &= -4\mathbf{P}_{\mathcal{D}_1} \cdot \mathbf{P}_{\mathcal{D}_2} \\ &= -4(\mathbf{P}_{\mathcal{C}_1^1} \wedge \mathbf{P}_{\mathcal{C}_1^2}) \cdot (\mathbf{P}_{\mathcal{C}_2^1} \wedge \mathbf{P}_{\mathcal{C}_2^2}) \\ &= -4(\mathbf{P}_{\mathcal{C}_1^1} \cdot ((\mathbf{P}_{\mathcal{C}_1^2} \cdot \mathbf{P}_{\mathcal{C}_2^1})\mathbf{P}_{\mathcal{C}_2^2} - \mathbf{P}_{\mathcal{C}_2^1}(\mathbf{P}_{\mathcal{C}_1^2} \cdot \mathbf{P}_{\mathcal{C}_2^2}))) \\ &= -4((\mathbf{P}_{\mathcal{C}_1^2} \cdot \mathbf{P}_{\mathcal{C}_2^1})(\mathbf{P}_{\mathcal{C}_1^1} \cdot \mathbf{P}_{\mathcal{C}_2^2}) - (\mathbf{P}_{\mathcal{C}_1^1} \cdot \mathbf{P}_{\mathcal{C}_2^1})(\mathbf{P}_{\mathcal{C}_1^2} \cdot \mathbf{P}_{\mathcal{C}_2^2})) \\ &= -4\left((0)(0) - \left(\frac{-d^2}{2}\right)\left(\frac{-d^2}{2}\right)\right). \end{aligned} \tag{52}$$

The squared distance d^2 between points is also

$$\begin{aligned} d^2 &= -2\frac{-\mathbf{P}_{\mathcal{D}_1} \cdot \mathbf{e}_{\infty 2}}{(\mathbf{P}_{\mathcal{D}_1} \cdot \mathbf{e}_{\infty 2}) \cdot \mathbf{e}_{\infty 1}} \cdot \frac{-\mathbf{P}_{\mathcal{D}_2} \cdot \mathbf{e}_{\infty 2}}{(\mathbf{P}_{\mathcal{D}_2} \cdot \mathbf{e}_{\infty 2}) \cdot \mathbf{e}_{\infty 1}} \\ &= -2\mathbf{P}_{\mathcal{C}_1^1} \cdot \mathbf{P}_{\mathcal{C}_2^1} \end{aligned} \tag{53}$$

where each DCGA point is contracted and renormalized into CGA1 points.

The projection of a DCGA point $\mathbf{P}_\mathcal{D}$ back to a vector $\mathbf{p}$ is

$$\mathbf{P}_{\mathcal{C}^1} \;\simeq\; \mathbf{P}_\mathcal{D} \cdot \mathbf{e}_{\infty 2} \tag{54}$$

$$\mathbf{p} \;=\; \frac{(\mathbf{P}_{\mathcal{C}^1} \cdot \mathbf{I}_{\mathcal{E}^1})\mathbf{I}_{\mathcal{E}^1}^{\sim}}{-\mathbf{P}_{\mathcal{C}^1} \cdot \mathbf{e}_{\infty 1}}. \tag{55}$$

The DCGA 2-vector point $\mathbf{P}_\mathcal{D}$ allows for the extraction of more polynomial terms than only the x, y, z, x^2, y^2, z^2 terms that CGA or QGA 1-vector points allow. The terms that can be extracted from a point determine what polynomial equations or entities that can be represented as GIPNS entities that test against the point.

When expanded, the DCGA point $\mathbf{T}_\mathcal{D} = \mathcal{D}(\mathbf{t})$ is

$$\begin{aligned} \mathbf{T}_\mathcal{D} &= \left(\mathbf{t}_{\mathcal{E}^1} + \frac{1}{2}\mathbf{t}^2\mathbf{e}_{\infty 1} + \mathbf{e}_{o1}\right) \wedge \left(\mathbf{t}_{\mathcal{E}^2} + \frac{1}{2}\mathbf{t}^2\mathbf{e}_{\infty 2} + \mathbf{e}_{o2}\right) \\ &= \mathbf{t}_{\mathcal{E}^1} \wedge \mathbf{t}_{\mathcal{E}^2} + \mathbf{t}_{\mathcal{E}^1} \wedge \mathbf{e}_{o2} + \mathbf{e}_{o1} \wedge \mathbf{t}_{\mathcal{E}^2} + \\ &\quad \frac{1}{2}\mathbf{t}^2\mathbf{e}_{\infty 1} \wedge (\mathbf{t}_{\mathcal{E}^2} + \mathbf{e}_{o2}) + \frac{1}{2}\mathbf{t}^2(\mathbf{t}_{\mathcal{E}^1} + \mathbf{e}_{o1}) \wedge \mathbf{e}_{\infty 2} + \\ &\quad \frac{1}{4}\mathbf{t}^4\mathbf{e}_\infty + \mathbf{e}_o \end{aligned} \tag{56}$$

where

$$\mathbf{t} = \mathbf{t}_{\mathcal{E}^1} \;=\; x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3 \tag{57}$$

$$\mathbf{t}_{\mathcal{E}^2} \;=\; x\mathbf{e}_6 + y\mathbf{e}_7 + z\mathbf{e}_8 \tag{58}$$

$$\mathbf{t}^2 \;=\; x^2 + y^2 + z^2 \tag{59}$$

$$\mathbf{t}^4 \;=\; x^4 + y^4 + z^4 + 2x^2y^2 + 2y^2z^2 + 2z^2x^2. \tag{60}$$

Fully expanding, $\mathbf{T}_\mathcal{D}$ is

$$\begin{aligned} \mathbf{T}_\mathcal{D} \;=\; & \frac{x}{2}\left(x^2+y^2+z^2-1\right)\mathbf{e}_1 \wedge \mathbf{e}_9 + \frac{x}{2}\left(x^2+y^2+z^2+1\right)\mathbf{e}_1 \wedge \mathbf{e}_{10} + \\ & \frac{x}{2}\left(x^2+y^2+z^2-1\right)\mathbf{e}_4 \wedge \mathbf{e}_6 + \frac{x}{2}\left(x^2+y^2+z^2+1\right)\mathbf{e}_5 \wedge \mathbf{e}_6 + \\ & \frac{y}{2}\left(x^2+y^2+z^2-1\right)\mathbf{e}_2 \wedge \mathbf{e}_9 + \frac{y}{2}\left(x^2+y^2+z^2+1\right)\mathbf{e}_2 \wedge \mathbf{e}_{10} + \end{aligned} \tag{61}$$

$$
\begin{aligned}
&\frac{y}{2}\left(x^2+y^2+z^2-1\right)\mathbf{e}_4\wedge\mathbf{e}_7+\frac{y}{2}\left(x^2+y^2+z^2+1\right)\mathbf{e}_5\wedge\mathbf{e}_7+\\
&\frac{z}{2}\left(x^2+y^2+z^2-1\right)\mathbf{e}_3\wedge\mathbf{e}_9+\frac{z}{2}\left(x^2+y^2+z^2+1\right)\mathbf{e}_3\wedge\mathbf{e}_{10}+\\
&\frac{z}{2}\left(x^2+y^2+z^2-1\right)\mathbf{e}_4\wedge\mathbf{e}_8+\frac{z}{2}\left(x^2+y^2+z^2+1\right)\mathbf{e}_5\wedge\mathbf{e}_8+\\
&x\,y\,\mathbf{e}_1\wedge\mathbf{e}_7+x\,y\,\mathbf{e}_2\wedge\mathbf{e}_6+\\
&y\,z\,\mathbf{e}_2\wedge\mathbf{e}_8+y\,z\,\mathbf{e}_3\wedge\mathbf{e}_7+\\
&x\,z\mathbf{e}_1\wedge\mathbf{e}_8+x\,z\,\mathbf{e}_3\wedge\mathbf{e}_6+\\
&x^2\mathbf{e}_1\wedge\mathbf{e}_6+y^2\,\mathbf{e}_2\wedge\mathbf{e}_7+z^2\,\mathbf{e}_3\wedge\mathbf{e}_8+\\
&\left(\frac{x^4}{4}+\frac{x^2\,y^2}{2}+\frac{x^2\,z^2}{2}+\frac{y^4}{4}+\frac{y^2\,z^2}{2}+\frac{z^4}{4}-\frac{1}{4}\right)\mathbf{e}_4\wedge\mathbf{e}_{10}+\\
&\left(\frac{x^4}{4}+\frac{x^2\,y^2}{2}+\frac{x^2\,z^2}{2}+\frac{y^4}{4}+\frac{y^2\,z^2}{2}+\frac{z^4}{4}-\frac{1}{4}\right)\mathbf{e}_5\wedge\mathbf{e}_9+\\
&\left(\frac{x^4}{4}+\frac{x^2\,y^2}{2}+\frac{x^2\,z^2}{2}-\frac{x^2}{2}+\frac{y^4}{4}+\frac{y^2\,z^2}{2}-\frac{y^2}{2}+\frac{z^4}{4}-\frac{z^2}{2}+\frac{1}{4}\right)\mathbf{e}_4\wedge\mathbf{e}_9+\\
&\left(\frac{x^4}{4}+\frac{x^2\,y^2}{2}+\frac{x^2\,z^2}{2}+\frac{x^2}{2}+\frac{y^4}{4}+\frac{y^2\,z^2}{2}+\frac{y^2}{2}+\frac{z^4}{4}+\frac{z^2}{2}+\frac{1}{4}\right)\mathbf{e}_5\wedge\mathbf{e}_{10}.
\end{aligned}
$$

The vector $\mathbf{t}$, and its DCGA point embedding $\mathbf{T}_{\mathcal{D}}=\mathcal{D}(\mathbf{t})$, will be used as a *test point* for position on surfaces. If we define the following value-extraction elements or operators on DCGA points,

$$
\begin{aligned}
T_x &= \frac{1}{2}(\mathbf{e}_1\wedge\mathbf{e}_{\infty 2}+\mathbf{e}_{\infty 1}\wedge\mathbf{e}_6) && (62)\\
T_y &= \frac{1}{2}(\mathbf{e}_2\wedge\mathbf{e}_{\infty 2}+\mathbf{e}_{\infty 1}\wedge\mathbf{e}_7) && (63)\\
T_z &= \frac{1}{2}(\mathbf{e}_3\wedge\mathbf{e}_{\infty 2}+\mathbf{e}_{\infty 1}\wedge\mathbf{e}_8) && (64)
\end{aligned}
$$

$$
\begin{aligned}
T_{xy} &= \frac{1}{2}(\mathbf{e}_7\wedge\mathbf{e}_1+\mathbf{e}_6\wedge\mathbf{e}_2) && (65)\\
T_{yz} &= \frac{1}{2}(\mathbf{e}_7\wedge\mathbf{e}_3+\mathbf{e}_8\wedge\mathbf{e}_2) && (66)\\
T_{zx} &= \frac{1}{2}(\mathbf{e}_8\wedge\mathbf{e}_1+\mathbf{e}_6\wedge\mathbf{e}_3) && (67)
\end{aligned}
$$

$$
\begin{aligned}
T_{x^2} &= \mathbf{e}_6\wedge\mathbf{e}_1 && (68)\\
T_{y^2} &= \mathbf{e}_7\wedge\mathbf{e}_2 && (69)\\
T_{z^2} &= \mathbf{e}_8\wedge\mathbf{e}_3 && (70)
\end{aligned}
$$

$$
\begin{aligned}
T_{x\mathbf{t}^2} &= (\mathbf{e}_1\wedge\mathbf{e}_{o2})+(\mathbf{e}_{o1}\wedge\mathbf{e}_6) && (71)\\
T_{y\mathbf{t}^2} &= (\mathbf{e}_2\wedge\mathbf{e}_{o2})+(\mathbf{e}_{o1}\wedge\mathbf{e}_7) && (72)\\
T_{z\mathbf{t}^2} &= (\mathbf{e}_3\wedge\mathbf{e}_{o2})+(\mathbf{e}_{o1}\wedge\mathbf{e}_8) && (73)
\end{aligned}
$$

$$
\begin{aligned}
T_1 &= -(\mathbf{e}_{\infty 1}\wedge\mathbf{e}_{\infty 2})=-\mathbf{e}_\infty && (74)\\
T_{\mathbf{t}^2} &= -(\mathbf{e}_{\infty 1}\wedge\mathbf{e}_{o2}+\mathbf{e}_{o1}\wedge\mathbf{e}_{\infty 2}) && (75)\\
T_{\mathbf{t}^4} &= -4(\mathbf{e}_{o1}\wedge\mathbf{e}_{o2})=-4\mathbf{e}_o && (76)
\end{aligned}
$$

then we can extract values from a DCGA point $\mathbf{T}_{\mathcal{D}}$ as $s = \mathbf{T}_{\mathcal{D}} \cdot T_s$. These extraction operators are used to define most of the DCGA GIPNS 2-vector surface entities. Two properties of these extractions are

$$\mathbf{e}_\infty \cdot T_s \;=\; \begin{cases} 0 \;:\; T_s \neq T_{\mathbf{t}^4} \\ 4 \;:\; T_s = T_{\mathbf{t}^4} \end{cases} \tag{77}$$

$$\mathbf{e}_o \cdot T_s \;=\; \begin{cases} 0 \;:\; T_s \neq T_1 \\ 1 \;:\; T_s = T_1. \end{cases} \tag{78}$$

The first property, about the point at infinity $\mathbf{e}_\infty$, has the consequence that all DCGA GIPNS 2-vector surface entities without a term in $T_{\mathbf{t}^4}$ are entities having the surface point $\mathbf{e}_\infty$. In particular, the DCGA GIPNS 2-vector ellipsoid surface entity is generally considered to be a finite closed surface, yet in DCGA it always has the surface point $\mathbf{e}_\infty$. Other surface entities can also unexpectedly have the surface point $\mathbf{e}_\infty$. This possible problem about $\mathbf{e}_\infty$ will be mentioned again in the section on the DCGA GIPNS 2-vector ellipsoid entity $\mathbf{E}$. This possible problem about $\mathbf{e}_\infty$ will also be discussed further in the sections on inversions in spheres and on cyclides. The second property, about the point at the origin $\mathbf{e}_o$, does not pose any known problems.

The spherical inverse surface entity $\mathbf{SES}^{\sim}$ of any surface entity $\mathbf{E}$ without a term in $T_{\mathbf{t}^4}$ will *always* have the inversion sphere $\mathbf{S}$ center point $\mathbf{P}_{\mathcal{D}}$ as a surface point. The point at infinity $\mathbf{e}_\infty$ always reflects into the inversion sphere center point $\mathbf{P}_{\mathcal{D}}$, or the reverse. All open surfaces are expected to have the point $\mathbf{e}_\infty$, and their inverse surfaces are expected to have the inversion sphere center point. Unexpectedly, the inverse surface entity of the ellipsoid entity when reflected in a sphere will always have a *singular outlier surface point* at the inversion sphere center point. A singular outlier point may be invisible in a surface plot.

4 DCGA GIPNS surfaces

The DCGA *geometric inner product null space* (GIPNS) surface entities are constructed using the value extractions $T_s \cdot \mathbf{T}_{\mathcal{D}}$ from the DCGA point entity. The DCGA GIPNS surface entities are the standard surface entities in DCGA since the direct construction of DCGA *geometric outer product null space* (GOPNS) surface entities is limited to the wedge of up to four DCGA points which cannot construct all of the DCGA GOPNS surface entities. The DCGA GIPNS surface entities can be rotated, dilated, and translated by DCGA versors, and they can be intersected with the bi-CGA GIPNS surface entities.

A DCGA test point $\mathbf{T}_{\mathcal{D}}$ that is on a DCGA GIPNS surface entity $\mathbf{S}$ must satisfy the GIPNS condition

$$\mathbf{T}_{\mathcal{D}} \cdot \mathbf{S} \;=\; 0.$$

The DCGA GIPNS k-vector surface entity $\mathbf{S}$ represents the set $\mathbb{NI}_G(\mathbf{S} \in \mathcal{G}^k_{8,2})$ of all 3D vector test points $\mathbf{t}$ that are surface points

$$\mathbb{NI}_G(\mathbf{S} \in \mathcal{G}^k_{8,2}) \;=\; \left\{\; \mathbf{t} \in \mathcal{G}^1_3 \;:\; (\mathcal{D}(\mathbf{t}) = \mathbf{T}_{\mathcal{D}}) \cdot \mathbf{S} = 0 \;\right\}.$$

4.1 DCGA GIPNS toroid

The implicit quartic equation for a circular toroid (torus), which is positioned at the origin and surrounds the z-axis, is

$$\mathbf{t}^4 + 2\mathbf{t}^2(R^2 - r^2) + (R^2 - r^2)^2 - 4R^2(x^2 + y^2) \;=\; 0 \tag{79}$$

where

$$\mathbf{t} \;=\; x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$$

is a test point, R is the major radius, and r is the minor radius. The equation is true if the test point $\mathbf{t}$ is on the toroid. The radius R is that of a circle around the origin in the xy-plane. The radius r is that of circles centered on the circle of R and which span the z-axis dimension for $z = \pm r$. The toroid spans $x, y = \pm(R + r)$.

The DCGA GIPNS 2-vector *toroid* surface entity $\mathbf{O}$ is defined as

$$\mathbf{O} \;=\; T_{\mathbf{t}^4} + 2(R^2 - r^2)T_{\mathbf{t}^2} + (R^2 - r^2)^2 T_1 - 4R^2(T_{x^2} + T_{y^2}). \tag{80}$$

A test DCGA point $\mathbf{T}_{\mathcal{D}} = \mathcal{D}(\mathbf{t})$ is on the toroid surface represented by $\mathbf{O}$ if $\mathbf{T}_{\mathcal{D}} \cdot \mathbf{O} = 0$. Using symbolic mathematics software, such as the *Geometric Algebra Module* [1] for *Sympy* [17] by Alan Bromborsky et al., the inner product $\mathbf{T}_{\mathcal{D}} \cdot \mathbf{O}$ generates the *scalar* implicit surface function of the toroid when $\mathbf{t}$ is a variable symbolic vector. When $\mathbf{t}$ is a specific vector, $\mathbf{T}_{\mathcal{D}} \cdot \mathbf{O}$ is a test operation on the toroid for the specific point.

We can denote the *DCGA-dual* of $\mathbf{O}$ as $\mathbf{O}^{*\mathcal{D}}$, and define it as

$$\mathbf{O}^{*\mathcal{D}} \;=\; \mathbf{O}/\mathbf{I}_{\mathcal{D}} = \mathbf{O}\mathbf{I}_{\mathcal{D}}^{-1} = -\mathbf{O} \cdot \mathbf{I}_{\mathcal{D}}. \tag{81}$$

The DCGA GOPNS 8-vector *toroid* surface entity is $\mathbf{O}^{*\mathcal{D}}$, where a test point $\mathbf{t}$ on the surface must satisfy the GOPNS condition $\mathbf{T}_{\mathcal{D}} \wedge \mathbf{O}^{*\mathcal{D}} = 0$. Since $\mathbf{T}_{\mathcal{D}}$ is a 2-vector and $\mathbf{O}^{*\mathcal{D}}$ is an 8-vector, then $\mathbf{T}_{\mathcal{D}} \wedge \mathbf{O}^{*\mathcal{D}}$ is the DCGA 10-vector *pseudoscalar* implicit surface function of the toroid when $\mathbf{t}$ is a variable symbolic vector. The *undual* operation returns the DCGA GIPNS surface $\mathbf{O} = \mathbf{O}^{*\mathcal{D}} \cdot \mathbf{I}_{\mathcal{D}}$. The other DCGA GOPNS surface entities will be discussed later in this paper.

Although the toroid $\mathbf{O}$ is created at the origin and aligned around the z-axis, it can then be rotated, dilated, and translated away from the origin using DCGA versor operations. Like all DCGA GIPNS surface entities, the DCGA GIPNS toroid can be intersected with any bi-CGA GIPNS (2, 4, or 6)-vector surface, which are 2-vector spheres and planes, 4-vector circles and lines, and 6-vector point-pairs.

Since the toroid $\mathbf{O}$ is constructed with an extraction term $T_{\mathbf{t}^4} = -4\mathbf{e}_o$, it is a DCGA *closed-surface* entity that does not include $\mathbf{e}_\infty$ as a surface point, and it can be dilated by a zero dilation factor $d = 0$ into $\mathbf{e}_o$. The inverse toroid entity, when reflected in a standard DCGA GIPNS 2-vector sphere, does not have a singular outlier surface point at the center point of the inversion sphere. The standard DCGA GIPNS 2-vector sphere $\mathbf{S}$ also has these closed-surface characteristics, but the ellipsoid $\mathbf{E}$ does not.

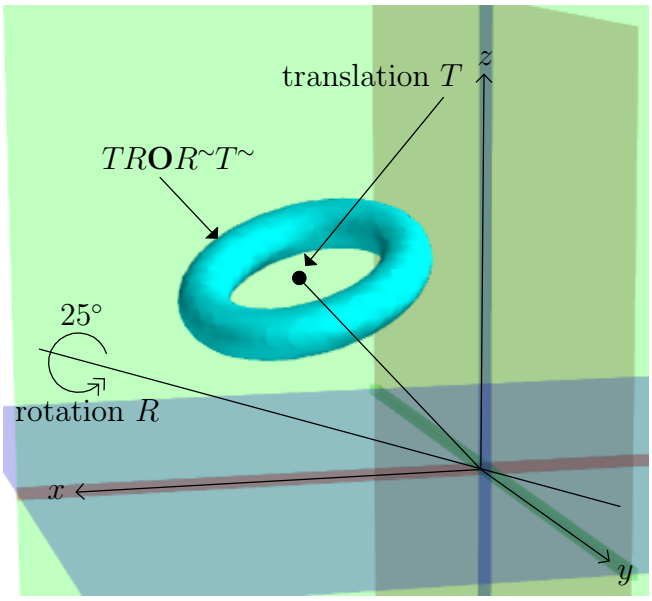

Figure 2. DCGA toroid rotated and translated

4.2 DCGA GIPNS ellipsoid

The implicit quadric equation for a principal axes-aligned ellipsoid is

$$\frac{(x-p_x)^2}{r_x^2}+\frac{(y-p_y)^2}{r_y^2}+\frac{(z-p_x)^2}{r_z^2}-1 \;=\; 0 \tag{82}$$

where $\mathbf{p}=p_x\mathbf{e}_1+p_y\mathbf{e}_2+p_z\mathbf{e}_3$ is the position (or shifted origin, or center) of the ellipsoid, and r_x, r_y, r_z are the semi-diameters (often denoted a, b, c). Expanding the squares, the equation can be written as

$$\frac{-2p_xx}{r_x^2}+\frac{-2p_yy}{r_y^2}+\frac{-2p_zz}{r_z^2}+\left(\frac{x^2}{r_x^2}+\frac{y^2}{r_y^2}+\frac{z^2}{r_z^2}\right)+\left(\frac{p_x^2}{r_x^2}+\frac{p_y^2}{r_y^2}+\frac{p_z^2}{r_z^2}-1\right) \;=\; 0. \tag{83}$$

Using the DCGA point value-extraction elements, an ellipsoid equation can be constructed. This construction will be similar for the remaining surface entities that follow.

The DCGA GIPNS 2-vector *ellipsoid* surface entity $\mathbf{E}$ is defined as

$$\begin{aligned}\mathbf{E} \;=\; & \frac{-2p_xT_x}{r_x^2}+\frac{-2p_yT_y}{r_y^2}+\frac{-2p_zT_z}{r_z^2}+ \\ & \frac{T_{x^2}}{r_x^2}+\frac{T_{y^2}}{r_y^2}+\frac{T_{z^2}}{r_z^2}+ \\ & \left(\frac{p_x^2}{r_x^2}+\frac{p_y^2}{r_y^2}+\frac{p_z^2}{r_z^2}-1\right)T_1.\end{aligned} \tag{84}$$

A DCGA 2-vector point $\mathbf{T}_{\mathcal{D}}=\mathcal{D}(\mathbf{t})$ is tested against the DCGA 2-vector ellipsoid $\mathbf{E}$ as

$$\mathbf{T}_{\mathcal{D}}\cdot\mathbf{E}\begin{cases}<0 \;:\; \mathbf{t} \text{ is } \textit{inside} \text{ ellipsoid}\\ =0 \;:\; \mathbf{t} \text{ is } \textit{on} \text{ ellipsoid}\\ >0 \;:\; \mathbf{t} \text{ is } \textit{outside} \text{ ellipsoid.}\end{cases} \tag{85}$$

It was first mentioned in Section 3, on the DCGA point $\mathbf{T}_{\mathcal{D}}$ and value-extraction operators T_s, that the ellipsoid entity $\mathbf{E}$ has the **possible problem** that it includes the point at infinity $\mathbf{e}_\infty$ as a surface point according to the test just given above. We could *define* the

invariant test $\mathbf{e}_\infty \cdot \mathbf{E} = 0$ as an invalid test, or we could accept that $\mathbf{e}_\infty$ is a valid surface point of the particular *DCGA ellipsoid entity* $\mathbf{E}$ but not of ellipsoids in general.

An inverse ellipsoid surface entity, which is an ellipsoid entity $\mathbf{E}$ that has been reflected in a standard DCGA 2-vector sphere $\mathbf{S}$ as $\mathbf{SES}^{\sim}$, will always have a *singular outlier surface point* that is exactly the center point $\mathbf{P}_{\mathcal{D}}$ of the inversion sphere $\mathbf{S}$ and the test $\mathbf{P}_{\mathcal{D}} \cdot (\mathbf{SES}^{\sim}) = 0$ will always hold true. The point $\mathbf{e}_\infty$ on the ellipsoid entity $\mathbf{E}$ is reflected into $\mathbf{P}_{\mathcal{D}}$. The inverse ellipsoid surface entity $\mathbf{SES}^{\sim}$ is otherwise a correctly formed surface entity of one of the types that should be expected, which is either a quartic Darboux cyclide or a cubic parabolic cyclide. The outlier point is often invisible in surface plots. See Figure 17, which shows an ellipsoid reflection that produces a Darboux cyclide.

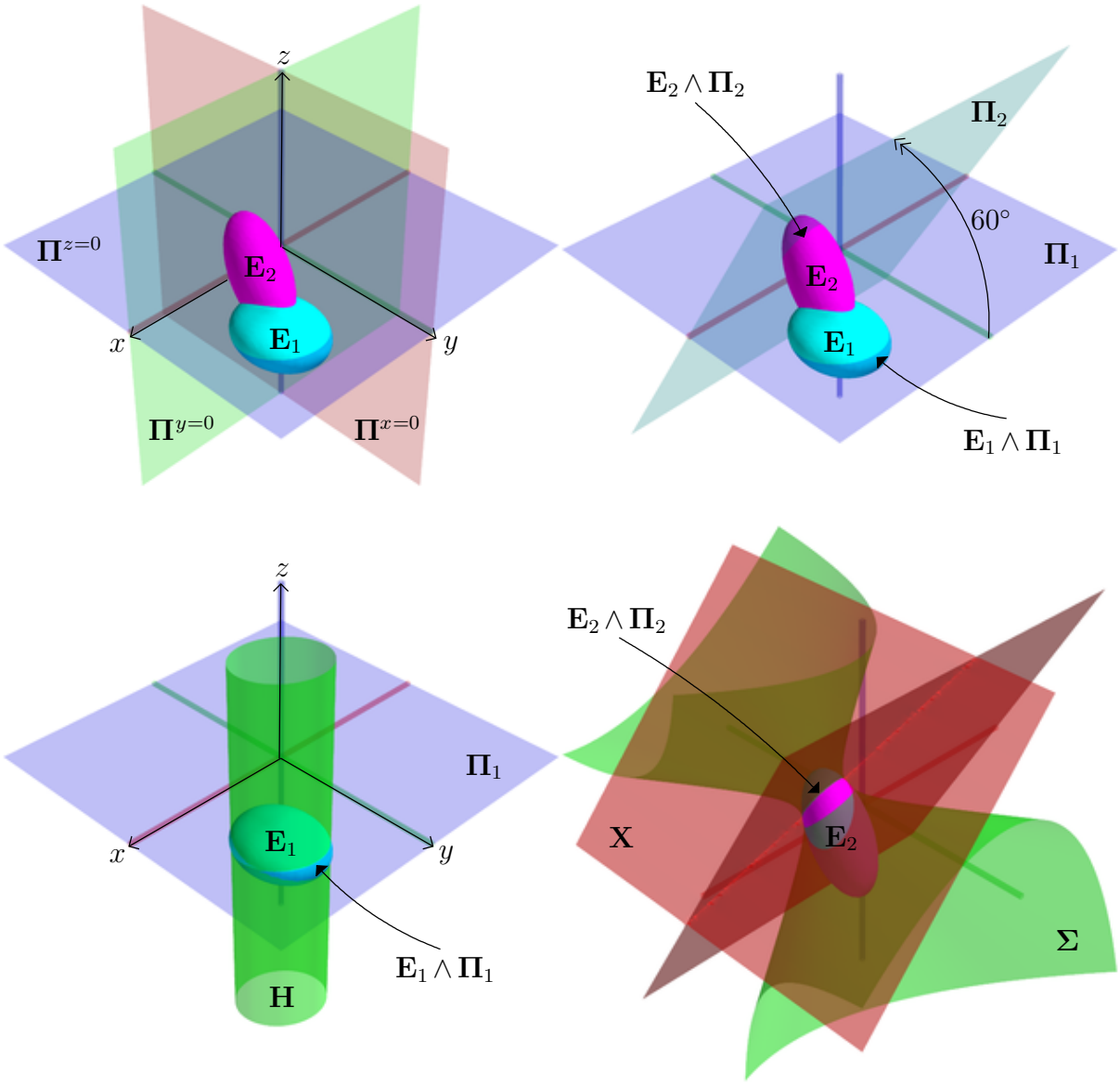

Figure 3. DCGA ellipsoids rotated, translated, and intersected with planes

Figure 3 shows two ellipsoids that have been rotated and translated into their intersecting positions using DCGA versor operations. The DCGA GIPNS ellipsoid $\mathbf{E}_1$ ($r_x = 4$, $r_y = 5$, $r_z = 3$) is rotated 25° around the line $\mathbf{n} = \frac{1}{\sqrt{2}}(-\mathbf{e}_1 + \mathbf{e}_2)$, then rotated 45° around the z-axis, then translated by $\mathbf{d} = 10\mathbf{e}_1 + 10\mathbf{e}_2$. The DCGA GIPNS ellipsoid $\mathbf{E}_2$ ($p_z = 6$,

$r_x = 2$, $r_y = 3$, $r_z = 6$) is rotated -35° around the line $\mathbf{n} = \frac{1}{\sqrt{2}}(-\mathbf{e}_1 + \mathbf{e}_2)$, then rotated 35° around the z-axis, then translated by $\mathbf{d} = 10\mathbf{e}_1 + 10\mathbf{e}_2$. The ellipsoids intersect in a curved ellipse which, unfortunately, could *not* be represented as an intersection entity.

Although not rigorously proved here, in tests performed by this author, the ellipsoid and all other DCGA entities can be intersected with the standard DCGA sphere, plane, line, and circle entities (bi-CGA entities), but DCGA entities cannot be intersected in full generality.

The upper-left image in Figure 3 shows the ellipsoids with standard planes drawn. The upper-right image shows the ellipsoids drawn with DCGA GIPNS plane $\mathbf{\Pi}_1$ representing the plane $z = 0$, and with the DCGA GIPNS plane $\mathbf{\Pi}_2$ representing the plane $z = 0$ rotated 60° around the x-axis. The lower-left image shows the DCGA GIPNS intersection entity $\mathbf{E}_1 \wedge \mathbf{\Pi}_1$; the green elliptic cylinder $\mathbf{H}$ is an intersection entity component and represents the ellipse in which they intersect. The lower-right image shows the DCGA GIPNS intersection entity $\mathbf{E}_2 \wedge \mathbf{\Pi}_2$; the green *hyperboloid of one sheet* $\mathbf{\Sigma}$ and the red *non-parallel planes pair* $\mathbf{X}$ are intersection entity components which are also coincident and represent the intersection.

4.3 DCGA GIPNS sphere

The standard DCGA GIPNS 2-vector *sphere* $\mathbf{S}$ will be defined as a bi-CGA sphere, *not* the DCGA GIPNS ellipsoid $\mathbf{E}$ with equal semi-diameters $r = r_x = r_y = r_z$.

The DCGA GIPNS ellipsoid $\mathbf{E}$ with $r = r_x = r_y = r_z$ can be *reformulated* into the DCGA GIPNS 2-vector *ellipsoid-based sphere* entity $\mathbf{\Theta}$ as

$$\begin{aligned} \mathbf{\Theta} &= -2(p_x T_x + p_y T_y + p_z T_z) + T_{x^2} + T_{y^2} + T_{z^2} + (p_x^2 + p_y^2 + p_z^2 - r^2)T_1 \\ &\simeq (p_x T_x + p_y T_y + p_z T_z) - \frac{1}{2}(p_x^2 + p_y^2 + p_z^2)T_1 - \frac{1}{2}(T_{x^2} + T_{y^2} + T_{z^2}) + \frac{1}{2}r^2 T_1. \end{aligned} \tag{86}$$

Taking $r = 0$ suggests that the sphere $\mathbf{\Theta}$ degenerates into some type of point entity. With $T_1 = -\mathbf{e}_\infty$, the middle term has a familiar CGA point form. However, if this were a CGA point, the last term should reduce to $\mathbf{e}_o$, but it does not. The result here is that, the sphere entity $\mathbf{\Theta}$ with $r = 0$ degenerates into a DCGA GIPNS *non-null* 2-vector *point* entity, which is not the standard DCGA *null* 2-vector *point* that we might expect. The DCGA GIPNS ellipsoid $\mathbf{E}$ can be reformulated into a kind of sphere entity $\mathbf{\Theta}$ that degenerates into a kind of non-null point entity when $r = 0$. However, $r = 0$ is *invalid* for an ellipsoid entity $\mathbf{E}$, and only in the limit $r \to 0$ does $\mathbf{E}$ approach a point $\mathbf{\Theta}$ with $r = 0$. We can also form a sphere in another way which *does* degenerate into a standard DCGA point.

The standard DCGA GIPNS 2-vector *sphere* surface entity $\mathbf{S}$, also being called a bi-CGA GIPNS 2-vector sphere, is defined as

$$\mathbf{S} = \mathbf{S}_{\mathcal{C}^1} \wedge \mathbf{S}_{\mathcal{C}^2} \tag{87}$$

where

$$\mathbf{S}_{\mathcal{C}^1} = \mathbf{P}_{\mathcal{C}^1} - \frac{1}{2}r^2\mathbf{e}_{\infty 1} \tag{88}$$

$$\mathbf{S}_{\mathcal{C}^2} = \mathbf{P}_{\mathcal{C}^2} - \frac{1}{2}r^2\mathbf{e}_{\infty 2}. \tag{89}$$

The CGA1 GIPNS sphere $\mathbf{S}_{\mathcal{C}^1}$ and the CGA2 GIPNS sphere $\mathbf{S}_{\mathcal{C}^2}$, both representing the same sphere, with radius r at center position $\mathbf{p}$ in our main 3D Euclidean space $\mathcal{E}^1$, are wedged to form the DCGA or bi-CGA GIPNS sphere $\mathbf{S}$. If $r = 0$, the sphere is degenerated

into a DCGA point

$$\mathbf{P}_{\mathcal{D}} \;=\; \mathbf{P}_{\mathcal{C}^1} \wedge \mathbf{P}_{\mathcal{C}^2} \tag{90}$$

that would represent the center position of the sphere. This form of sphere allows greater consistency, and it can also be intersected with any DCGA GIPNS entity. A sphere that is formed using the DCGA GIPNS ellipsoid can only be intersected with bi-CGA GIPNS entities. In general, the other bi-CGA GIPNS entities for lines, circles, and planes follow this same pattern, that they are the wedge of the CGA1 and CGA2 copies of the entity.

A DCGA 2-vector point $\mathbf{T}_{\mathcal{D}} = \mathcal{D}(\mathbf{t})$ is tested against the standard DCGA GIPNS 2-vector sphere $\mathbf{S}$ as

$$-2\left(\frac{-\mathbf{T}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 2}}{(\mathbf{T}_{\mathcal{D}} \cdot \mathbf{e}_{\infty 2}) \cdot \mathbf{e}_{\infty 1}}\right) \cdot \left(\frac{-\mathbf{S} \cdot \mathbf{e}_{\infty 2}}{(\mathbf{S} \cdot \mathbf{e}_{\infty 2}) \cdot \mathbf{e}_{\infty 1}}\right) \begin{cases} <0 \;:\; \mathbf{t} \text{ is } \textit{inside} \text{ sphere} \\ =0 \;:\; \mathbf{t} \text{ is } \textit{on} \text{ sphere} \\ >0 \;:\; \mathbf{t} \text{ is } \textit{outside} \text{ sphere} \\ >0 \;:\; = d^2, \text{ squared tangent.} \end{cases} \tag{91}$$

To determine inside or outside, this incidence test requires the bi-CGA point $\mathbf{T}_{\mathcal{D}}$ to be contracted into a CGA1 point, and the bi-CGA sphere $\mathbf{S}$ to be contracted into a CGA1 sphere, and both are renormalized. The entity $\mathbf{e}_{\infty 2}$ is both a CGA2 point and a CGA2 sphere of infinite radius, and it serves as the contraction operator on both the point and sphere into CGA1 entities, up to scale. The result is reduced to a CGA1 incidence test. When the test is positive, it is the squared distance d^2 from the point to the sphere along any line tangent to the sphere surface. Similarly for other bi-CGA entities, they can be contracted into CGA1 entities and then all the usual CGA tests are available on them.

4.4 DCGA GIPNS line

The DCGA GIPNS 4-vector *line* 1D surface entity $\mathbf{L}$ is defined as

$$\mathbf{L} \;=\; \mathbf{L}_{\mathcal{C}^1} \wedge \mathbf{L}_{\mathcal{C}^2} \tag{92}$$

where

$$\mathbf{L}_{\mathcal{C}^1} \;=\; \mathbf{D}_{\mathcal{E}^1} - (\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{D}_{\mathcal{E}^1})\mathbf{e}_{\infty 1} \tag{93}$$

$$\mathbf{L}_{\mathcal{C}^2} \;=\; \mathbf{D}_{\mathcal{E}^2} - (\mathbf{p}_{\mathcal{E}^2} \cdot \mathbf{D}_{\mathcal{E}^2})\mathbf{e}_{\infty 2}. \tag{94}$$

This is the wedge of the line as represented in CGA1 with the same line as represented in CGA2. It could also be called a bi-CGA GIPNS line entity. The $\mathbf{D}$ are unit bivectors perpendicular to the line, and $\mathbf{p}$ is any sample point on the line. The undual unit vector $\mathbf{d} = \mathbf{D}\mathbf{I}_3$, or $\mathbf{d}_{\mathcal{E}^1} = \mathbf{D}_{\mathcal{E}^1}\mathbf{I}_{\mathcal{E}^1}$ and $\mathbf{d}_{\mathcal{E}^2} = \mathbf{D}_{\mathcal{E}^2}\mathbf{I}_{\mathcal{E}^2}$, is in the direction of the line.

4.5 DCGA GIPNS plane

The DCGA GIPNS 2-vector *plane* surface entity $\mathbf{\Pi}$ is defined as

$$\mathbf{\Pi} \;=\; \mathbf{\Pi}_{\mathcal{C}^1} \wedge \mathbf{\Pi}_{\mathcal{C}^2} \tag{95}$$

where

$$\mathbf{\Pi}_{\mathcal{C}^1} \;=\; \mathbf{n}_{\mathcal{E}^1} + d\mathbf{e}_{\infty 1} \tag{96}$$

$$\mathbf{\Pi}_{\mathcal{C}^2} \;=\; \mathbf{n}_{\mathcal{E}^2} + d\mathbf{e}_{\infty 2}. \tag{97}$$

This is the wedge of the plane as represented in CGA1 with the same plane as represented in CGA2. It could also be called a bi-CGA GIPNS plane entity. The vector $\mathbf{n}$ is a unit vector perpendicular (normal) to the plane, and the scalar d is the distance of the plane from the origin.

The DCGA GIPNS 4-vector line $\mathbf{L}$ can also be defined as the intersection of two DCGA GIPNS planes as

$$\begin{aligned}
\mathbf{L} &= \mathbf{\Pi}_1 \wedge \mathbf{\Pi}_2 \\
&= (\mathbf{n}_{1\mathcal{E}^1} + d_1\mathbf{e}_{\infty 1}) \wedge (\mathbf{n}_{1\mathcal{E}^2} + d_1\mathbf{e}_{\infty 2}) \wedge (\mathbf{n}_{2\mathcal{E}^1} + d_2\mathbf{e}_{\infty 1}) \wedge (\mathbf{n}_{2\mathcal{E}^2} + d_2\mathbf{e}_{\infty 2}) \\
&= -((\mathbf{n}_{1\mathcal{E}^1} + d_1\mathbf{e}_{\infty 1}) \wedge (\mathbf{n}_{2\mathcal{E}^1} + d_2\mathbf{e}_{\infty 1})) \wedge ((\mathbf{n}_{1\mathcal{E}^2} + d_1\mathbf{e}_{\infty 2}) \wedge (\mathbf{n}_{2\mathcal{E}^2} + d_2\mathbf{e}_{\infty 2})) \\
&\simeq (\mathbf{n}_{1\mathcal{E}^1} \wedge \mathbf{n}_{2\mathcal{E}^1} - (d_1\mathbf{n}_{2\mathcal{E}^1} - d_2\mathbf{n}_{1\mathcal{E}^1})\mathbf{e}_{\infty 1}) \wedge (\mathbf{n}_{1\mathcal{E}^2} \wedge \mathbf{n}_{2\mathcal{E}^2} - (d_1\mathbf{n}_{2\mathcal{E}^2} - d_2\mathbf{n}_{1\mathcal{E}^2})\mathbf{e}_{\infty 2}) \\
&= (\mathbf{D}_{\mathcal{E}^1} - (\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{D}_{\mathcal{E}^1})\mathbf{e}_{\infty 1}) \wedge (\mathbf{D}_{\mathcal{E}^2} - (\mathbf{p}_{\mathcal{E}^2} \cdot \mathbf{D}_{\mathcal{E}^2})\mathbf{e}_{\infty 2}) \\
&= \mathbf{L}_{\mathcal{C}^1} \wedge \mathbf{L}_{\mathcal{C}^2}
\end{aligned} \tag{98}$$

where

$$\begin{aligned}
\mathbf{D}_{\mathcal{E}^1} &= \mathbf{n}_{1\mathcal{E}^1} \wedge \mathbf{n}_{2\mathcal{E}^1} \\
\mathbf{D}_{\mathcal{E}^2} &= \mathbf{n}_{1\mathcal{E}^2} \wedge \mathbf{n}_{2\mathcal{E}^2} \\
\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{D}_{\mathcal{E}^1} &= (\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{n}_{1\mathcal{E}^1})\mathbf{n}_{2\mathcal{E}^1} - (\mathbf{p}_{\mathcal{E}^1} \cdot \mathbf{n}_{2\mathcal{E}^1})\mathbf{n}_{1\mathcal{E}^1} \\
&= d_1\mathbf{n}_{2\mathcal{E}^1} - d_2\mathbf{n}_{1\mathcal{E}^1} \\
\mathbf{p}_{\mathcal{E}^2} \cdot \mathbf{D}_{\mathcal{E}^2} &= (\mathbf{p}_{\mathcal{E}^2} \cdot \mathbf{n}_{1\mathcal{E}^2})\mathbf{n}_{2\mathcal{E}^2} - (\mathbf{p}_{\mathcal{E}^2} \cdot \mathbf{n}_{2\mathcal{E}^2})\mathbf{n}_{1\mathcal{E}^2} \\
&= d_1\mathbf{n}_{2\mathcal{E}^2} - d_2\mathbf{n}_{1\mathcal{E}^2}
\end{aligned}$$

such that $\mathbf{p}$ is any point on both planes (the line), and $\mathbf{D} = \mathbf{d}^{*\mathcal{E}} = \mathbf{d}\,/\,\mathbf{I}_{\mathcal{E}}$ is the unit bivector perpendicular to the line. The unit vector $\mathbf{d} = \mathbf{D}\mathbf{I}_{\mathcal{E}}$ points in the direction of the line. Other bi-CGA GIPNS entities are formed similarly as the wedge of the entity in CGA1 with the same entity in CGA2.

Some of the subscripting notation may seem confusing. For example, $\mathbf{n}_{1\mathcal{E}^1}$ is the first of the two Euclidean 3D unit vectors in the CGA1 space, and this could also be denoted as $\mathbf{n}_{\mathcal{E}^1_1}$. Recall that $\mathcal{E}^1$ is the space of the unit pseudoscalar $\mathbf{I}_{\mathcal{E}^1} = \mathbf{I}_{3^1} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$ and it is a subspace of the $\mathcal{C}^1$ CGA1 space $\mathbf{I}_{\mathcal{C}^1} = \mathbf{I}_{5^1} = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4\mathbf{e}_5$. The CGA2 space uses notations $\mathbf{n}_{1\mathcal{E}^2}$ or $\mathbf{n}_{\mathcal{E}^2_1}$, where $\mathbf{I}_{\mathcal{E}^2} = \mathbf{I}_{3^2} = \mathbf{e}_6\mathbf{e}_7\mathbf{e}_8$ and $\mathbf{I}_{\mathcal{C}^2} = \mathbf{I}_{5^2} = \mathbf{e}_6\mathbf{e}_7\mathbf{e}_8\mathbf{e}_9\mathbf{e}_{10}$. The subscripting indicates the index number for multiple entities sharing the same name, and also the space in which the entity exists. Finally, $\mathbf{n}_{1\mathcal{E}^1}$ and $\mathbf{n}_{1\mathcal{E}^2}$ have the same index number 1, so they represent the same 3D unit vector $\mathbf{n}$ copied or doubled into the $\mathcal{E}^1$ and $\mathcal{E}^2$ Euclidean subspace of the $\mathcal{C}^1$ CGA1 and $\mathcal{C}^2$ CGA2 space, respectively.

4.6 DCGA GIPNS circle

A circle is the intersection of a sphere and plane. We can intersect a bi-CGA GIPNS 2-vector plane $\mathbf{\Pi}$ with either a bi-CGA GIPNS 2-vector sphere $\mathbf{S}$ or with a spherical DCGA GIPNS 2-vector ellipsoid $\mathbf{E}$ and get two different GIPNS 4-vector circle entities. The first can be intersected again with any other entity, but the latter can only be intersected again with another bi-CGA GIPNS entity.

Intersections are limited to an GIPNS intersection entity of maximum grade 8, so up to four 2-vector entities, two 4-vector entities, or a 4-vector entity and two 2-vector GIPNS entities can be intersected, but only one of the intersecting entities can be a non-biCGA quadric surface or toroid GIPNS entity.

As the standard DCGA GIPNS 4-vector *circle* 1D surface entity $\mathbf{C}$, we will define it as the bi-CGA GIPNS circle

$$\begin{aligned}
\mathbf{C} &= \mathbf{S} \wedge \mathbf{\Pi} \\
&= \mathbf{S}_{\mathcal{C}^1} \wedge \mathbf{S}_{\mathcal{C}^2} \wedge \mathbf{\Pi}_{\mathcal{C}^1} \wedge \mathbf{\Pi}_{\mathcal{C}^2}
\end{aligned} \tag{99}$$

$$\begin{aligned} &= -(\mathbf{S}_{\mathcal{C}^1} \wedge \mathbf{\Pi}_{\mathcal{C}^1}) \wedge (\mathbf{S}_{\mathcal{C}^2} \wedge \mathbf{\Pi}_{\mathcal{C}^2}) \\ &\simeq \mathbf{C}_{\mathcal{C}^1} \wedge \mathbf{C}_{\mathcal{C}^2}. \end{aligned}$$

4.7 DCGA GIPNS elliptic cylinder

An axes-aligned elliptic cylinder is the limit of an ellipsoid as one of the semi-diameters approaches ∞. This limit eliminates the terms of the cylinder axis from the implicit ellipsoid equation.

The x-axis aligned cylinder takes $r_x \to \infty$, reducing the ellipsoid equation to

$$\frac{(y-p_y)^2}{r_y^2} + \frac{(z-p_z)^2}{r_z^2} - 1 = 0. \tag{100}$$

Similarly, the y-axis and z-axis aligned cylinders are

$$\frac{(x-p_x)^2}{r_x^2} + \frac{(z-p_z)^2}{r_z^2} - 1 = 0 \tag{101}$$

$$\frac{(x-p_x)^2}{r_x^2} + \frac{(y-p_y)^2}{r_y^2} - 1 = 0 \tag{102}$$

where $\mathbf{p} = p_x\mathbf{e}_1 + p_y\mathbf{e}_2 + p_z\mathbf{e}_3$ is the position (or shifted origin, or center) of the ellipsoid, and r_x, r_y, r_z are the semi-diameters (often denoted a, b, c).

The DCGA GIPNS 2-vector *x,y,z-axis aligned cylinder* surface entities $\mathbf{H}^{\|\{x,y,z\}}$ are defined as

$$\mathbf{H}^{\|x} = \frac{-2p_yT_y}{r_y^2} + \frac{-2p_zT_z}{r_z^2} + \frac{T_{y^2}}{r_y^2} + \frac{T_{z^2}}{r_z^2} + \left(\frac{p_y^2}{r_y^2} + \frac{p_z^2}{r_z^2} - 1\right)T_1 \tag{103}$$

$$\mathbf{H}^{\|y} = \frac{-2p_xT_x}{r_x^2} + \frac{-2p_zT_z}{r_z^2} + \frac{T_{x^2}}{r_x^2} + \frac{T_{z^2}}{r_z^2} + \left(\frac{p_x^2}{r_x^2} + \frac{p_z^2}{r_z^2} - 1\right)T_1 \tag{104}$$

$$\mathbf{H}^{\|z} = \frac{-2p_xT_x}{r_x^2} + \frac{-2p_yT_y}{r_y^2} + \frac{T_{x^2}}{r_x^2} + \frac{T_{y^2}}{r_y^2} + \left(\frac{p_x^2}{r_x^2} + \frac{p_y^2}{r_y^2} - 1\right)T_1. \tag{105}$$

These elliptic cylinders are created as axes-aligned, but like all DCGA entities, they can be rotated, dilated, and translated using DCGA versor operations.

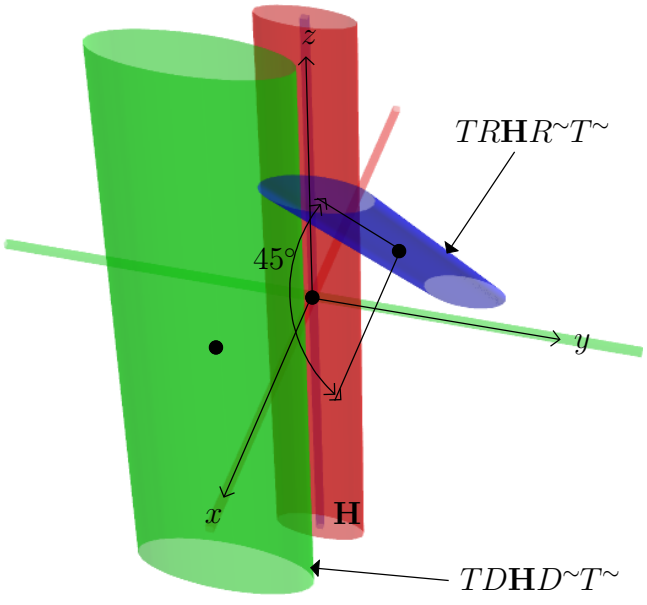

Figure 4. DCGA elliptic cylinders

Figure 4 shows a red DCGA GIPNS z-axis aligned elliptic cylinder $\mathbf{H}$ at the origin with semi-diameters $r_x = 1$ and $r_y = 3$. The green cylinder is the red cylinder dilated by factor 2 and translated $5\mathbf{e}_1 - 5\mathbf{e}_2$ using DCGA versors. The blue cylinder is the red cylinder rotated 45° around the y-axis and translated $-5\mathbf{e}_1 + 5\mathbf{e}_2$.

4.8 DCGA GIPNS elliptic cone

An axis-aligned elliptic cone is an axis-aligned cylinder that is linearly scaled along the axis.

The implicit quadric equation for an x-axis aligned cone is

$$\frac{(y-p_y)^2}{r_y^2} + \frac{(z-p_z)^2}{r_z^2} - \frac{(x-p_x)^2}{r_x^2} = 0. \tag{106}$$

where $\mathbf{p} = p_x\mathbf{e}_1 + p_y\mathbf{e}_2 + p_z\mathbf{e}_3$ is the position (or shifted origin, or center) of the cone apex, and r_x, r_y, r_z are the semi-diameters (often denoted a, b, c) of the ellipsoid upon which the cone is based. When

$$\frac{(x-p_x)^2}{r_x^2} = 1 \tag{107}$$

the cross section of the cone is the size of the similar cylinder. When $x = p_x$ the cross section of the cone is degenerated into the cone apex point.

Similarly, the implicit equations for y-axis and z-axis aligned cones are

$$\frac{(x-p_x)^2}{r_x^2} + \frac{(z-p_z)^2}{r_z^2} - \frac{(y-p_y)^2}{r_y^2} = 0 \tag{108}$$

$$\frac{(x-p_x)^2}{r_x^2} + \frac{(y-p_y)^2}{r_y^2} - \frac{(z-p_z)^2}{r_z^2} = 0. \tag{109}$$

The GIPNS cone entities are constructed similarly to the ellipsoid and cylinder entities.

The DCGA GIPNS 2-vector *{x,y,z}-axis aligned elliptic cone* surface entities $\mathbf{K}^{||\{x,y,z\}}$ are defined as

$$\mathbf{K}^{||x} = 2\left(\frac{p_xT_x}{r_x^2} - \frac{p_yT_y}{r_y^2} - \frac{p_zT_z}{r_z^2}\right) - \frac{T_{x^2}}{r_x^2} + \frac{T_{y^2}}{r_y^2} + \frac{T_{z^2}}{r_z^2} + \left(\frac{p_y^2}{r_y^2} + \frac{p_z^2}{r_z^2} - \frac{p_x^2}{r_x^2}\right)T_1 \tag{110}$$

$$\mathbf{K}^{||y} = 2\left(\frac{p_yT_y}{r_y^2} - \frac{p_zT_z}{r_z^2} - \frac{p_xT_x}{r_x^2}\right) + \frac{T_{x^2}}{r_x^2} - \frac{T_{y^2}}{r_y^2} + \frac{T_{z^2}}{r_z^2} + \left(\frac{p_x^2}{r_x^2} - \frac{p_y^2}{r_y^2} + \frac{p_z^2}{r_z^2}\right)T_1 \tag{111}$$

$$\mathbf{K}^{||z} = 2\left(\frac{p_zT_z}{r_z^2} - \frac{p_yT_y}{r_y^2} - \frac{p_xT_x}{r_x^2}\right) + \frac{T_{x^2}}{r_x^2} + \frac{T_{y^2}}{r_y^2} - \frac{T_{z^2}}{r_z^2} + \left(\frac{p_x^2}{r_x^2} + \frac{p_y^2}{r_y^2} - \frac{p_z^2}{r_z^2}\right)T_1. \tag{112}$$

These elliptic cones are created as axes-aligned, but they can be rotated, dilated, and translated using DCGA versor operations. All the DCGA surfaces can have general position, but we initially define them in axes-aligned position for simplicity. Defining the surfaces in general position may be possible if the value-extraction operations T_{xy}, T_{yz}, and T_{zx} are employed.

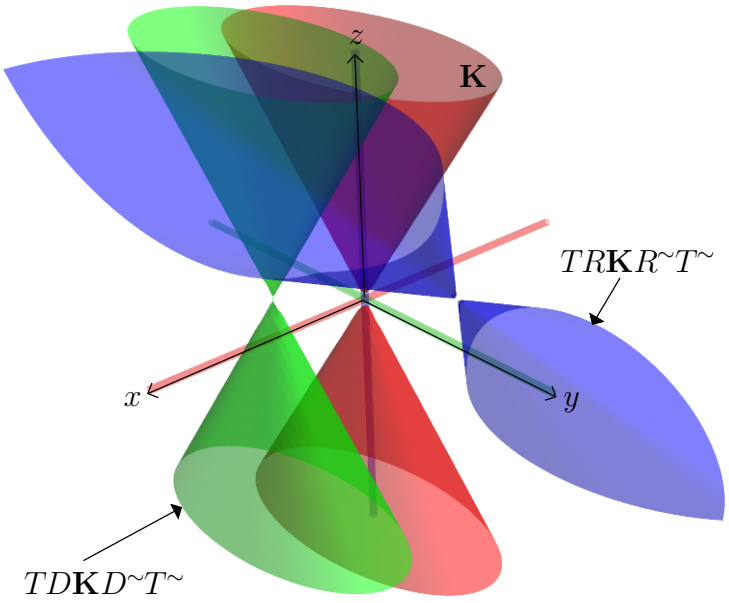

Figure 5. DCGA elliptic cones

Figure 5 shows some DCGA GIPNS cones positioned and transformed similar to the elliptic cylinders of Figure 4. The dilation of a cone does not change the cone shape, but it does dilate the cone center position to effectively translate a cone that is not initially at the origin to be further from the origin by the dilation factor.

4.9 DCGA GIPNS elliptic paraboloid

The elliptic paraboloid has a cone-like shape that opens up or down. The other paraboloid that would open the other way is imaginary with no real solution points.

The implicit quadric equation of a z-axis aligned elliptic paraboloid is

$$\frac{(x-p_x)^2}{r_x^2}+\frac{(y-p_y)^2}{r_y^2}-\frac{(z-p_z)}{r_z} = 0. \tag{113}$$

The surface opens up the z-axis for $r_z>0$, and opens down the z-axis for $r_z<0$. Similar equations for x-axis and y-axis aligned elliptic paraboloids are

$$\frac{(z-p_z)^2}{r_z^2}+\frac{(y-p_y)^2}{r_y^2}-\frac{(x-p_x)}{r_x} = 0 \tag{114}$$

$$\frac{(x-p_x)^2}{r_x^2}+\frac{(z-p_z)^2}{r_z^2}-\frac{(y-p_y)}{r_y} = 0. \tag{115}$$

Expanding the squares, the z-axis aligned equation is

$$\frac{-2\,p_x x}{r_x^2}+\frac{-2\,p_y y}{r_y^2}+\frac{-z}{r_z}+\frac{x^2}{r_x^2}+\frac{y^2}{r_y^2}+\left(\frac{p_x^2}{r_x^2}+\frac{p_y^2}{r_y^2}+\frac{p_z}{r_z}\right) = 0 \tag{116}$$

and the x-axis and y-axis aligned equations are

$$\frac{-2\,p_z z}{r_z^2}+\frac{-2\,p_y y}{r_y^2}+\frac{-x}{r_x}+\frac{z^2}{r_z^2}+\frac{y^2}{r_y^2}+\left(\frac{p_z^2}{r_z^2}+\frac{p_y^2}{r_y^2}+\frac{p_x}{r_x}\right) = 0 \tag{117}$$

$$\frac{-2\,p_x x}{r_x^2}+\frac{-2\,p_z z}{r_z^2}+\frac{-y}{r_y}+\frac{x^2}{r_x^2}+\frac{z^2}{r_z^2}+\left(\frac{p_x^2}{r_x^2}+\frac{p_z^2}{r_z^2}+\frac{p_y}{r_y}\right) = 0. \tag{118}$$

The DCGA GIPNS 2-vector *{x,y,z}-axis aligned elliptic paraboloid* surface entities $\mathbf{V}^{||\{x,y,z\}}$ are defined as

$$\mathbf{V}^{||x} = \frac{-2\,p_z T_z}{r_z^2} + \frac{-2\,p_y T_y}{r_y^2} + \frac{-T_x}{r_x} + \frac{T_{z^2}}{r_z^2} + \frac{T_{y^2}}{r_y^2} + \left(\frac{p_z^2}{r_z^2} + \frac{p_y^2}{r_y^2} + \frac{p_x}{r_x} \right) T_1 \tag{119}$$

$$\mathbf{V}^{||y} = \frac{-2\,p_x T_x}{r_x^2} + \frac{-2\,p_z T_z}{r_z^2} + \frac{-T_y}{r_y} + \frac{T_{x^2}}{r_x^2} + \frac{T_{z^2}}{r_z^2} + \left(\frac{p_x^2}{r_x^2} + \frac{p_z^2}{r_z^2} + \frac{p_y}{r_y} \right) T_1 \tag{120}$$

$$\mathbf{V}^{||z} \quad \frac{-2\,p_x T_x}{r_x^2} + \frac{-2\,p_y T_y}{r_y^2} + \frac{-T_z}{r_z} + \frac{T_{x^2}}{r_x^2} + \frac{T_{y^2}}{r_y^2} + \left(\frac{p_x^2}{r_x^2} + \frac{p_y^2}{r_y^2} + \frac{p_z}{r_z} \right) T_1. \tag{121}$$

A DCGA 2-vector point $\mathbf{T}_{\mathcal{D}} = \mathcal{D}(\mathbf{t})$ is tested against the DCGA 2-vector paraboloid $\mathbf{V}$ as

$$\mathbf{T}_{\mathcal{D}} \cdot \mathbf{V} \begin{cases} <0 & : \ \mathbf{t} \text{ is } \textit{inside} \text{ paraboloid} \\ =0 & : \ \mathbf{t} \text{ is } \textit{on} \text{ paraboloid} \\ >0 & : \ \mathbf{t} \text{ is } \textit{outside} \text{ paraboloid.} \end{cases} \tag{122}$$

This is similar to the ellipsoid incidence test, and this test is similar for many of the surfaces.

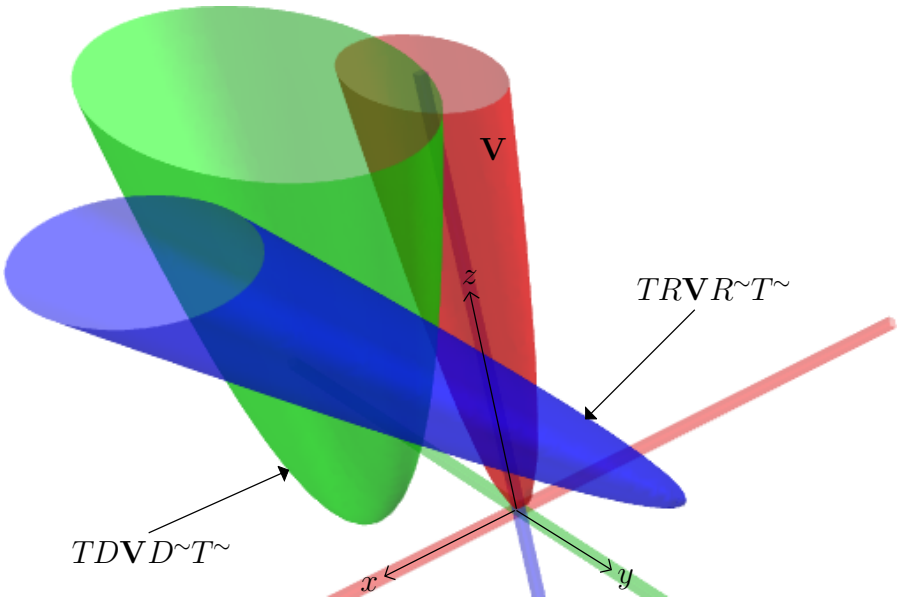

Figure 6. DCGA elliptic paraboloids

4.10 DCGA GIPNS hyperbolic paraboloid

The hyperbolic paraboloid has a saddle shape. The saddle can be mounted or aligned on a *saddle* axis with another axis chosen as the *up* axis. The third axis may be called the *straddle* axis.

The implicit quadric equation of a hyperbolic paraboloid is

$$\frac{(x-p_x)^2}{r_x^2} - \frac{(y-p_y)^2}{r_y^2} - \frac{(z-p_z)}{r_z} = 0. \tag{123}$$

This particular form of the equation has saddle x-axis, straddle y-axis, and up z-axis for $r_z > 0$ or up negative z-axis for $r_z < 0$. By its similarity to the z-axis aligned elliptic paraboloid with the elliptic y-axis inverted, this particular form can be seen as z-axis aligned. Other forms can be made by transposing axes, or by rotation around diagonal lines using DCGA rotor operations.

Expanding the squares, the equation is

$$\frac{-2p_x x}{r_x^2} + \frac{2p_y y}{r_y^2} + \frac{-z}{r_z} + \frac{x^2}{r_x^2} + \frac{-y^2}{r_y^2} + \left(\frac{p_x^2}{r_x^2} - \frac{p_y^2}{r_y^2} + \frac{p_z}{r_z} \right) = 0. \tag{124}$$

The DCGA GIPNS 2-vector *z-axis aligned hyperbolic paraboloid* surface entity $\mathbf{M}$ is defined as

$$\mathbf{M} = \frac{-2p_x T_x}{r_x^2} + \frac{2p_y T_y}{r_y^2} + \frac{-T_z}{r_z} + \frac{T_{x^2}}{r_x^2} + \frac{-T_{y^2}}{r_y^2} + \left(\frac{p_x^2}{r_x^2} - \frac{p_y^2}{r_y^2} + \frac{p_z}{r_z} \right) T_1. \tag{125}$$

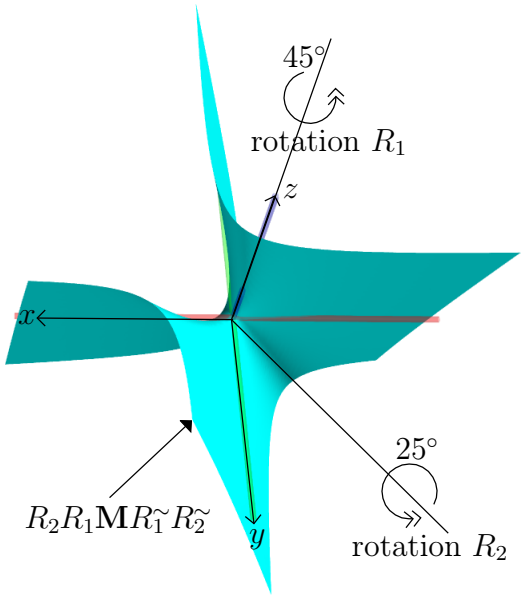

Figure 7. DCGA hyperbolic paraboloid rotated twice

Figure 7 shows the hyperbolic paraboloid entity $\mathbf{M}$, which is centered on the origin with parameters $r_x = r_y = r_z = 1$, and which was initially z-axis aligned. It was then rotated twice. The first rotation was 45° around the blue z-axis, pointing nearly out of the page. The second rotation was 25° around the line $\mathbf{n} = \frac{1}{\sqrt{2}}(-\mathbf{e}_1 + \mathbf{e}_2)$ pointing toward the lower-right of the page. The rotations follow the right-hand rule on a right-handed axes model.

4.11 DCGA GIPNS hyperboloid of one sheet

The hyperboloid of one sheet has a shape that is similar to an hourglass which continues to open both upward and downward. The implicit quadric equation is

$$\frac{(x - p_x)^2}{r_x^2} + \frac{(y - p_y)^2}{r_y^2} - \frac{(z - p_z)^2}{r_z^2} - 1 = 0. \tag{126}$$

This particular form opens up and down the z-axis. Planes parallel to the z-axis cut hyperbola sections. Planes perpendicular to the z-axis cut ellipse sections. At $z = p_z$, the ellipse section has a minimum size of the similar cylinder. Other forms can be made by transposing axes, or by rotation around diagonal lines using DCGA rotor operations.

Expanding the squares, the equation is

$$\frac{-2p_x x}{r_x^2} + \frac{-2p_y y}{r_y^2} + \frac{2p_z z}{r_z^2} + \frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} + \frac{-z^2}{r_z^2} + \left(\frac{p_x^2}{r_x^2} + \frac{p_y^2}{r_y^2} - \frac{p_z^2}{r_z^2} - 1 \right) = 0. \tag{127}$$

The DCGA GIPNS 2-vector *z-axis aligned hyperboloid of one sheet* surface entity $\mathbf{\Sigma}$ is defined as

$$\mathbf{\Sigma} = 2\left(\frac{p_z T_z}{r_z^2} - \frac{p_x T_x}{r_x^2} - \frac{p_y T_y}{r_y^2} \right) + \frac{T_{x^2}}{r_x^2} + \frac{T_{y^2}}{r_y^2} - \frac{T_{z^2}}{r_z^2} + \left(\frac{p_x^2}{r_x^2} + \frac{p_y^2}{r_y^2} - \frac{p_z^2}{r_z^2} - 1 \right) T_1. \tag{128}$$

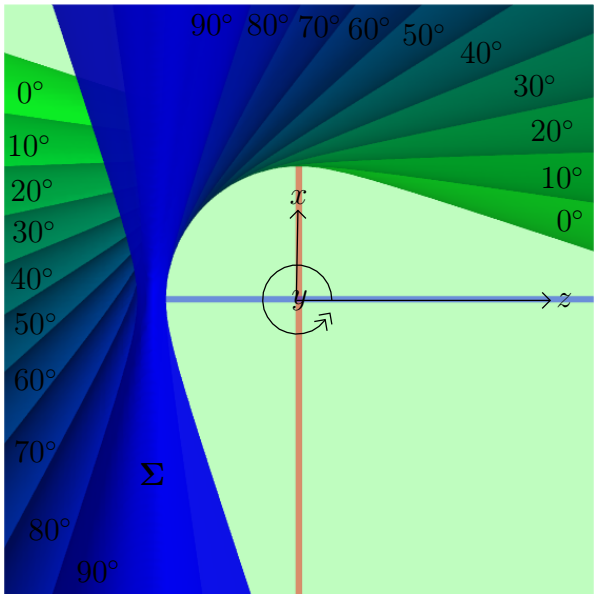

Figure 8. Rotation of DCGA hyperboloid of one sheet

Figure 8 is an orthographic (parallel projection) view from above the zx-plane that shows the hyperboloid $\mathbf{\Sigma}$ with $r_x = 1$, $r_y = 2$, $r_z = 3$, initially with green color, positioned at $p_x = 10$, and aligned up and down the z-axis. It is then rotated using a DCGA rotor R by 90° in 10° steps as its color fades to blue, with final position at $p_z = -10$ and aligned up and down the x-axis. The rotation is counter-clockwise around the y-axis coming out of the page on a right-handed system of axes. The x-axis is red and positive up, the y-axis is green (not visible), and the z-axis is blue and positive to the right. The axes are drawn by rendering thin elliptic cylinder entities. The right-hand rule, holding the y-axis, provides orientation for this rotation. The hyperboloid is rotated about the origin, around the y-axis, as a rigid body of points. In the symbolic computer algebra system (CAS) *Sympy* [17], the hyperboloid equation itself, as a DCGA entity, was rotated symbolically and graphed at each step using the *MayaVi* [13] data visualization software.

4.12 DCGA GIPNS hyperboloid of two sheets

The hyperboloid of two sheets has the shapes of two separate hyperbolic dishes; one opens upward, and the other one opens downward. The shape is like an hourglass that is pinched closed and the two halves are also separated by some distance. The implicit quadric equation is

$$-\frac{(x-p_x)^2}{r_x^2}-\frac{(y-p_y)^2}{r_y^2}+\frac{(z-p_z)^2}{r_z^2}-1 \;=\; 0. \tag{129}$$

This particular form has the two dishes opening up and down the z-axis. The dishes are separated by distance $2r_z$ centered at p_z. At $|z-p_z|=\sqrt{2}r_z$, the sections perpendicular to the z-axis are the size of the similar cylinder.

Expanding the squares, the equation is

$$\frac{2p_xx}{r_x^2}+\frac{2p_yy}{r_y^2}-\frac{2p_zz}{r_z^2}-\frac{x^2}{r_x^2}-\frac{y^2}{r_y^2}+\frac{z^2}{r_z^2}+\left(\frac{-p_x^2}{r_x^2}+\frac{-p_y^2}{r_y^2}+\frac{p_z^2}{r_z^2}-1\right) \;=\; 0. \tag{130}$$

The DCGA GIPNS 2-vector *z-axis aligned hyperboloid of two sheets* surface entity $\boldsymbol{\Xi}$ is defined as

$$\boldsymbol{\Xi} \;=\; 2\left(\frac{p_xT_x}{r_x^2}+\frac{p_yT_y}{r_y^2}-\frac{p_zT_z}{r_z^2}\right)-\frac{T_{x^2}}{r_x^2}-\frac{T_{y^2}}{r_y^2}+\frac{T_{z^2}}{r_z^2}+\left(\frac{p_z^2}{r_z^2}-\frac{p_x^2}{r_x^2}-\frac{p_y^2}{r_y^2}-1\right)T_1. \tag{131}$$

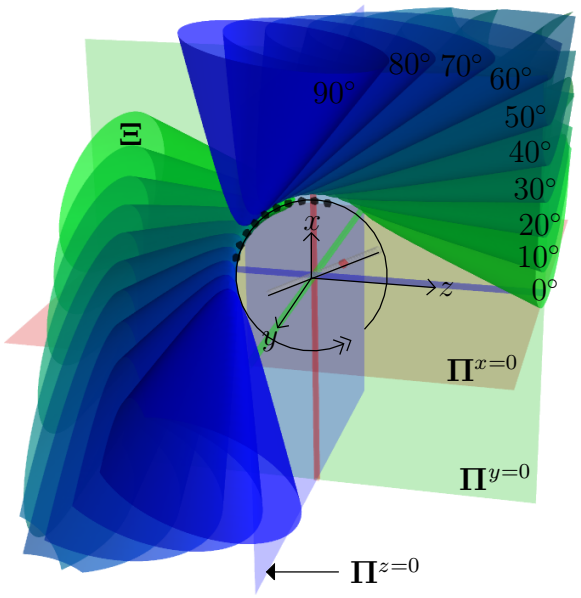

Figure 9. Rotation of DCGA hyperboloid of two sheets

Figure 9 shows a perspective view of the hyperboloid of two sheets $\boldsymbol{\Xi}$ initially with green color, centered at $p_x=5$, $p_y=-5$, and with semi-diameters $r_x=1$, $r_y=2$, $r_z=3$. The black dots (small sphere entities) are the center positions as the surface is rotated

around the white line through the origin and the red point $5\mathbf{e}_1+10\mathbf{e}_2+5\mathbf{e}_3$. The rotation is by 90° in 10° steps until it reaches the position of the blue surface. The first black dot is on the xy-plane (blue plane), and then the black dots go under the blue plane along an arc directly around the axis of rotation. The surface is carried along as a rigid body by the rotation using a DCGA rotor operation. The symbolic CAS *Sympy* was used for each rotation step, where an exact symbolic equation of the hyperboloid was generated by the rotated entity and graphed using *MayaVi* data visualization software.

4.13 DCGA GIPNS parabolic cylinder

The implicit quadric equation for the z-axis aligned parabolic cylinder is

$$\frac{(x-p_x)^2}{r_x^2}-\frac{(y-p_y)}{r_y} = 0. \tag{132}$$

The z coordinate is free, which creates a type of cylinder with parabolic sections that open up the y-axis for $r_y>0$, and open down the y-axis for $r_y<0$. The similar equations for x-axis and y-axis aligned parabolic cylinders are

$$\frac{(y-p_y)^2}{r_y^2}-\frac{(z-p_z)}{r_z} = 0 \tag{133}$$

$$\frac{(x-p_x)^2}{r_x^2}-\frac{(z-p_z)}{r_z} = 0 \tag{134}$$

with parabolas that open up or down the z-axis. Other forms can be made by transpositions or by using DCGA versor operations.

Expanding the squares, the equations are

$$\frac{-2p_xx}{r_x^2}-\frac{y}{r_y}+\frac{x^2}{r_x^2}+\left(\frac{p_x^2}{r_x^2}+\frac{p_y}{r_y}\right) = 0 \tag{135}$$

$$\frac{-2p_yy}{r_y^2}-\frac{z}{r_z}+\frac{y^2}{r_y^2}+\left(\frac{p_y^2}{r_y^2}+\frac{p_z}{r_z}\right) = 0 \tag{136}$$

$$\frac{-2p_xx}{r_x^2}-\frac{z}{r_z}+\frac{x^2}{r_x^2}+\left(\frac{p_x^2}{r_x^2}+\frac{p_z}{r_z}\right) = 0. \tag{137}$$

The DCGA GIPNS 2-vector *{x,y,z}-axis aligned parabolic cylinder* surface entities $\mathbf{B}^{||\{x,y,z\}}$ are defined as

$$\mathbf{B}^{||x} = \frac{-2p_yT_y}{r_y^2}-\frac{T_z}{r_z}+\frac{T_{y^2}}{r_y^2}+\left(\frac{p_y^2}{r_y^2}+\frac{p_z}{r_z}\right)T_1 \tag{138}$$

$$\mathbf{B}^{||y} = \frac{-2p_xT_x}{r_x^2}-\frac{T_z}{r_z}+\frac{T_{x^2}}{r_x^2}+\left(\frac{p_x^2}{r_x^2}+\frac{p_z}{r_z}\right)T_1 \tag{139}$$

$$\mathbf{B}^{||z} = \frac{-2p_xT_x}{r_x^2}-\frac{T_y}{r_y}+\frac{T_{x^2}}{r_x^2}+\left(\frac{p_x^2}{r_x^2}+\frac{p_y}{r_y}\right)T_1. \tag{140}$$

These are created as axes-aligned surfaces, but can be rotated, dilated, and translated using DCGA versor operations.

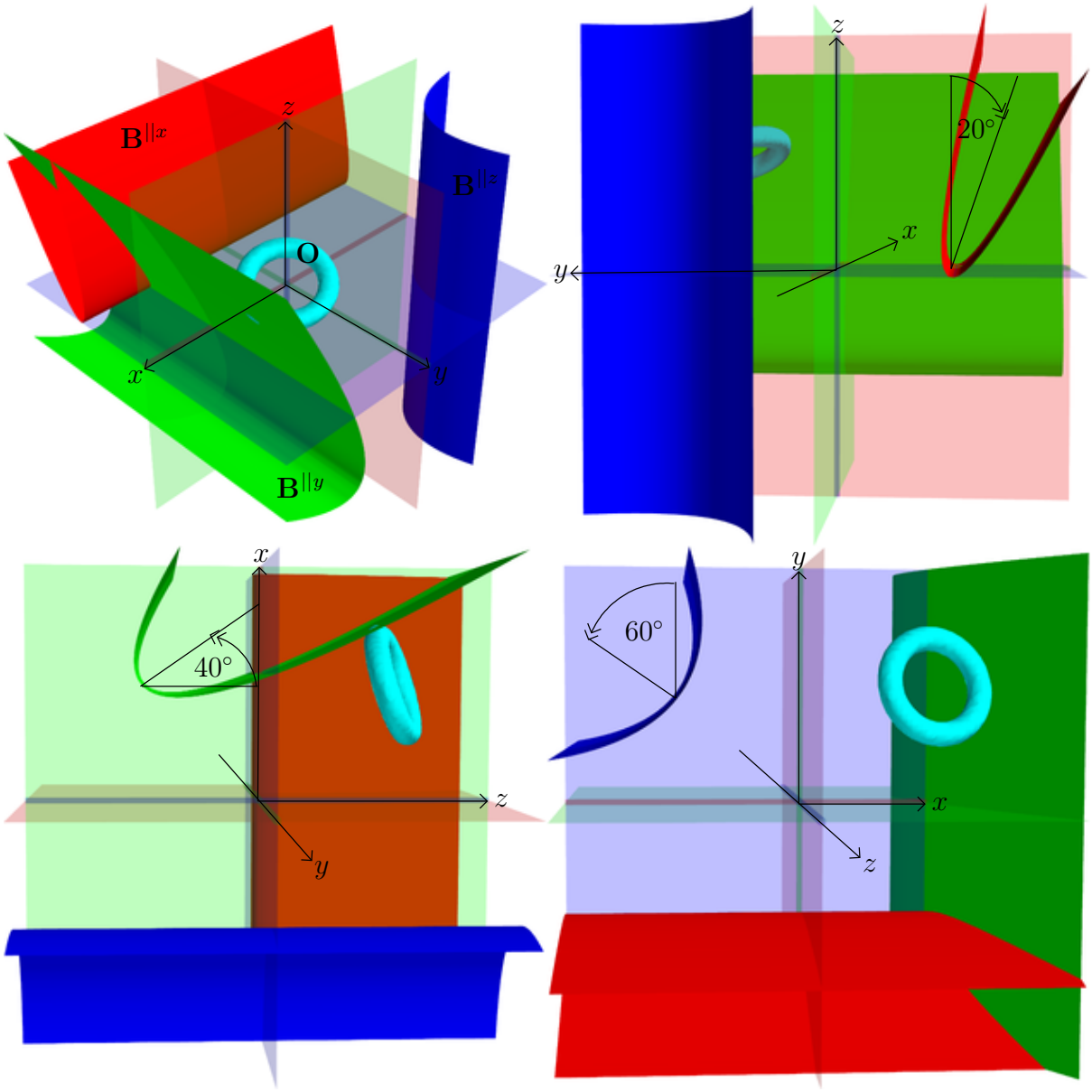

Figure 10. DCGA parabolic cylinders and toroid rotated and translated

Figure 10 shows multiple perspective views of the DCGA GIPNS 2-vector parabolic cylinders and toroid surface entities rendered together in one scene. The red cylinder is x-axis aligned, $r_y = 1$, $r_z = 1$, rotated 20° around the x-axis, and then translated by $\mathbf{d} = -10\mathbf{e}_2$ from the origin. The green cylinder is y-axis aligned, $r_x = 2$, $r_z = 1$, rotated 40° around the y-axis, and then translated by $\mathbf{d} = 10\mathbf{e}_1 - 10\mathbf{e}_3$ from the origin. The blue cylinder is z-axis aligned, $r_x = 4$, $r_y = 1$, rotated 60° around the z-axis, and then translated by $\mathbf{d} = -10\mathbf{e}_1 + 10\mathbf{e}_2$ from the origin. The toroid, with $R = 4$ and $r = 1$, is rotated 25° around the axis $\mathbf{n} = \frac{1}{\sqrt{2}}(-\mathbf{e}_1 + \mathbf{e}_2)$, and then translated by $\mathbf{d} = 10\mathbf{e}_1 + 10\mathbf{e}_2 + 10\mathbf{e}_3$ from the origin. The rotations follow the right-hand rule on right-handed axes. The rotation-translations were performed as compositions of DCGA rotors and translators. Symbolic

CAS *Sympy* was used to generate exact equations of the transformed entities, which were then graphed using the *MayaVi* data visualization software.

4.14 DCGA GIPNS hyperbolic cylinder

The implicit quadric equation for the z-axis aligned hyperbolic cylinder is

$$\frac{(x-p_x)^2}{r_x^2} - \frac{(y-p_y)^2}{r_y^2} - 1 \;=\; 0. \tag{141}$$

The z coordinate is free, which creates a type of cylinder with hyperbolic sections that open up and down the x-axis. The hyperbola branches are separated by distance $2r_x$ centered at $\mathbf{p} = p_x\mathbf{e}_1 + p_y\mathbf{e}_2 + z\mathbf{e}_3$. The asymptotes are the lines

$$(y-p_y) \;=\; \pm\frac{r_y}{r_x}(x-p_x) \tag{142}$$

through (p_x, p_y), where in the limit as $x \to \pm\infty$ the -1 becomes insignificant.

The similar equations for x-axis and y-axis aligned hyperbolic cylinders are

$$\frac{(y-p_y)^2}{r_y^2} - \frac{(z-p_z)^2}{r_z^2} - 1 \;=\; 0 \tag{143}$$

$$\frac{(z-p_z)^2}{r_z^2} - \frac{(x-p_x)^2}{r_x^2} - 1 \;=\; 0 \tag{144}$$

with hyperbolas that open up and down the y-axis or z-axis. Other forms can be made by transpositions or by using DCGA versor operations.

Expanding the squares, the equations for x, y, z-aligned hyperbolic cylinders are

$$\frac{-2p_y y}{r_y^2} + \frac{2p_z z}{r_z^2} + \frac{y^2}{r_y^2} - \frac{z^2}{r_z^2} + \left(\frac{p_y^2}{r_y^2} - \frac{p_z^2}{r_z^2} - 1\right) \;=\; 0 \tag{145}$$

$$\frac{-2p_z z}{r_z^2} + \frac{2p_x x}{r_x^2} + \frac{z^2}{r_z^2} - \frac{x^2}{r_x^2} + \left(\frac{p_z^2}{r_z^2} - \frac{p_x^2}{r_x^2} - 1\right) \;=\; 0 \tag{146}$$

$$\frac{-2p_x x}{r_x^2} + \frac{2p_y y}{r_y^2} + \frac{x^2}{r_x^2} - \frac{y^2}{r_y^2} + \left(\frac{p_x^2}{r_x^2} - \frac{p_y^2}{r_y^2} - 1\right) \;=\; 0. \tag{147}$$

The DCGA GIPNS 2-vector *{x,y,z}-axis aligned hyperbolic cylinder* surface entities $\mathbf{J}^{||\{x,y,z\}}$ are defined as

$$\mathbf{J}^{||x} \;=\; \frac{-2p_y T_y}{r_y^2} + \frac{2p_z T_z}{r_z^2} + \frac{T_{y^2}}{r_y^2} - \frac{T_{z^2}}{r_z^2} + \left(\frac{p_y^2}{r_y^2} - \frac{p_z^2}{r_z^2} - 1\right)T_1 \tag{148}$$

$$\mathbf{J}^{||y} \;=\; \frac{-2p_z T_z}{r_z^2} + \frac{2p_x T_x}{r_x^2} + \frac{T_{z^2}}{r_z^2} - \frac{T_{x^2}}{r_x^2} + \left(\frac{p_z^2}{r_z^2} - \frac{p_x^2}{r_x^2} - 1\right)T_1 \tag{149}$$

$$\mathbf{J}^{||z} \;=\; \frac{-2p_x T_x}{r_x^2} + \frac{2p_y T_y}{r_y^2} + \frac{T_{x^2}}{r_x^2} - \frac{T_{y^2}}{r_y^2} + \left(\frac{p_x^2}{r_x^2} - \frac{p_y^2}{r_y^2} - 1\right)T_1. \tag{150}$$

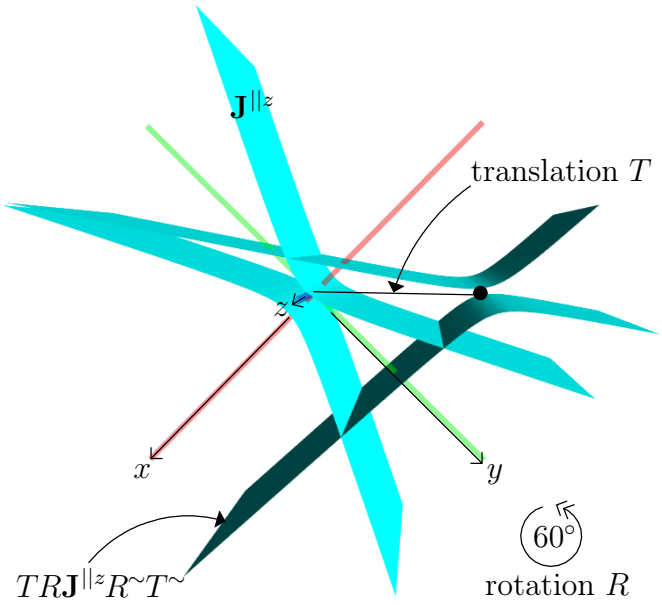

Figure 11. DCGA hyperbolic cylinder rotated and translated

Figure 11 shows the z-axis aligned hyperbolic cylinder, with initial parameters $p_x = 0$, $p_y = 0$, $r_x = 1$, and $r_y = 2$. The second rendering of it is rotated 60° around the z-axis and then translated by $\mathbf{d} = -10\mathbf{e}_1 + 10\mathbf{e}_2$ using a composition of DCGA rotor R and translator T operations.

4.15 DCGA GIPNS parallel planes pair

Parallel pairs of axes-aligned planes are represented by the simple quadratic equations in one variable

$$(x - p_{x1})(x - p_{x2}) = 0 \tag{151}$$
$$(y - p_{y1})(y - p_{y2}) = 0 \tag{152}$$
$$(z - p_{z1})(z - p_{z2}) = 0. \tag{153}$$

Each solution is a plane. Expanding the equations gives

$$x^2 - (p_{x1} + p_{x2})x + p_{x1}p_{x2} = 0 \tag{154}$$
$$y^2 - (p_{y1} + p_{y2})y + p_{y1}p_{y2} = 0 \tag{155}$$
$$z^2 - (p_{z1} + p_{z2})z + p_{z1}p_{z2} = 0. \tag{156}$$

The DCGA GIPNS 2-vector *parallel {x,y,z}-planes pair* entities $\mathbf{\Pi}^{\perp\{x,y,z\}}$ are defined as

$$\mathbf{\Pi}^{\perp x} = T_{x^2} - (p_{x1} + p_{x2})T_x + p_{x1}p_{x2}T_1 \tag{157}$$
$$\mathbf{\Pi}^{\perp y} = T_{y^2} - (p_{y1} + p_{y2})T_y + p_{y1}p_{y2}T_1 \tag{158}$$
$$\mathbf{\Pi}^{\perp z} = T_{z^2} - (p_{z1} + p_{z2})T_z + p_{z1}p_{z2}T_1. \tag{159}$$

These surfaces can also be described as being types of cylinders with cross sections being two parallel lines.

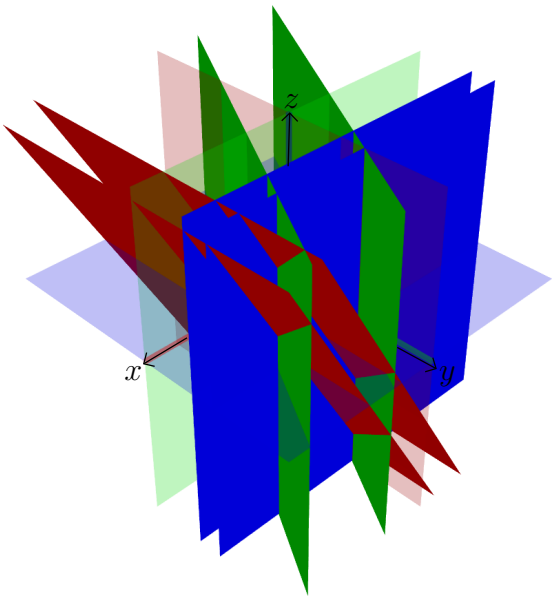

Figure 12. DCGA parallel planes pairs rotated

Figure 12 shows the DCGA GIPNS parallel planes pair entities rotated using DCGA rotor operations. The red planes pair is initially perpendicular to the x-axis through points $p_{x1} = 4$ and $p_{x2} = 8$, then it is rotated 30° around the y-axis. The green planes pair is initially perpendicular to the y-axis through points $p_{y1} = -5$ and $p_{y2} = 5$, then it is rotated 60° around the z-axis. The blue planes pair is initially perpendicular to the z-axis through points $p_{z1} = -10$ and $p_{z2} = -7$, then it is rotated 90° around the x-axis until it is perpendicular to the y-axis through the points $p_{y1} = 10$ and $p_{y2} = 7$.

4.16 DCGA GIPNS non-parallel planes pair

The implicit quadric equation for a pair of intersecting, non-parallel planes that are parallel to the z-axis is

$$\frac{(x-p_x)^2}{r_x^2} - \frac{(y-p_y)^2}{r_y^2} = 0. \tag{160}$$

This equation can be written as

$$(y-p_y) = \pm\frac{r_y}{r_x}(x-p_x) \tag{161}$$

with the z coordinate free to range. This surface can also be described as a kind of cylinder with a cross section in plane z that is two lines with slopes $\pm\frac{r_y}{r_x}$ intersecting at $\mathbf{p} = p_x\mathbf{e}_1 + p_y\mathbf{e}_2 + z\mathbf{e}_3$.

Expanding the squares, the equation is

$$\frac{-2p_x x}{r_x^2} + \frac{2p_y y}{r_y^2} + \frac{x^2}{r_x^2} - \frac{y^2}{r_y^2} + \left(\frac{p_x^2}{r_x^2} - \frac{p_y^2}{r_y^2}\right) = 0. \tag{162}$$

The DCGA GIPNS 2-vector *{x,y,z}-axis aligned non-parallel planes pair* entities $\mathbf{X}^{||\{x,y,z\}}$ are defined as

$$\mathbf{X}^{||x} = \frac{-2p_y T_y}{r_y^2} + \frac{2p_z T_z}{r_z^2} + \frac{T_{y^2}}{r_y^2} - \frac{T_{z^2}}{r_z^2} + \left(\frac{p_y^2}{r_y^2} - \frac{p_z^2}{r_z^2}\right)T_1 \tag{163}$$

$$\mathbf{X}^{||y} = \frac{-2p_z T_z}{r_z^2} + \frac{2p_x T_x}{r_x^2} + \frac{T_{z^2}}{r_z^2} - \frac{T_{x^2}}{r_x^2} + \left(\frac{p_z^2}{r_z^2} - \frac{p_x^2}{r_x^2}\right)T_1 \tag{164}$$

$$\mathbf{X}^{||z} = \frac{-2p_xT_x}{r_x^2} + \frac{2p_yT_y}{r_y^2} + \frac{T_{x^2}}{r_x^2} - \frac{T_{y^2}}{r_y^2} + \left(\frac{p_x^2}{r_x^2} - \frac{p_y^2}{r_y^2}\right)T_1. \tag{165}$$

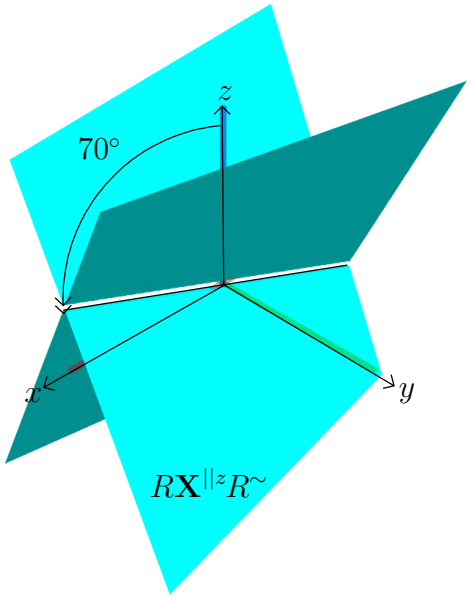

Figure 13. DCGA non-parallel planes pair rotated

Figure 13 shows the entity $\mathbf{X}^{||z}$, initially having planes with slopes $\pm\frac{r_y}{r_x} = \pm\frac{1}{2}$ that cross at the origin point $p_x = 0$, $p_y = 0$ in the xy-plane. It is then rotated using a DCGA rotor R around the y-axis by 70°. The line of crossing points was initially the z-axis, but after rotation the crossing line is at 70° off the z-axis, around the y-axis. Like the other DCGA entities, the non-parallel planes pair entities can be transformed into general positions using DCGA versor operations.

4.17 DCGA GIPNS ellipse

The ellipse is a conic section, and like all conic sections it can be made as the intersection of a plane and cone, but we are not limited to intersecting with cones. A simple ellipse representation is made as the intersection of a plane and elliptic cylinder. The parabola and hyperbola are also conic sections, and their simple representations are as planes intersecting parabolic and hyperbolic cylinders. We can just define these conic sections as these plane and cylinder intersections, but these conic sections could be formed by a wide variety of other possible intersections.

The DCGA GIPNS 4-vector *xy-plane ellipse* 1D surface entity $\boldsymbol{\epsilon}^{||xy}$ is defined as

$$\boldsymbol{\epsilon}^{||xy} = \boldsymbol{\Pi}^{z=0} \wedge \mathbf{H}^{||z} \tag{166}$$

where the DCGA GIPNS 2-vector *plane* $\boldsymbol{\Pi}^{z=0}$ is the entity for the plane $z = 0$, and the DCGA GIPNS 2-vector *elliptic cylinder* $\mathbf{H}^{||z}$ is as previously defined and directly represents an ellipse in the xy-plane. Other similar ellipse entities are the wedges of other planes with other elliptic cylinders that are aligned differently.

A DCGA GIPNS ellipse entity $\boldsymbol{\epsilon}$, or its dual DCGA GOPNS ellipse entity $\boldsymbol{\epsilon}^{*\mathcal{D}} = \boldsymbol{\epsilon}\,/\,\mathbf{I}_{\mathcal{D}}$, can be rotated, dilated, and translated using DCGA versor operations, where *versor outermorphism* is applied to the wedge of plane and cylinder that form the ellipse entity. In versor operations on the ellipse entity, the plane and cylinder are each transformed by the versor operations, and then the transformed plane and cylinder are intersected.

The invariant test $\mathbf{e}_\infty \cdot \boldsymbol{\epsilon}^{||xy} = 0$ seems to indicate that the ellipse reaches to infinity, but this should be considered as an *invalid* test.

4.18 DCGA GIPNS parabola

The DCGA GIPNS 4-vector *xy-plane parabola* 1D surface entity $\boldsymbol{\rho}^{||xy}$ is defined as

$$\boldsymbol{\rho}^{||xy} = \mathbf{\Pi}^{z=0} \wedge \mathbf{B}^{||z} \tag{167}$$

where the DCGA GIPNS 2-vector *plane* $\mathbf{\Pi}^{z=0}$ is the entity for the plane $z = 0$, and the DCGA GIPNS 2-vector *parabolic cylinder* $\mathbf{B}^{||z}$ is as previously defined and directly represents a parabola in the xy-plane. Other similar parabola entities are the wedges of other planes with other parabolic cylinders that are aligned differently.

4.19 DCGA GIPNS hyperbola

The DCGA GIPNS 4-vector *xy-plane hyperbola* 1D surface entity $\boldsymbol{\eta}^{||xy}$ is defined as

$$\boldsymbol{\eta}^{||xy} = \mathbf{\Pi}^{z=0} \wedge \mathbf{J}^{||z} \tag{168}$$

where the DCGA GIPNS 2-vector *plane* $\mathbf{\Pi}^{z=0}$ is the entity for the plane $z = 0$, and the DCGA GIPNS 2-vector *hyperbolic cylinder* $\mathbf{J}^{||z}$ is as previously defined and directly represents a hyperbola in the xy-plane. Other similar hyperbola entities are the wedges of other planes with other hyperbolic cylinders that are aligned differently.

4.20 DCGA GIPNS cyclide

The implicit quartic equation for a *Darboux cyclide* [11] surface is

$$\begin{aligned} A\mathbf{t}^4 + B\mathbf{t}^2 + \\ Cx\mathbf{t}^2 + Dy\mathbf{t}^2 + Ez\mathbf{t}^2 + \\ Fx^2 + Gy^2 + Hz^2 + \\ Ixy + Jyz + Kzx + \\ Lx + My + Nz + O &= 0 \end{aligned} \tag{169}$$

where $\mathbf{t} = x\mathbf{e}_1 + y\mathbf{e}_2 + z\mathbf{e}_3$ is a test point and the $A...O$ are 15 real scalar constants. The point $\mathbf{t}$ is on the cyclide surface if the equation holds good.

The DCGA GIPNS 2-vector *Darboux cyclide* surface entity $\mathbf{\Omega}$ is defined as

$$\begin{aligned} \mathbf{\Omega} = & AT_{\mathbf{t}^4} + BT_{\mathbf{t}^2} + \\ & CT_{x\mathbf{t}^2} + DT_{y\mathbf{t}^2} + ET_{z\mathbf{t}^2} + \\ & FT_{x^2} + GT_{y^2} + HT_{z^2} + \\ & IT_{xy} + JT_{yz} + KT_{zx} + \\ & LT_x + MT_y + NT_z + OT_1. \end{aligned} \tag{170}$$

All DCGA versor operations are valid on the Darboux cyclide entity $\mathbf{\Omega}$ and its dual $\mathbf{\Omega}^{*\mathcal{D}}$. The Darboux cyclide entity $\mathbf{\Omega}$ can be intersected with DCGA GIPNS planes, spheres, lines, and circles.

Entities with $A \neq 0$ have valid dilator operations with all dilation factors, including dilation factor 0. If $A \neq 0$, then $\mathbf{\Omega}$ dilates by factor 0 into $AT_{\mathbf{t}^4} = -4A\mathbf{e}_o$, which is a *valid* result representing the point at the origin. If $A = 0$, then $\mathbf{\Omega}$ dilates by factor 0 into scalar 0, which is an *invalid* result. Said differently, GIPNS entities that have $\mathbf{e}_o$ as a term dilate by factor 0 into $\mathbf{e}_o$ (up to scale), and other GIPNS entities dilate by factor 0 into scalar 0. The duals of such dilations are either $\mathbf{e}_o^{*\mathcal{D}}$ or 0.

It was first discussed in Section 3, on the DCGA point $\mathbf{T}_\mathcal{D}$ and extraction operators T_s, and then mentioned again in the section on the DCGA GIPNS 2-vector ellipsoid surface entity $\mathbf{E}$, that any DCGA GIPNS 2-vector surface entity without a term in $T_{\mathbf{t}^4}$ has the surface point $\mathbf{e}_\infty$. This includes some closed surfaces that would not be expected to have the point $\mathbf{e}_\infty$.

The constant B and the constants F, G, H allow alternative formulations of an entity $\mathbf{\Omega}$. If $F = G = H$, then F could be added to B to form a simpler entity having fewer terms by eliminating F, G, H. If an amount b is subtracted from each of F, G, H, then it can be added back as $(B + b)$, or the reverse. The surface represented by the entity $\mathbf{\Omega}$ is not affected by the specific choice of how to use B, F, G, H, but other metrical properties could be affected. Metrical properties include the scalar results returned by the inner products of entities, which are often distance measures between surfaces.

The Darboux cyclide entity $\mathbf{\Omega}$ is the most general form of DCGA GIPNS 2-vector surface entity that can be defined using the DCGA point $\mathbf{T}_\mathcal{D}$ value-extraction operators $s = T_s \cdot \mathbf{T}_\mathcal{D}$. DCGA could be described as a conformal geometric algebra on Darboux cyclide surface entities in 3D space. The tentative name, DCGA, could also be $\mathcal{G}_{8,2}$ *Darboux Cyclide Geometric Algebra*. All of the DCGA GIPNS 2-vector quadric surface entities and the toroid entity, and also their inversive or cyclidic surface forms when reflected in DCGA spheres, can be represented as instances of the Darboux cyclide entity $\mathbf{\Omega}$.

An instance of the DCGA GIPNS 2-vector Darboux cyclide surface entity $\mathbf{\Omega}$ can be produced by one or more *inversions in* DCGA GIPNS 2-vector *spheres* $\mathbf{S}_i$ of any DCGA GIPNS 2-vector surface entity $\mathbf{\Upsilon}$. For example, the inversion of a DCGA GIPNS 2-vector quadric or toroid surface entity $\mathbf{\Upsilon}$ in a DCGA GIPNS 2-vector sphere entity $\mathbf{S}$ is the reflection $\mathbf{\Omega} = \mathbf{S}\mathbf{\Upsilon}\mathbf{S}^\sim$, which is an instance of the Darboux cyclide surface entity $\mathbf{\Omega}$ that appears to be $\mathbf{\Upsilon}$ reflected in the sphere $\mathbf{S}$. The sphere $\mathbf{S}$ can be visualized as a spherical mirrored surface when $\mathbf{\Upsilon}$ is located entirely outside $\mathbf{S}$, and the cyclidic reflection of $\mathbf{\Upsilon}$ is seen on the surface of $\mathbf{S}$ or inside of $\mathbf{S}$. Successive inversions or reflections of $\mathbf{\Upsilon}$ in multiple spheres $\mathbf{S}_i$ transforms $\mathbf{\Upsilon}$ into a succession of different cyclide surface entities, all based on the initial shape of $\mathbf{\Upsilon}$. The distinction between inversion and reflection, which concerns whether or not the orientation of the surface remains the same or becomes inside-out, is not being made here.

Dual DCGA GOPNS 8-vector surface entities $\mathbf{\Upsilon}^{*\mathcal{D}}$ can also be reflected in a sphere $\mathbf{S}$, or in its dual $\mathbf{S}^{*\mathcal{D}}$, to produce an instance of the dual DCGA GOPNS 8-vector Darboux cyclide surface entity $\mathbf{\Omega}^{*\mathcal{D}}$.

A singular outlier surface point $\mathbf{P}_\mathcal{D}$ will exist on the inverse surface entity $\mathbf{S}\mathbf{\Upsilon}\mathbf{S}^\sim$ of any DCGA GIPNS 2-vector *closed surface* entity $\mathbf{\Upsilon}$ without a term in $T_{\mathbf{t}^4} = -4\mathbf{e}_o$. The singular outlier surface point $\mathbf{P}_\mathcal{D}$ is always the center point of the inversion sphere $\mathbf{S}$. The inverse surface entity $\mathbf{S}\mathbf{\Upsilon}\mathbf{S}^\sim$ of an *open surface* entity $\mathbf{\Upsilon}$ that is known to reach $\mathbf{e}_\infty$ is expected to have the point $\mathbf{P}_\mathcal{D}$, as it does. If $\mathbf{\Upsilon}$ is a *closed surface* entity, then it does not actually reach to infinity, and yet *any such entity* $\mathbf{\Upsilon}$ without a term in $T_{\mathbf{t}^4}$ has the surface point $\mathbf{e}_\infty$ and has an inverse surface $\mathbf{S}\mathbf{\Upsilon}\mathbf{S}^\sim$ that has the inversion sphere

center point $\mathbf{P}_{\mathcal{D}}$ as a singular outlier surface point. The inverse of point $\mathbf{e}_\infty$ is always the inversion sphere center point $\mathbf{P}_{\mathcal{D}}$, or the reverse. A singular outlier surface point may be invisible on a surface plot. In particular, for any DCGA GIPNS 2-vector ellipsoid surface entity $\mathbf{E}$, its inverse surface entity $\mathbf{SES}^\sim$ has the inversion sphere $\mathbf{S}$ center point $\mathbf{P}_{\mathcal{D}}$ as an (invisible) singular outlier surface point. An unexpected $\mathbf{e}_\infty$ or outlier point $\mathbf{P}_{\mathcal{D}}$ is a **possible problem** for an application, but awareness of their existence may allow for a workaround to mitigate any possible problem caused by their existence.

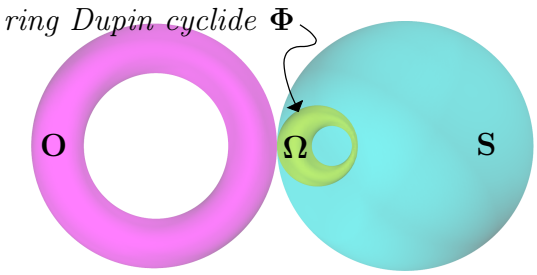

Figure 14. Toroid $\mathbf{O}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{SOS}^\sim$

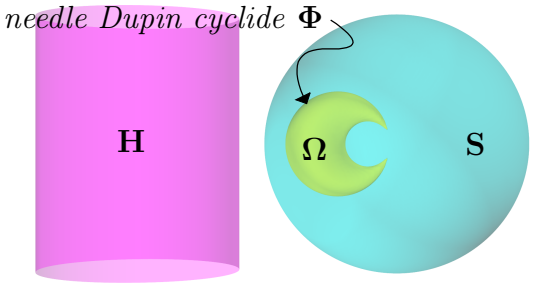

Figure 15. Cylinder $\mathbf{H}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{SHS}^\sim$

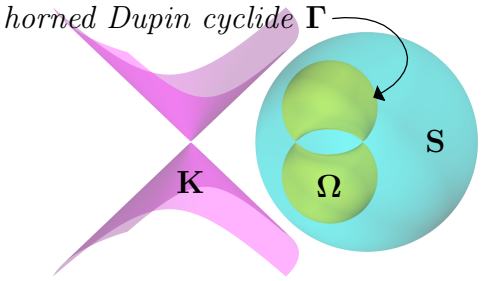

Figure 16. Cone $\mathbf{K}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{SKS}^\sim$

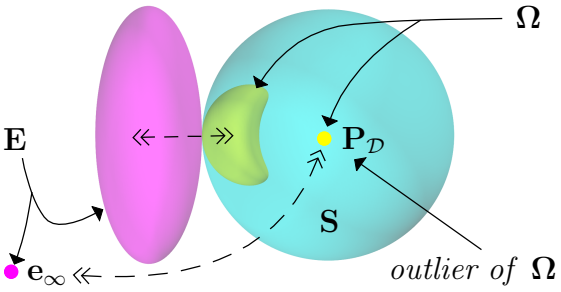

Figure 17. Ellipsoid $\mathbf{E}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{SES}^\sim$

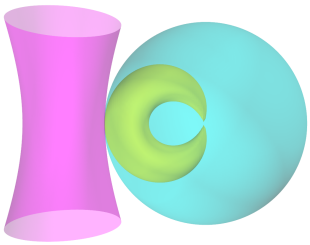

Figure 18. Hyperboloid of one sheet $\mathbf{\Sigma}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{S\Sigma S}^{\sim}$

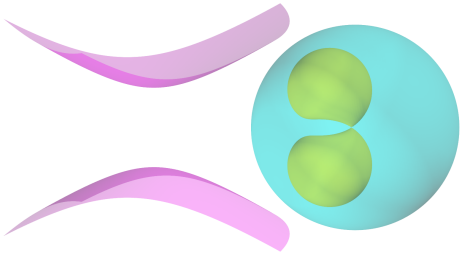

Figure 19. Hyperboloid of two sheets $\mathbf{\Xi}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{S\Xi S}^{\sim}$

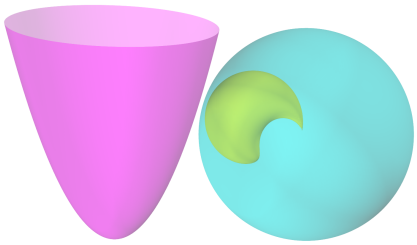

Figure 20. Paraboloid $\mathbf{V}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{SVS}^{\sim}$

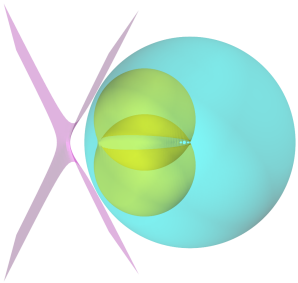

Figure 21. Hyperbolic paraboloid $\mathbf{M}$ reflected in sphere $\mathbf{S}$, $\mathbf{\Omega} = \mathbf{SMS}^{\sim}$

It is beyond the scope of this paper to analyze and define every possible type of cyclide that can be defined as instances of the Darboux cyclide entity $\mathbf{\Omega}$. However, as an example of what can be defined, we can consider the subsets of cyclides known as *Dupin cyclides* and *parabolic cyclides* [5][15]. The Dupin cyclides can generalize the circular toroid (torus) by creating cyclides based on torus inversion in a sphere. As shown in Figures 14,15,16,17,18,19,20,21, it is possible to define many other specific cyclide entities based on each of the quadric surfaces reflected in spheres.

4.20.1 DCGA GIPNS Dupin cyclide

The implicit quartic equation for a Dupin cyclide surface is

$$(\mathbf{t}^2+(b^2-\mu^2))^2-4(a\,x-c\mu)^2-4b^2y^2 \;=\; 0 \tag{171}$$

where $\mathbf{t}=x\mathbf{e}_1+y\mathbf{e}_2+z\mathbf{e}_3$ is a test point. Expanding this equation gives

$$\mathbf{t}^4+2\mathbf{t}^2(b^2-\mu^2)+(b^2-\mu^2)^2-4(a^2x^2-2a\,c\mu x+c^2\mu^2)-4b^2y^2 \;=\; 0. \tag{172}$$

The DCGA GIPNS 2-vector *Dupin cyclide* surface entity $\mathbf{\Phi}$ is defined as

$$\begin{aligned}\mathbf{\Phi} \;=\;& T_{\mathbf{t}^4}+2T_{\mathbf{t}^2}(b^2-\mu^2)+ \\ & -4a^2T_{x^2}-4b^2T_{y^2}+ \\ & 8a\,c\mu T_x+((b^2-\mu^2)^2-4c^2\mu^2)T_1.\end{aligned} \tag{173}$$

The scalar parameters of the surface are a, b, c, μ, with b always squared. The Dupin cyclide can be described as a surface that envelops a family of spheres defined by two initial spheres of *minor radii*, r_1 and r_2, centered on a circle of *major radius* R. To gain a more intuitive expression of the Dupin cyclide equation, we can define the parameters as

$$a \;=\; R \tag{174}$$
$$\mu \;=\; \frac{1}{2}(r_1+r_2) \tag{175}$$
$$c \;=\; \frac{1}{2}(r_1-r_2) \tag{176}$$
$$b^2 \;=\; a^2-c^2. \tag{177}$$

The Dupin cyclide $\mathbf{\Phi}$ is now defined by the three radii parameters, R, r_1, r_2. When $r=r_1=r_2$, the Dupin cyclide $\mathbf{\Phi}$ is exactly the same entity as the toroid $\mathbf{O}$ with parameters R and r.

The DCGA GIPNS Dupin cyclide $\mathbf{\Phi}$ has the following *related points*:

$$\begin{aligned}\text{Center of initial sphere } \mathbf{S}_1 \text{ with radius } r_1 \;&:\; -R\mathbf{e}_1 \\ \text{Center of initial sphere } \mathbf{S}_2 \text{ with radius } r_2 \;&:\; +R\mathbf{e}_1 \\ \text{Center of ring or spindle hole in the cyclide} \;&:\; +c\mathbf{e}_1 \\ \text{Center of sphere enclosing entire cyclide} \;&:\; -c\mathbf{e}_1 \\ \text{Radius around } -c\mathbf{e}_1 \text{ enclosing entire cyclide} \;&:\; \mu+R.\end{aligned}$$

Twelve *surface points* on the Dupin cyclide $\mathbf{\Phi}$ are:

$$\begin{aligned}\text{2 surface points on } \mathbf{S}_1 \text{ with radius } r_1 \;&:\; -R\mathbf{e}_1\pm r_1\mathbf{e}_1 \\ \text{2 surface points on } \mathbf{S}_1 \text{ with radius } r_1 \;&:\; -R\mathbf{e}_1\pm r_1\mathbf{e}_3 \\ \text{2 surface points on } \mathbf{S}_2 \text{ with radius } r_2 \;&:\; +R\mathbf{e}_1\pm r_2\mathbf{e}_1 \\ \text{2 surface points on } \mathbf{S}_2 \text{ with radius } r_2 \;&:\; +R\mathbf{e}_1\pm r_2\mathbf{e}_3 \\ \text{2 surface points} \;&:\; -c\mathbf{e}_1\pm(\mu+R)\mathbf{e}_2 \\ \text{2 surface points} \;&:\; +c\mathbf{e}_1\pm(\mu-R)\mathbf{e}_2.\end{aligned}$$

The Dupin cyclide $\mathbf{\Phi}$ is initially created having these 12 points. All DCGA versor operations are valid on the Dupin cyclide $\mathbf{\Phi}$ and can be used to rotate, dilate, and translate it into another general position.

The type of cyclide or torus represented by $\mathbf{\Phi}$ is determined by:

$$\begin{aligned}
\textit{Ring cyclide} \text{ when } &: (r_1+r_2)<2R \\
\textit{Spindle cyclide} \text{ when } &: (r_1+r_2)>2R \\
\textit{Horn cyclide} \text{ when } &: (r_1+r_2)=2R \\
\textit{Ring torus} \text{ when } &: (r_1=r_2)<R \\
\textit{Spindle torus} \text{ when } &: (r_1=r_2)>R \\
\textit{Horn torus} \text{ when } &: (r_1=r_2)=R.
\end{aligned}$$

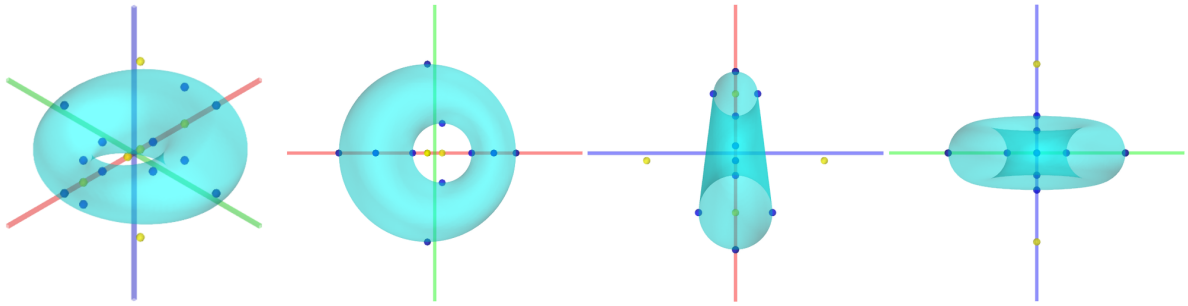

Figure 22. Ring cyclide $\mathbf{\Phi}$, $(r_1+r_2)<2R$

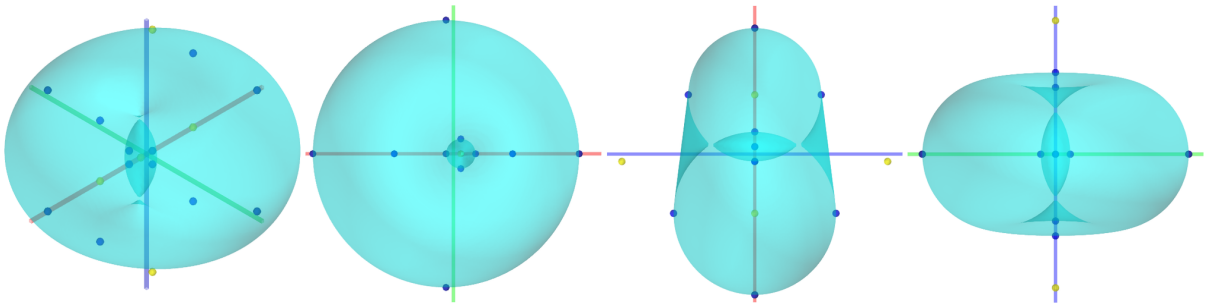

Figure 23. Spindle cyclide $\mathbf{\Phi}$, $(r_1+r_2)>2R$

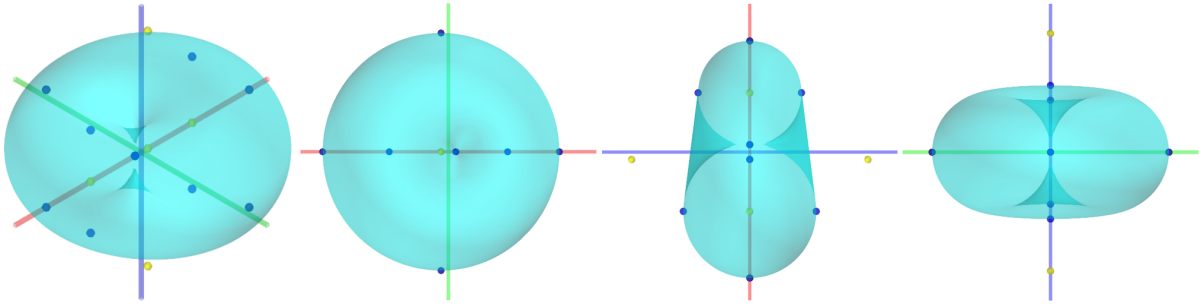

Figure 24. Horn cyclide $\mathbf{\Phi}$, $(r_1+r_2)=2R$

Figures 22,23,24 show three types of the Dupin cyclide $\mathbf{\Phi}$. The torus types, not shown, are ordinary toroid surfaces and they are exactly the same entities as formed by the toroid entity $\mathbf{O}$. The inversion of a circular cylinder in a sphere can form a *needle cyclide* [15], which is a ring cyclide having $r_1=0$ or $r_2=0$, $r_1\neq r_2$.

4.20.2 DCGA GIPNS horned Dupin cyclide

The *horned Dupin cyclide* is a modification of the *Dupin cyclide* that causes both of the initial spheres to shrink until they meet in points. The horned Dupin cyclide is formed by swapping μ and c in the implicit equation of the Dupin cyclide. The parameters a, b, c, μ are defined as

$$\begin{aligned} a &= R && (178)\\ \mu &= \frac{1}{2}(r_1 + r_2) && (179)\\ c &= \frac{1}{2}(r_1 - r_2) && (180)\\ b^2 &= a^2 - \mu^2. && (181) \end{aligned}$$

The DCGA GIPNS 2-vector *horned Dupin cyclide* surface entity $\mathbf{\Gamma}$ is defined as

$$\begin{aligned} \mathbf{\Gamma} = {} & T_{\mathbf{t}^4} + 2T_{\mathbf{t}^2}(b^2 - c^2) + \\ & -4a^2 T_{x^2} - 4b^2 T_{y^2} + \\ & 8a\,c\mu T_x + ((b^2 - c^2)^2 - 4c^2\mu^2)T_1. \end{aligned} \quad (182)$$

The DCGA GIPNS horned Dupin cyclide $\mathbf{\Gamma}$ has the following *related points*:

$$\begin{aligned} \text{Center of initial sphere } \mathbf{S}_1 \text{ with radius } r_1 &: -R\mathbf{e}_1\\ \text{Center of initial sphere } \mathbf{S}_2 \text{ with radius } r_2 &: +R\mathbf{e}_1\\ \text{Center of ring or spindle hole in the cyclide} &: +c\mathbf{e}_1\\ \text{Center of sphere enclosing entire cyclide} &: -c\mathbf{e}_1\\ \text{Radius around } -c\mathbf{e}_1 \text{ enclosing entire cyclide} &: \mu + R. \end{aligned}$$

Twelve *surface points* on the horned Dupin cyclide $\mathbf{\Gamma}$ are:

$$\begin{aligned} \text{2 surface points on } \mathbf{S}_1 \text{ with radius } r_1 &: -R\mathbf{e}_1 \pm r_1\mathbf{e}_1\\ \text{2 surface points on } \mathbf{S}_1 \text{ with radius } r_1 &: -R\mathbf{e}_1 \pm r_1\mathbf{e}_3\\ \text{2 surface points on } \mathbf{S}_2 \text{ with radius } r_2 &: +R\mathbf{e}_1 \pm r_2\mathbf{e}_1\\ \text{2 surface points on } \mathbf{S}_2 \text{ with radius } r_2 &: +R\mathbf{e}_1 \pm r_2\mathbf{e}_3\\ \text{2 surface points} &: -\mu\mathbf{e}_1 \pm (c + R)\mathbf{e}_2\\ \text{2 surface points} &: +\mu\mathbf{e}_1 \pm (c - R)\mathbf{e}_2. \end{aligned}$$

The horned Dupin cyclide $\mathbf{\Gamma}$ is initially created having these 12 points. All DCGA versor operations are valid on the horned Dupin cyclide $\mathbf{\Gamma}$ and can be used to rotate, dilate, and translate it into another general position.

The type of cyclide or torus represented by $\mathbf{\Gamma}$ is determined by:

$$\begin{aligned} \textit{Horned ring cyclide} \text{ when} &: (r_1 + r_2) < 2R\\ \textit{Horned spindle cyclide} \text{ when} &: (r_1 + r_2) > 2R\\ \textit{Horned spheres} \text{ (two } \textit{tangent spheres}\text{) when} &: (r_1 + r_2) = 2R\\ \textit{Horned ring torus} \text{ when} &: (r_1 = r_2) < R\\ \textit{Horned spindle torus} \text{ when} &: (r_1 = r_2) > R\\ \textit{Horned spheres} \text{ when} &: (r_1 = r_2) = R. \end{aligned}$$

The *horned spheres* represents the union or product of two implicit surface functions for two spheres of radius r_1 and r_2 that touch in a single tangent point, and it is an instance of a *spheres pair* cyclide entity. A different and more general spheres pair entity, the DCGA GIPNS 2-vector *spheres pair* entity $\boldsymbol{\xi}$, can be defined as the wedge of a CGA1 GIPNS sphere $\mathbf{S}_{1\mathcal{C}^1}$ and another CGA2 GIPNS sphere $\mathbf{S}_{2\mathcal{C}^2}$ as $\boldsymbol{\xi} = \mathbf{S}_{1\mathcal{C}^1} \wedge \mathbf{S}_{2\mathcal{C}^2}$. The spheres pair $\boldsymbol{\xi}$ can be transformed by the DCGA versors and intersected with standard DCGA spheres, planes, lines, and circles but not with any quadric or cyclidic surface entities.

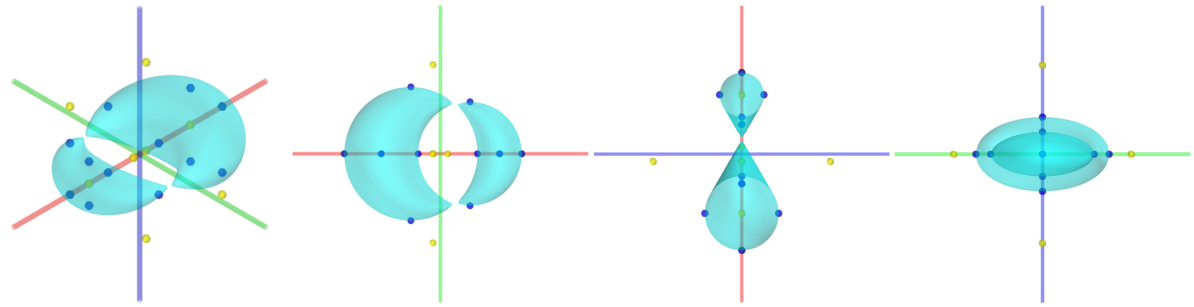

Figure 25. Horned ring cyclide $\boldsymbol{\Gamma}$, $(r_1 + r_2) < 2R$

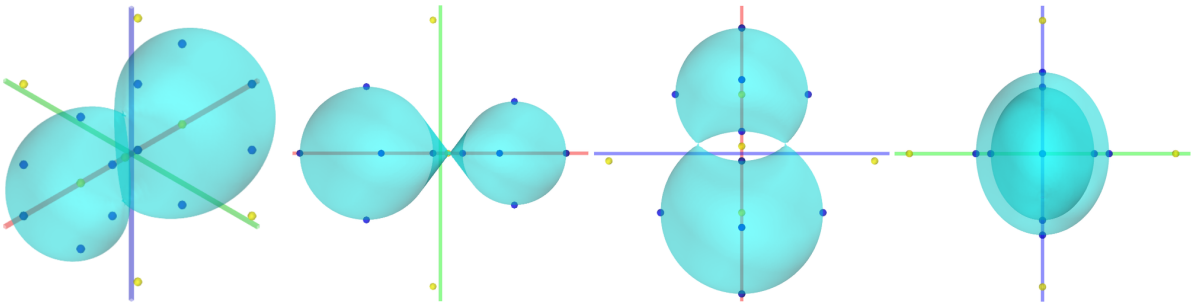

Figure 26. Horned spindle cyclide $\boldsymbol{\Gamma}$, $(r_1 + r_2) > 2R$

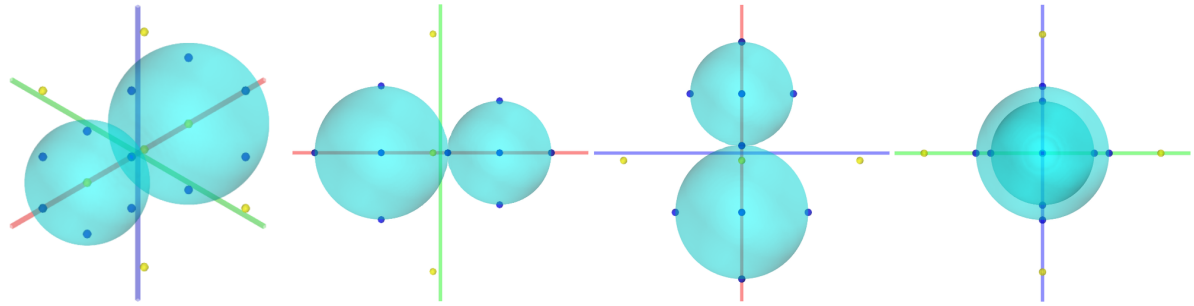

Figure 27. Horned spheres cyclide $\boldsymbol{\Gamma}$, $(r_1 + r_2) = 2R$

Figures 25,26,27 show three types of the horned Dupin cyclide $\boldsymbol{\Gamma}$. The three other types with $r_1 = r_2$, not shown, are symmetrical versions of the three types shown. As defined, the Dupin cyclides are symmetrical across the planes $y = 0$ and $z = 0$, and are also symmetrical across the plane $x = 0$ only when $r_1 = r_2$. The horned Dupin cyclides can be formed as the inversions of circular cones in spheres.

All of the Dupin cyclide entities have term $T_{\mathbf{t}^4} = -4\mathbf{e}_o$ and are true closed surfaces that do not have surface point $\mathbf{e}_\infty$. Therefore, the Dupin cyclide entities $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$, like the standard sphere $\mathbf{S}$ and toroid $\mathbf{O}$ entities, are well-behaved entities that do not have a singular outlier point at the inversion sphere center under inversion in a sphere.

4.20.3 DCGA GIPNS parabolic cyclide

The DCGA GIPNS 2-vector *parabolic cyclide* surface entity $\mathbf{\Psi}$ can be defined as

$$\begin{aligned}\mathbf{\Psi} &= BT_{\mathbf{t}^2} + CT_{x\mathbf{t}^2} + DT_{y\mathbf{t}^2} + ET_{z\mathbf{t}^2} + \\ &\quad FT_{x^2} + GT_{y^2} + HT_{z^2} + \\ &\quad IT_{xy} + JT_{yz} + KT_{zx} + \\ &\quad LT_x + MT_y + NT_z + OT_1\end{aligned} \tag{183}$$

with C, D, E not all zero. A *degenerate parabolic cyclide* has $C = D = E = 0$. All DCGA versor operations are valid on the parabolic cyclide entity $\mathbf{\Psi}$, and it can be intersected with the standard DCGA GIPNS sphere, plane, line, and circle entities that are defined as special bi-CGA entities.

The parabolic cyclide entity $\mathbf{\Psi}$ is simply the Darboux cyclide entity $\mathbf{\Omega}$ with $A = 0$, and it is a cubic surface entity. Without a term in $T_{\mathbf{t}^4} = -4\mathbf{e}_o$, the surface entity $\mathbf{\Psi}$ has surface point $\mathbf{e}_\infty$ and is generally an open-surface entity. The DCGA GIPNS 2-vector ellipsoid entity $\mathbf{E}$ is a degenerate parabolic cyclide entity that becomes a closed-surface entity with a singular outlier surface point at $\mathbf{e}_\infty$.

An instance of the DCGA 2-vector *parabolic cyclide* surface entity $\mathbf{\Psi}$ can be produced as the inversion of a DCGA GIPNS 2-vector Darboux cyclide surface entity $\mathbf{\Omega}$ in a standard DCGA GIPNS 2-vector sphere surface entity $\mathbf{S}$ that is *centered on a surface point* of $\mathbf{\Omega}$.

The inversion sphere $\mathbf{S}$, with center point $\mathbf{P}_\mathcal{D} = \mathcal{D}(\mathbf{p})$ *on the surface* of an entity $\mathbf{\Omega}$, gives the inverse surface $\mathbf{\Psi}$ as

$$\begin{aligned}\mathbf{\Psi} &= \mathbf{S}\mathbf{\Omega}\mathbf{S}^\sim = (\mathbf{S}_{\mathcal{C}^1} \wedge \mathbf{S}_{\mathcal{C}^2})\mathbf{\Omega}(\mathbf{S}_{\mathcal{C}^2} \wedge \mathbf{S}_{\mathcal{C}^1}) = \mathbf{S}_{\mathcal{C}^1}\mathbf{S}_{\mathcal{C}^2}\mathbf{\Omega}\mathbf{S}_{\mathcal{C}^2}\mathbf{S}_{\mathcal{C}^1} \\ &= \left(\mathbf{P}_{\mathcal{C}^1} - \frac{1}{2}r^2\mathbf{e}_{\infty 1}\right)\left(\mathbf{P}_{\mathcal{C}^2} - \frac{1}{2}r^2\mathbf{e}_{\infty 2}\right)\mathbf{\Omega}\mathbf{S}_{\mathcal{C}^2}\mathbf{S}_{\mathcal{C}^1}.\end{aligned} \tag{184}$$

If expanded further with the surface point condition $\mathbf{P}_\mathcal{D} \cdot \mathbf{\Omega} = 0$, then it is found that $(\mathbf{P}_\mathcal{D} = \mathbf{P}_{\mathcal{C}^1} \wedge \mathbf{P}_{\mathcal{C}^2}) \longleftrightarrow \mathbf{e}_\infty$, or that $\mathbf{P}_\mathcal{D}$ goes to $\mathbf{e}_\infty$ and $\mathbf{e}_\infty$ goes to $\mathbf{P}_\mathcal{D}$. Surface points of $\mathbf{\Omega}$ that are outside $\mathbf{S}$ are brought inside $\mathbf{S}$, and surface points of $\mathbf{\Omega}$ that are inside $\mathbf{S}$ are taken outside $\mathbf{S}$.

A surface point $\mathcal{D}(\mathbf{p}+\mathbf{d})$ of $\mathbf{\Omega}$ is transformed by inversion in sphere $\mathbf{S}$ centered at $\mathbf{p}$ with radius r as

$$\mathbf{S}\mathcal{D}(\mathbf{p}+\mathbf{d})\mathbf{S}^{-1} = \begin{cases} \mathcal{D}\left(\mathbf{p} + \frac{r^2}{\mathbf{d}^2}\mathbf{d}\right) & : \ \mathbf{d}^2 \neq 0 \\ \mathbf{e}_\infty & : \ \mathbf{d}^2 \to 0 \\ \mathcal{D}(\mathbf{p}) & : \ \mathbf{d}^2 \to \infty \\ \mathcal{D}(\mathbf{p}+\mathbf{d}^{-1}) & : \ r = 1 \\ \mathcal{D}(\mathbf{p}+\mathbf{d}) & : \ \mathbf{d}^2 = r^2 \end{cases} \tag{185}$$

$$\mathbf{S}^{-1} = \mathbf{S}^{-2}\mathbf{S} = \frac{1}{-r^4}\mathbf{S} = \frac{1}{r^4}\mathbf{S}^\sim. \tag{186}$$

The displacement $\mathbf{d}$ from the inversion sphere center $\mathbf{p}$ is literally inversed to $\mathbf{d}^{-1}$ when the inversion sphere has radius $r = 1$. As an *inver*sion operat*or*, an inversion sphere $\mathbf{S}$ could be called an *inversor*, especially if it has radius $r = 1$ where $\mathbf{S}^2 = -1$ as a proper *versor* with unit magnitude.

In general, an inversion sphere $\mathbf{S}$ can have any center point $\mathbf{P}_\mathcal{D}$ and any radius r. If $r = 0$, then $\mathbf{S} = \mathbf{P}_\mathcal{D}$, which is a finite point that could be $\mathbf{e}_o$. An infinite radius $r = \infty$ is represented by $\mathbf{S} = \mathbf{e}_\infty$. Inversion in any point $\mathbf{S} = \mathbf{P}_\mathcal{D}$ or $\mathbf{S} = \mathbf{e}_\infty$ sends everything into

the point and produces the point $\mathbf{S}$, *unless* the point $\mathbf{S}$ is a surface point of $\mathbf{\Omega}$ and then any point sent into itself produces the nil scalar 0 result for the entire surface inversion.

Open surfaces $\mathbf{\Psi}$ that extend out to infinity $\mathbf{e}_\infty$ are the only surfaces that reflect $\mathbf{S\Psi S}^\sim$ continuously into the center point $\mathbf{P}_\mathcal{D}$ of an inversion sphere $\mathbf{S}$. Conversely, by placing the center point $\mathbf{P}_\mathcal{D}$ of an inversion sphere $\mathbf{S}$ on any surface $\mathbf{\Omega}$ and then reflecting $\mathbf{S\Omega S}^\sim$ the surface outward, the resulting open surface $\mathbf{\Psi} = \mathbf{S\Omega S}^\sim$ extends out to infinity $\mathbf{e}_\infty$ and must be a plane or curved sheet that is either a parabolic cyclide $\mathbf{\Psi}$ or a degenerate parabolic cyclide that is one of the quadric surfaces. Quadric surfaces are degenerate parabolic cyclides, and all other curved-sheet cubic surface entities are instances of the parabolic cyclide entity.

Sphere, plane, line, and circle entities can be created as the *standard* DCGA GIPNS sphere, plane, line, and circle entities $\mathbf{S}$, $\mathbf{\Pi}$, $\mathbf{L}$, $\mathbf{C}$ which are defined as bi-CGA entities. Sphere, plane, line, and circle entities can also be created as non-standard entities that are instances of degenerate parabolic cyclides using only linear and quadratic extraction terms. Only the *standard* sphere, plane, line, and circle entities operate as inversion or reflection operators. All DCGA surface entities $\mathbf{\Upsilon}$ can be reflected in the standard sphere $\mathbf{S}$, plane $\mathbf{\Pi}$, line $\mathbf{L}$, and circle $\mathbf{C}$. Reflection in a line $\mathbf{L\Upsilon L}^\sim$ rotates $\mathbf{\Upsilon}$ by 180° around the line. Inversion in a circle $\mathbf{C\Upsilon C}^\sim$ is equal to a composition of a planar reflection and spherical inversion as $\mathbf{S\Pi\Upsilon\Pi S}$ where $\mathbf{C} = \mathbf{S} \wedge \mathbf{\Pi}$. The results of inversion or reflection operations on standard and non-standard sphere, plane, line, and circle entities in the standard ones are not the same. Reflections and inversions of the standard entities produce another one of the standard entities. Reflections and inversions of the non-standard entities can produce cubic surfaces that represent the expected surfaces but which also have a singular outlier surface point at the inversion sphere center point. All parabolic cyclides and degenerate parabolic cyclides have the point $\mathbf{e}_\infty$ which reflects into an inversion sphere center point.

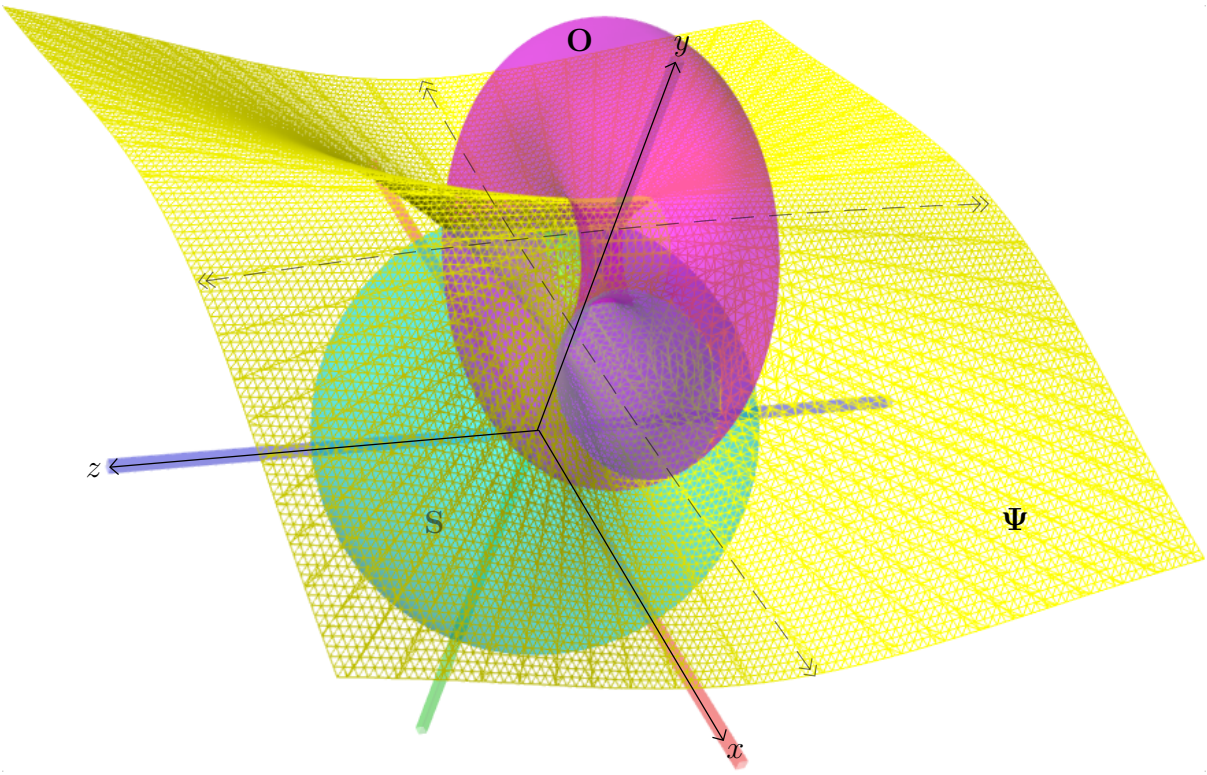

Figure 28. Toroid $\mathbf{O}$ on inversion sphere $\mathbf{S}$ center $\mathbf{P}_\mathcal{D} = \mathbf{e}_o$, $\mathbf{\Psi} = \mathbf{SOS}^\sim$

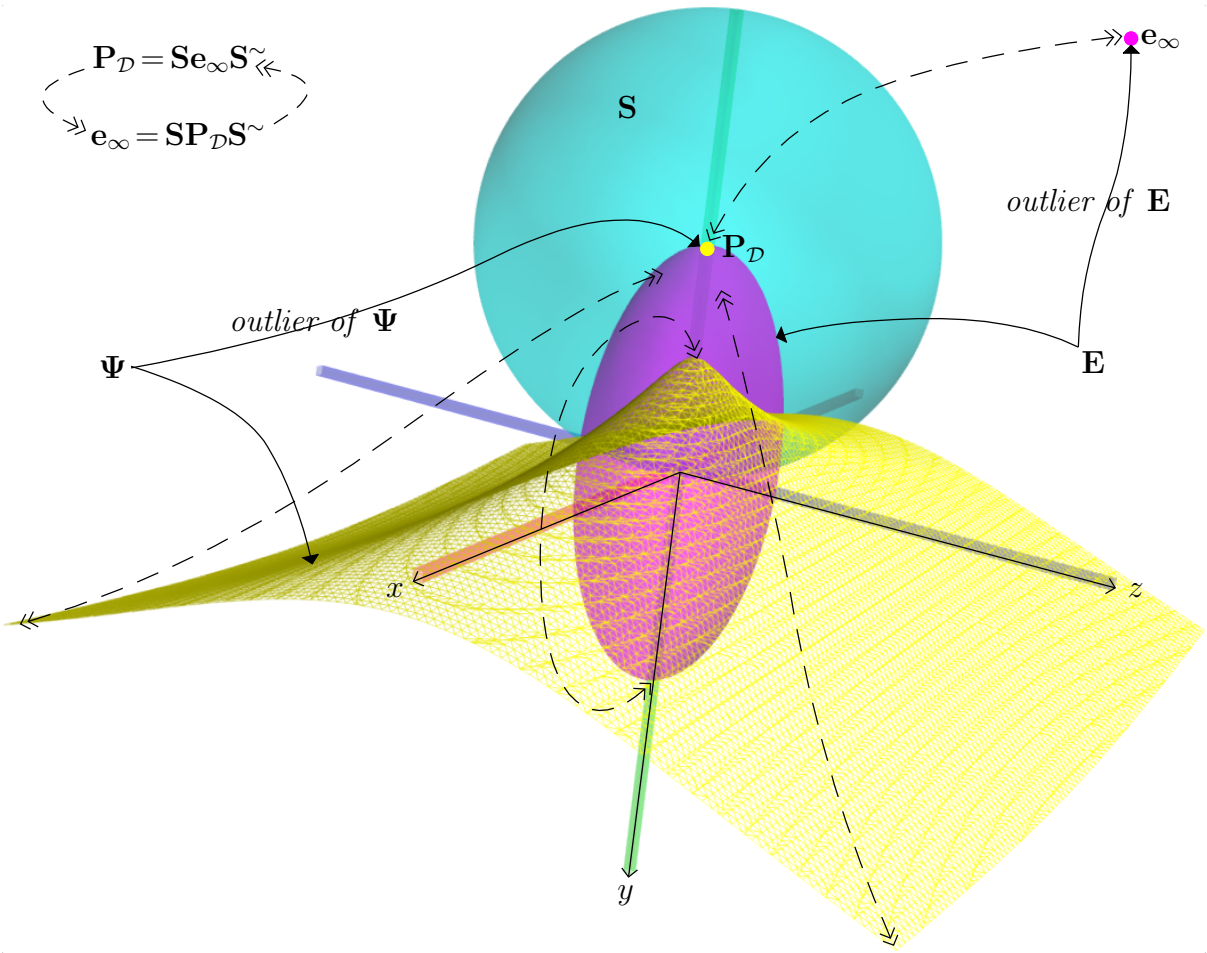

Figure 29. Ellipsoid $\mathbf{E}$ on inversion sphere $\mathbf{S}$ center $\mathbf{P}_\mathcal{D} = \mathcal{D}(-10\mathbf{e}_2)$, $\mathbf{\Psi} = \mathbf{S}\mathbf{E}\mathbf{S}^\sim$

5 DCGA GOPNS surfaces

Up to four DCGA points can be wedged to form DCGA *geometric outer product null space* (GOPNS) 4,6,8-vector surface entities of the surface types available in CGA. Unfortunately, the wedge of more than four points, as required for the quadric surfaces, does not work with DCGA points.

The DCGA GOPNS surface entities for quadric surfaces and the toroid would require more than four points to define them. For quadric surfaces in general position, it takes 5 points in 2D, and 9 points in 3D to define a quadric surface. If limited to principal axes-aligned surfaces, it still requires 6 points in 3D to define a quadric surface, as in QGA. Therefore, it seems that it is not possible in DCGA to directly represent the DCGA GOPNS quadric surfaces as the wedge of DCGA surface points. When more than four DCGA surface points are required to define a surface, then more complicated formulas are still possible but they resolve back to the GIPNS entities.

In general, we can always obtain a DCGA GOPNS surface entity $\mathbf{S}^{*\mathcal{D}}$ by taking the DCGA dual of a DCGA GIPNS surface entity $\mathbf{S}$ as $\mathbf{S}^{*\mathcal{D}} = \mathbf{S}/\mathbf{I}_\mathcal{D}$. All DCGA versor operations are valid on both the DCGA GIPNS entities and their dual DCGA GOPNS entities.

The following four subsections define the four DCGA GOPNS surface entities which can be constructed as wedges of up four DCGA surface points. These four DCGA GOPNS surface entities are just the DCGA analogues of the CGA GOPNS surface entities.

A DCGA test point $\mathbf{T}_{\mathcal{D}}$ that is on a DCGA GOPNS surface entity $\mathbf{S}^{*\mathcal{D}}$ must satisfy the GOPNS condition

$$\mathbf{T}_{\mathcal{D}} \wedge \mathbf{S}^{*\mathcal{D}} = 0.$$

The DCGA GOPNS k-vector surface entity $\mathbf{S}^{*\mathcal{D}}$ represents the set $\mathbb{NO}_G(\mathbf{S}^{*\mathcal{D}} \in \mathcal{G}^k_{8,2})$ of all 3D vector test points $\mathbf{t}$ that are surface points

$$\mathbb{NO}_G(\mathbf{S}^{*\mathcal{D}} \in \mathcal{G}^k_{8,2}) = \left\{ \mathbf{t} \in \mathcal{G}^1_3 \ : \ (\mathcal{D}(\mathbf{t}) = \mathbf{T}_{\mathcal{D}}) \wedge \mathbf{S}^{*\mathcal{D}} = 0 \right\}.$$

5.1 DCGA GOPNS sphere

The DCGA GOPNS 8-vector *sphere* $\mathbf{S}^{*\mathcal{D}}$ is defined as the wedge of four DCGA points $\mathbf{P}_{\mathcal{D}_i}$ on the sphere as

$$\begin{aligned} \mathbf{S}^{*\mathcal{D}} &= \mathbf{P}_{\mathcal{D}_1} \wedge \mathbf{P}_{\mathcal{D}_2} \wedge \mathbf{P}_{\mathcal{D}_3} \wedge \mathbf{P}_{\mathcal{D}_4} \\ &= \mathbf{S}/\mathbf{I}_{\mathcal{D}} \end{aligned} \tag{187}$$

and is the DCGA dual of the DCGA GIPNS 2-vector sphere $\mathbf{S}$.

5.2 DCGA GOPNS plane

The DCGA GOPNS 8-vector *plane* $\mathbf{\Pi}^{*\mathcal{D}}$ is defined as the wedge of three DCGA points $\mathbf{P}_{\mathcal{D}_i}$ on the plane and the DCGA point at infinity $\mathbf{e}_\infty$ as

$$\begin{aligned} \mathbf{\Pi}^{*\mathcal{D}} &= \mathbf{P}_{\mathcal{D}_1} \wedge \mathbf{P}_{\mathcal{D}_2} \wedge \mathbf{P}_{\mathcal{D}_3} \wedge \mathbf{e}_\infty \\ &= \mathbf{\Pi}/\mathbf{I}_{\mathcal{D}} \end{aligned} \tag{188}$$

and is the DCGA dual of the DCGA GIPNS 2-vector plane $\mathbf{\Pi}$.

5.3 DCGA GOPNS line

The DCGA GOPNS 6-vector *line* $\mathbf{L}^{*\mathcal{D}}$ is defined as the wedge of two DCGA points $\mathbf{P}_{\mathcal{D}_i}$ on the line and the DCGA point at infinity $\mathbf{e}_\infty$ as

$$\begin{aligned} \mathbf{L}^{*\mathcal{D}} &= \mathbf{P}_{\mathcal{D}_1} \wedge \mathbf{P}_{\mathcal{D}_2} \wedge \mathbf{e}_\infty \\ &= \mathbf{L}/\mathbf{I}_{\mathcal{D}} \end{aligned} \tag{189}$$

and is the DCGA dual of the DCGA GIPNS 4-vector line $\mathbf{L}$.

5.4 DCGA GOPNS circle

The DCGA GOPNS 6-vector *circle* $\mathbf{C}^{*\mathcal{D}}$ is defined as the wedge of three DCGA points $\mathbf{P}_{\mathcal{D}_i}$ on the circle as

$$\begin{aligned} \mathbf{C}^{*\mathcal{D}} &= \mathbf{P}_{\mathcal{D}_1} \wedge \mathbf{P}_{\mathcal{D}_2} \wedge \mathbf{P}_{\mathcal{D}_3} \\ &= \mathbf{C}/\mathbf{I}_{\mathcal{D}} \end{aligned} \tag{190}$$

and is the DCGA dual of the DCGA GIPNS 4-vector circle $\mathbf{C}$.

6 DCGA operations

The DCGA operations are very similar to the CGA operations, but the DCGA versors are the wedges of the two likewise CGA versors in both CGA1 and CGA2.

6.1 DCGA rotor

The DCGA rotor R is defined as

$$R = R_{\mathcal{C}^1} \wedge R_{\mathcal{C}^2}. \tag{191}$$

The CGA rotors for the same rotation operation in CGA1 and CGA2 are wedged as the DCGA rotor R. All DCGA entities $\mathbf{X}$, including both GIPNS and GOPNS, can be generally rotated around any axis by any angle by the DCGA rotor operation

$$\mathbf{X}' = R\mathbf{X}R^{\sim}. \tag{192}$$

6.2 DCGA dilator

The DCGA dilator D is defined as

$$D = D_{\mathcal{C}^1} \wedge D_{\mathcal{C}^2}. \tag{193}$$

The CGA dilators for the same dilation operation in CGA1 and CGA2 are wedged as the DCGA dilator D. All DCGA entities $\mathbf{X}$, including both GIPNS and GOPNS, can be dilated by the DCGA dilator operation

$$\mathbf{X}' = D\mathbf{X}D^{\sim}. \tag{194}$$

Keep in mind that dilation also dilates the position of an entity, which may cause an unexpected translational movement. To *scale* an entity, it should be translated to be centered on the origin, dilated around the origin, and then translated back.

6.3 DCGA translator

The DCGA translator T is defined as

$$T = T_{\mathcal{C}^1} \wedge T_{\mathcal{C}^2}. \tag{195}$$

The CGA translators for the same translation operation in CGA1 and CGA2 are wedged as the DCGA translator T. All DCGA entities $\mathbf{X}$, including both GIPNS and GOPNS, can be translated by the DCGA translator operation

$$\mathbf{X}' = T\mathbf{X}T^{\sim}. \tag{196}$$

6.4 DCGA motor

The DCGA motor M is defined as

$$M = M_{\mathcal{C}^1} \wedge M_{\mathcal{C}^2}. \tag{197}$$

The CGA motors for the same motion operation in CGA1 and CGA2 are wedged as the DCGA motor M. All DCGA entities $\mathbf{X}$, including both GIPNS and GOPNS, can be moved by the DCGA motor operation

$$\mathbf{X}' = M\mathbf{X}M^{\sim}. \tag{198}$$

6.5 DCGA intersection

Inversions in the *standard* DCGA GIPNS 2-vector sphere $\mathbf{S}$, and reflections in the standard DCGA GIPNS 2-vector plane $\mathbf{\Pi}$ are valid operations on all DCGA GIPNS entities. The dilator D operation is defined by inversions in spheres. The rotor R operation is defined by reflections in planes. These are operations that are known to work correctly, based on inversions and reflections.

If any DCGA GIPNS 2-vector entity $\boldsymbol{\Upsilon}$ and sphere $\mathbf{S}$ *are intersecting* in curve $\mathbf{X}$ on $\mathbf{S}$, then the inversion of $\boldsymbol{\Upsilon}$ in the sphere $\boldsymbol{\Omega} = \mathbf{S}\boldsymbol{\Upsilon}\mathbf{S}^{\sim}$ is also intersecting with both $\boldsymbol{\Upsilon}$ and $\mathbf{S}$ through the same intersection $\mathbf{X}$. If entity $\boldsymbol{\Upsilon}$ and plane $\boldsymbol{\Pi}$ are intersecting, then the reflection of $\boldsymbol{\Upsilon}$ in the plane $\boldsymbol{\Pi}\boldsymbol{\Upsilon}\boldsymbol{\Pi}^{\sim}$ is also intersecting with both $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Pi}$ through the same intersection. Planar reflection is a special case of spherical inversion when the sphere radius $r \to \infty$ and $\mathbf{S} \to \boldsymbol{\Pi}$ through three plane points. As $\boldsymbol{\Upsilon}$ becomes a sphere $\mathbf{S}_2$ or plane $\boldsymbol{\Pi}$ denoted by $\boldsymbol{\Upsilon} \to \mathbf{S}_2 | \boldsymbol{\Pi}$, then $\boldsymbol{\Omega} \to \boldsymbol{\Pi} | \mathbf{S}_2$, and $\mathbf{X} \to \mathbf{C}$ to form a circular intersection, but $\boldsymbol{\Omega}$ and $\mathbf{X}$ have various possible surface and curve shapes when $\boldsymbol{\Upsilon} \neq \mathbf{S}_2 | \boldsymbol{\Pi}$.

For the inversion, we have

$$\begin{aligned} \boldsymbol{\Omega} &= \mathbf{S}\boldsymbol{\Upsilon}\mathbf{S}^{\sim} = (\mathbf{S}\cdot\boldsymbol{\Upsilon} + \mathbf{S}\times\boldsymbol{\Upsilon} + \mathbf{S}\wedge\boldsymbol{\Upsilon})\mathbf{S}^{\sim} \qquad (199)\\ &= (\mathbf{S}\cdot\boldsymbol{\Upsilon})\mathbf{S}^{\sim} + (\boldsymbol{\Upsilon}\times\mathbf{S})\mathbf{S} - \mathbf{S}^2(\boldsymbol{\Upsilon}\wedge\mathbf{S})\mathbf{S}\mathbf{S}^{-2} \\ &= (\mathbf{S}\cdot\boldsymbol{\Upsilon})\mathbf{S}^{\sim} + \frac{1}{2}(\boldsymbol{\Upsilon}\mathbf{S} - \mathbf{S}\boldsymbol{\Upsilon})\mathbf{S} - \mathbf{S}^2\mathcal{P}_{\mathbf{S}}^{\perp}(\boldsymbol{\Upsilon}) \\ &= 2(\mathbf{S}\cdot\boldsymbol{\Upsilon})\mathbf{S}^{\sim} + \boldsymbol{\Upsilon}\mathbf{S}^2 - 2\mathbf{S}^2\mathcal{P}_{\mathbf{S}}^{\perp}(\boldsymbol{\Upsilon}) \\ \mathbf{X} = \boldsymbol{\Omega}\wedge\mathbf{S} &= \mathbf{S}^2(\boldsymbol{\Upsilon}\wedge\mathbf{S}) - 2\mathbf{S}^2\mathcal{P}_{\mathbf{S}}^{\perp}(\boldsymbol{\Upsilon})\mathbf{S} = -\mathbf{S}^2(\boldsymbol{\Upsilon}\wedge\mathbf{S}). \qquad (200) \end{aligned}$$

This expression of $\mathbf{X}$ implies that $\mathbf{X}$ represents *something* in common with both $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Omega}$ in relation to $\mathbf{S}$. That something is their intersection, and $\mathbf{X}$ is the entity that constructs and represents their intersection.

The product $\times$ is the commutator product on 2-vectors that produces another 2-vector. The operation $\mathcal{P}_{\mathbf{S}}^{\perp}(\boldsymbol{\Upsilon}) = (\boldsymbol{\Upsilon}\wedge\mathbf{S})\mathbf{S}^{-1}$ is the perpendicular projection or *rejection* of $\boldsymbol{\Upsilon}$ from $\mathbf{S}$, and $\mathcal{P}_{\mathbf{S}}(\boldsymbol{\Upsilon}) = (\boldsymbol{\Upsilon}\cdot\mathbf{S})\mathbf{S}^{-1}$ is the parallel *projection* of $\boldsymbol{\Upsilon}$ on $\mathbf{S}$ [8]. The operation $\mathcal{P}_{\mathbf{S}}^{\times}(\boldsymbol{\Upsilon}) = (\boldsymbol{\Upsilon}\times\mathbf{S})\mathbf{S}^{-1}$ is another projection of $\boldsymbol{\Upsilon}$ on $\mathbf{S}$. For 2-vectors, $\mathbf{S}\times\boldsymbol{\Upsilon} = -\boldsymbol{\Upsilon}\times\mathbf{S}$, but $\mathbf{S}\wedge\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}\wedge\mathbf{S}$. Note that $\mathbf{S}^{\sim} = -\mathbf{S} = -\mathbf{S}^2\mathbf{S}^{-1}$, and $\mathbf{S}^2 = -r^4$ where r is the radius of sphere $\mathbf{S}$. If $r \to \infty$, then $\mathbf{S} \to \boldsymbol{\Pi}$, $\mathbf{S}^2 \to \boldsymbol{\Pi}^2 = -1$, and $\mathbf{X} = \boldsymbol{\Omega}\wedge\mathbf{S} = \boldsymbol{\Upsilon}\wedge\mathbf{S}$.

The pair of inverse surface entities $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Omega}$ could also be defined as

$$\begin{aligned} \boldsymbol{\Upsilon} &= \boldsymbol{\Upsilon}\mathbf{S}\mathbf{S}^{-1} = \mathcal{P}_{\mathbf{S}}(\boldsymbol{\Upsilon}) + \mathcal{P}_{\mathbf{S}}^{\times}(\boldsymbol{\Upsilon}) + \mathcal{P}_{\mathbf{S}}^{\perp}(\boldsymbol{\Upsilon}) \qquad (201)\\ \boldsymbol{\Omega} &= \mathbf{S}\boldsymbol{\Upsilon}\mathbf{S}^{-1} = \mathcal{P}_{\mathbf{S}}(\boldsymbol{\Upsilon}) - \mathcal{P}_{\mathbf{S}}^{\times}(\boldsymbol{\Upsilon}) + \mathcal{P}_{\mathbf{S}}^{\perp}(\boldsymbol{\Upsilon}) \qquad (202) \end{aligned}$$

and then $\mathbf{X} = \boldsymbol{\Omega}\wedge\mathbf{S} = \boldsymbol{\Upsilon}\wedge\mathbf{S}$ exactly, but $\boldsymbol{\Omega} = \mathbf{S}\boldsymbol{\Upsilon}\mathbf{S}^{\sim}$ will be assumed henceforth.

The test for a point $\mathbf{T}_{\mathcal{D}}$ on the intersection entity $\mathbf{X}$ is

$$\mathbf{T}_{\mathcal{D}}\cdot\mathbf{X} = \mathbf{T}_{\mathcal{D}}\cdot(\boldsymbol{\Omega}\wedge\mathbf{S}) = -\mathbf{S}^2\mathbf{T}_{\mathcal{D}}\cdot(\boldsymbol{\Upsilon}\wedge\mathbf{S}) \qquad (203)$$

where if $\mathbf{T}_{\mathcal{D}}\cdot\mathbf{X} = 0$, then the point $\mathbf{T}_{\mathcal{D}}$ is on the intersection of all three surfaces represented by $\mathbf{S}, \boldsymbol{\Upsilon}$, and $\boldsymbol{\Omega}$.

The DCGA GIPNS 4-vector intersection entity $\mathbf{X} = \boldsymbol{\Upsilon}\wedge\mathbf{S}$ is derived from the inversion of $\boldsymbol{\Upsilon}$ in $\mathbf{S}$. Proving this precisely may be simple if certain algebraic steps are taken correctly. $\boldsymbol{\Upsilon}, \boldsymbol{\Omega}$, and $\mathbf{X}$ are generally *not blades*, and $\mathbf{T}_{\mathcal{D}}$ is a null 2-blade. Most of the usual algebraic identities are valid only on blades that are the product of non-null vectors or blades. Therefore, the algebraic steps leading to a clear proof may require unusual identities or other results.

DCGA GIPNS intersection entities $\boldsymbol{\Upsilon}\wedge\boldsymbol{\Pi}$, $\boldsymbol{\Upsilon}\wedge\mathbf{L}$, and $\boldsymbol{\Upsilon}\wedge\mathbf{C}$, for intersections of any DCGA GIPNS entity $\boldsymbol{\Upsilon}$ with the standard DCGA GIPNS 2-vector plane $\boldsymbol{\Pi}$, 4-vector line $\mathbf{L}$, and 4-vector circle $\mathbf{C}$, are also derived from reflections or inversions. Line $\mathbf{L} = \boldsymbol{\Pi}_1\wedge\boldsymbol{\Pi}_2$ and circle $\mathbf{C} = \mathbf{S}\wedge\boldsymbol{\Pi}$ are just intersections of sphere and plane entities.

Inversions or reflections work only in the standard DCGA 2-vector sphere $\mathbf{S}$ and plane $\mathbf{\Pi}$. Inversions do not work in other entities, and therefore other entities cannot form intersection entities with each other. For example, the DCGA GIPNS 2-vector quadric surface entities do not work as inversion or reflection operators, and therefore they cannot form intersection entities with each other. The intersection of two entities depends on at least one of them being a valid inversion operator on the other entity.

In symbolic calculations, the simplest intersection entities, such as $\mathbf{X} = \mathbf{\Upsilon} \wedge \mathbf{S}$, are 4-vectors with many 4-blade terms or components. The scalar magnitude of each 4-blade is an implicit surface function for a surface that is coincident with the intersection represented by $\mathbf{\Upsilon} \wedge \mathbf{S}$. Not every blade holds a unique implicit surface function, but the number of unique functions can exceed ten. Figures 30 and 31 are plots of intersections that are showing all of the unique implicit surfaces that are extracted from the blades of the intersection entities.

The foregoing discussion has not given any rigorous proof of the correctness of the intersection entities. $\mathcal{G}_{8,2}$ DCGA is a large and complicated pseudo-Euclidean algebra, and there could be unforeseen cases where intersections do not work as expected. Therefore, the following box serves as a mild warning before continuing.

Although not rigorously proved here, the intersection tests performed by this author supported the following claims given in this subsection about DCGA intersection. Detailed examinations of ellipsoid-plane and ellipsoid-sphere intersections are shown in Figures 30 and 31. These claims should be considered preliminary, and require additional research to prove for certain what intersections are valid or invalid.

The set $\mathcal{S} = \{\mathbf{S}, \mathbf{\Pi}\}$ of *standard bi-CGA GIPNS entities* includes all instances of the DCGA GIPNS 2-vector sphere $\mathbf{S}$ and plane $\mathbf{\Pi}$. These two entities are defined in previous sections on them. The DCGA GIPNS 4-vector line $\mathbf{L} = \mathbf{\Pi}_1 \wedge \mathbf{\Pi}_2$ and circle $\mathbf{C} = \mathbf{S} \wedge \mathbf{\Pi}$ are extended standard bi-CGA GIPNS entities that are the intersections of spheres and planes.

The DCGA GIPNS *intersection* entity $\mathbf{X}$ is the wedge of $2 \leq n \leq 4$ standard bi-CGA GIPNS entities $\mathbf{B}_i \in \mathcal{S}$, or is the wedge of $1 \leq n \leq 3$ entities $\mathbf{B}_i \in \mathcal{S}$ and *one* DCGA GIPNS 2-vector entity $\mathbf{A}_{\langle 2 \rangle} \notin \mathcal{S}$ that is not a standard bi-CGA GIPNS entity. Only *one* DCGA GIPNS 2-vector Darboux cyclide surface entity $\mathbf{A}_{\langle 2 \rangle} = \mathbf{\Omega}$ (or any degenerate) can be included in a wedge that forms an intersection entity $\mathbf{X}$. Unfortunately, the Darboux cyclide entities, including the quadric surfaces, cannot be intersected directly with each other by wedge products since they are invalid inversion operators. These claims are summarized by the following definition.

The DCGA GIPNS *intersection* entity $\mathbf{X}$ of grade $4 \leq k \leq 8$ is defined as

$$\mathbf{X}_{\langle 4 \leq k \leq 8 \rangle} = \begin{cases} \bigwedge_{i=1}^{2 \leq n \leq 4} \mathbf{B}_i & : \ \mathbf{B}_i \in \mathcal{S} \text{ and } \mathcal{S} = \{\mathbf{S}, \mathbf{\Pi}\} \\ \mathbf{A}_{\langle 2 \rangle} \wedge \bigwedge_{i=1}^{1 \leq n \leq 3} \mathbf{B}_i & : \ \mathbf{A}_{\langle 2 \rangle} \notin \mathcal{S} \text{ and } \mathbf{B}_i \in \mathcal{S}. \end{cases} \tag{204}$$

The maximum grade for a valid intersection entity $\mathbf{X}$ is grade 8. The grade of the wedges is divisible by 2, making the next grade above 8 to be 10, proportional to the DCGA unit pseudoscalar $\mathbf{I}_{\mathcal{D}}$. No valid entity is a pseudoscalar.

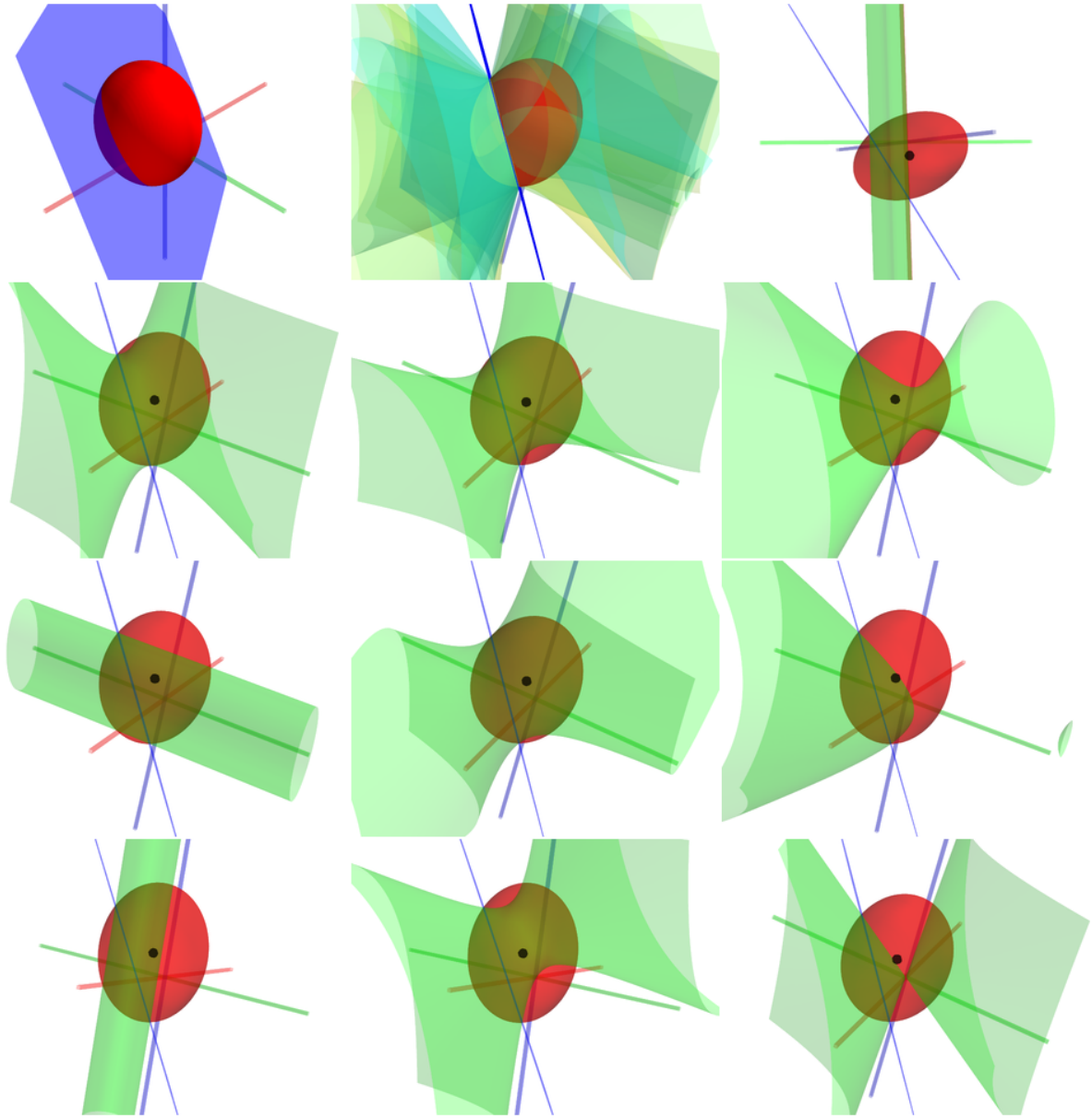

Figure 30. Intersection of ellipsoid and plane in general positions

Figure 30 shows the details of a DCGA GIPNS 4-vector intersection entity $\mathbf{E} \wedge \mathbf{\Pi}$ representing the intersection of a DCGA GIPNS 2-vector ellipsoid $\mathbf{E}$ and DCGA GIPNS 2-vector plane $\mathbf{\Pi}$, both rotated and translated differently into general positions that have an intersection. The red ellipsoid $\mathbf{E}$ has initial parameters $r_x = 5$, $r_y = 7$, $r_z = 9$, $p_x = 1$, $p_y = -2$, $p_z = 3$, and is then rotated 30° around the blue z-axis. The *Sympy* test code for the ellipsoid was:

```
Rotor(e3,30*pi*Pow(180,-1))*
GIPNS_Ellipsoid(1,-2,3,5,7,9)*
Rotor(e3,30*pi*Pow(180,-1)).rev()
```

The black dot is the ellipsoid center position. The blue plane $\mathbf{\Pi}$ is initially perpendicular to the x-axis through the origin, then transformed according to the following code:

```
    Rotor(e1,30*pi*Pow(180,-1))*
    Translator(-4*e2)*
    Rotor(e3,-60*pi*Pow(180,-1))*
    GIPNS_Plane(e1,0)*
    Rotor(e3,-60*pi*Pow(180,-1)).rev()*
    Translator(-4*e2).rev()*
    Rotor(e1,30*pi*Pow(180,-1)).rev()
```

Their DCGA GIPNS intersection is $\mathbf{X} = \mathbf{E} \wedge \mathbf{\Pi}$. The various images in Figure 30 show components of $\mathbf{X}$ that represent other surfaces that are all coincident with the intersection of the ellipsoid and plane. There were ten unique components in $\mathbf{X}$. These components are cylinders, hyperboloids, and a cone. The intersection entity $\mathbf{X}$ represents the locus of points that are simultaneously located on all ten of these surfaces, which is an ellipse-shaped intersection of the ellipsoid and plane.

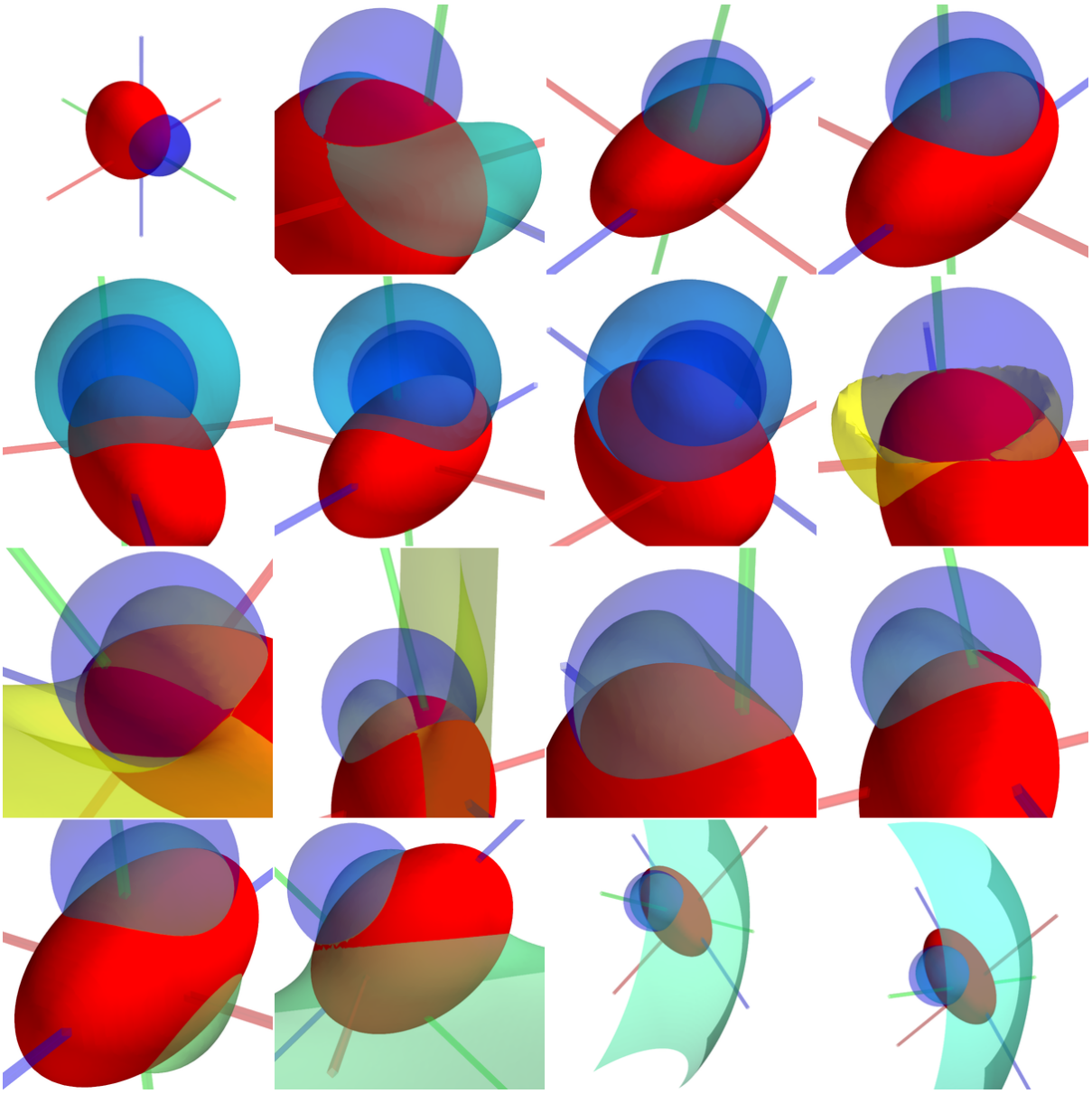

Figure 31. Intersection of ellipsoid and sphere in general positions

Figure 31 shows the same red DCGA GIPNS ellipsoid $\mathbf{E}$ as in Figure 30, but now

intersected with a blue DCGA GIPNS sphere $\mathbf{S}$ of radius $r=5$ at position $\mathbf{e}_1+5\mathbf{e}_2+3\mathbf{e}_3$. The DCGA GIPNS intersection entity is now $\mathbf{X}=\mathbf{E}\wedge\mathbf{S}$. The shape of the intersection appears like a curved ellipse or curved circle. The components of the entity $\mathbf{X}$ represent 15 other unique surfaces that are also coincident with the intersection of $\mathbf{E}$ and $\mathbf{S}$. The images of Figure 31 show how each of these 15 surfaces intersect with the intersection of $\mathbf{E}$ and $\mathbf{S}$. Some of these surfaces are unusually shaped, and some have two sheets. The DCGA GIPNS intersection entity $\mathbf{X}$ represents the simultaneous locus or intersection of all of the involved surfaces and appears to be a valid intersection entity for the ellipsoid and sphere.

The DCGA GIPNS quadric surface entities, of the types not available in CGA, could *not* be wedged with each other to form valid intersection entities - *incorrect* or *invalid* intersection entities resulted from their wedge. More generally, the DCGA GIPNS Darboux cyclide entities $\mathbf{\Omega}$ cannot be intersected with each other by wedge products, but *one* can be intersected with standard bi-CGA GIPNS entities. As a curiosity, it was noticed that the sum and the difference of two intersecting DCGA GIPNS quadric surface entities represent two more coincident intersecting surfaces.

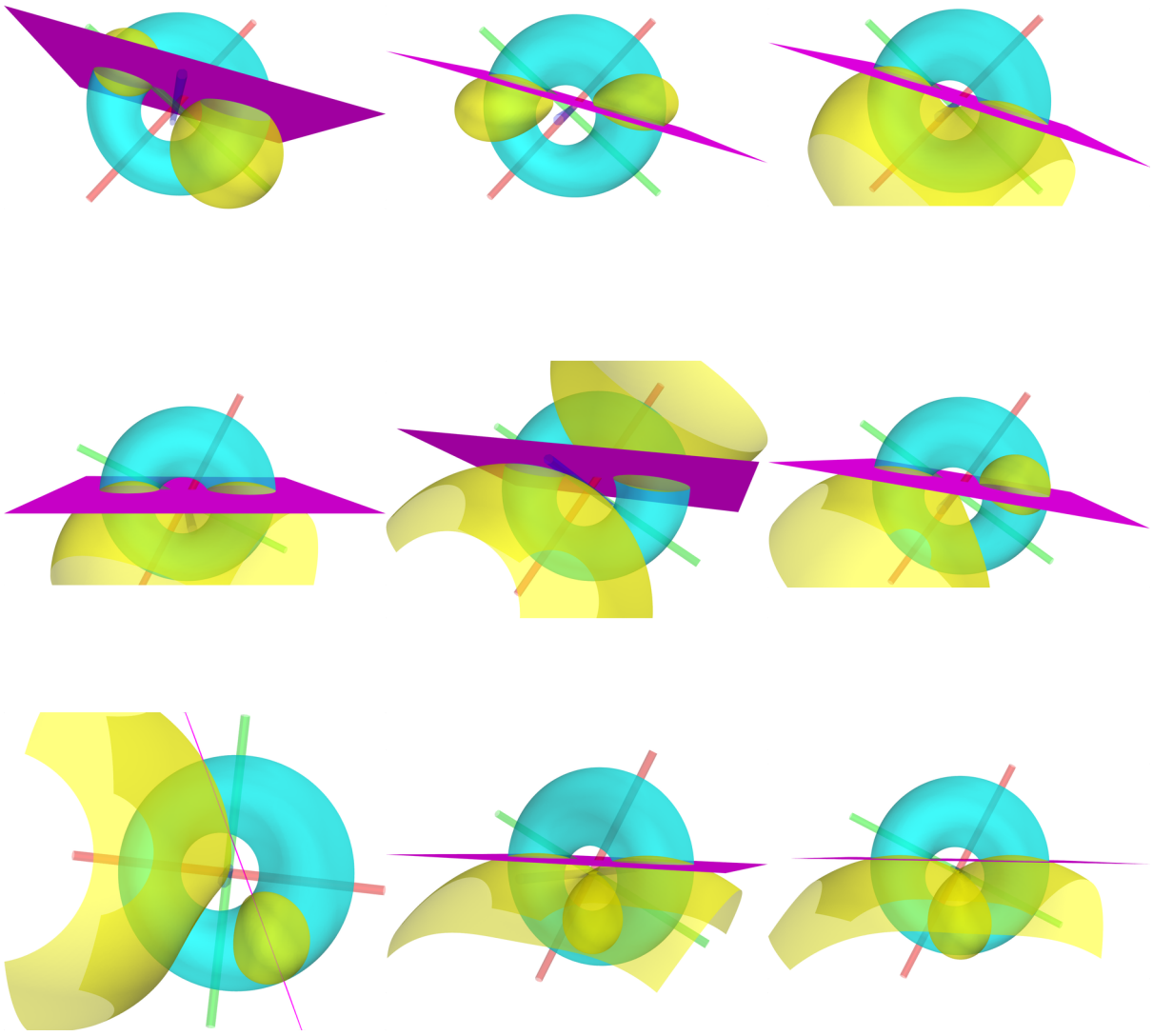

Figure 32. Intersection $\mathbf{\Phi}\wedge\mathbf{\Pi}$ of ring Dupin cyclide $\mathbf{\Phi}$ and plane $\mathbf{\Pi}$

6.6 DCGA dualization

The DCGA unit pseudoscalar $\mathbf{I}_{\mathcal{D}}$ is defined as

$$\begin{aligned}\mathbf{I}_{\mathcal{D}} &= \mathbf{I}_{\mathcal{C}^1} \wedge \mathbf{I}_{\mathcal{C}^2} \\ &= \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4\mathbf{e}_5\mathbf{e}_6\mathbf{e}_7\mathbf{e}_8\mathbf{e}_9\mathbf{e}_{10}\end{aligned} \tag{205}$$

and is the DCGA dualization operator on all DCGA entities.

Properties of $\mathbf{I}_{\mathcal{D}}$ include

$$\mathbf{I}_{\mathcal{D}}^{\sim} = (-1)^{10(10-1)/2}\mathbf{I}_{\mathcal{D}} = -\mathbf{I}_{\mathcal{D}} \tag{206}$$

$$\mathbf{I}_{\mathcal{D}}^{2} = -\mathbf{I}_{\mathcal{D}}\mathbf{I}_{\mathcal{D}}^{\sim} = -1 \tag{207}$$

$$\mathbf{I}_{\mathcal{D}}^{-1} = \mathbf{I}_{\mathcal{D}}^{\sim} = -\mathbf{I}_{\mathcal{D}}. \tag{208}$$

According to the sign rule $(-1)^{r(10-1)}$ for the commutation of the inner product of two blades, the DCGA pseudoscalar $\mathbf{I}_{\mathcal{D}}$ commutes with blades of even grade r, such as the DCGA 2-vector points, DCGA GIPNS 2,4,6,8-vector surfaces, and their dual DCGA GOPNS surfaces.

A DCGA GIPNS k-vector surface entity $\mathbf{X}$ is *dualized* into its *dual* DCGA GOPNS $(10-k)$-vector surface entity $\mathbf{X}^{*\mathcal{D}}$ as

$$\mathbf{X}^{*\mathcal{D}} = \mathbf{X}/\mathbf{I}_{\mathcal{D}} = -\mathbf{X}\cdot\mathbf{I}_{\mathcal{D}}. \tag{209}$$

A DCGA GOPNS k-vector surface entity $\mathbf{X}^{*\mathcal{D}}$ is *undualized* into its *undual* DCGA GIPNS $(10-k)$-vector surface entity $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{X}^{*\mathcal{D}}\mathbf{I}_{\mathcal{D}} = \mathbf{X}^{*\mathcal{D}}\cdot\mathbf{I}_{\mathcal{D}}. \tag{210}$$

This definition of dual and undual preserves the sign on the entities, otherwise the dual applied twice changes signs.

It is understandable that many authors may call the GIPNS entities *dual* and the GOPNS entities *standard*, but since in DCGA we cannot wedge DCGA points into all of the GOPNS entities, the GIPNS entities are considered the *standard* or *undual* entities and the GOPNS entities are the *dual* entities. Most of the DCGA GOPNS entities can only be obtained by the dualization operation as duals.

7 DCGA computing using Gaalop

The Geometric Algebra Algorithms Optimizer *Gaalop* is a Geometric Algebra Computing and Visualization software developed by DIETMAR HILDENBRAND et al. Hildenbrand introduces Gaalop in his book [9] and recent paper [6]. The Gaalop Precompiler is introduced by PATRICK CHARRIER in his master's thesis [2]. The Visualizer plugin for Gaalop is introduced by CHRISTIAN STEINMETZ in his master's thesis [16].

Gaalop uses the **CLUScript** domain-specific scripting language for Geometric Algebra Algorithms that was originally written and conceived by CHRISTIAN PERWASS as the scripting module of his CLUCALC and CLUVIZ software packages. Perwass introduces CLUCalc in his book [12].

Using Gaalop, geometric algebra computations or algorithms can be developed in CLUScript. Gaalop plugins can translate or compile the CLUScript into other programming languages or codes for applications or visualizations. Gaalop is a tool that enables Geometric Algebra to be used as a practical unified mathematical language for science and engineering applications. Gaalop allows the user to define and use arbitrary geometric algebras $\mathcal{G}_{p,q,r}$ of p positive-signature vector-units, q negative-signature vector-units, and r null vector-units. Gaalop comes preconfigured to support many well-known geometric algebras that have been introduced in papers and books. $\mathcal{G}_{8,2}$ DCGA, being introduced only now in this paper, was unknown and therefore not preconfigured in Gaalop at the time of writing this paper.

The remainder of this section shows how to define DCGA in Gaalop and then to visualize some of the DCGA entities using the Visualizer plugin.

7.1 definition.csv

Following Steinmetz's thesis, to define our "Own" algebras in Gaalop, we start by creating a folder, which we can call `algebras`, to hold subfolders for each algebra we want to define. Next, we create the subfolder `algebras/DCGA` to hold the files that define the DCGA algebra. The first file we must create is `algebras/DCGA/definition.csv`, which must contain the following 5 lines (the 2nd and 5th lines are blank):

```
1,e1,e2,e3,e4,e5,e6,e7,e8,e9,e10

1,e1,e2,e3,e4,e5,e6,e7,e8,e9,e10
e1=1,e2=1,e3=1,e4=1,e5=-1,e6=1,e7=1,e8=1,e9=1,e10=-1
```

7.2 macros.clu

The `definition.csv` file only defines the generic $\mathcal{G}_{8,2}$ Geometric Algebra basis vectors. The next file we must create is `algebras/DCGA/macros.clu`, which contains CLUScript macro/function definitions that define everything else that is specific to DCGA:

```
// DCGA macros.clu
// CGA1, CGA2, and DCGA origin and infinity points
eo1 = { (1/2)*(-e4+e5) }
eo2 = { (1/2)*(-e9+e10) }
ei1 = { e4+e5 }
ei2 = { e9+e10 }
eo = { eo1()^eo2() }
ei = { ei1()^ei2() }

// DCGA point value-extraction operators for defining DCGA GIPNS 2-vector entities
Tx = { (1/2)*(e1^ei2()+ei1()^e6) }
Ty = { (1/2)*(e2^ei2()+ei1()^e7) }
Tz = { (1/2)*(e3^ei2()+ei1()^e8) }
Txy = { (1/2)*(e7^e1+e6^e2) }
Tyz = { (1/2)*(e7^e3+e8^e2) }
Tzx = { (1/2)*(e8^e1+e6^e3) }
Txx = { e6^e1 }
```

```
Tyy = { e7^e2 }
Tzz = { e8^e3 }
Txt2 = { e1^eo2()+eo1()^e6 }
Tyt2 = { e2^eo2()+eo1()^e7 }
Tzt2 = { e3^eo2()+eo1()^e8 }
T1 = { -ei() }
Tt2 = { eo2()^ei1()+ei2()^eo1() }
Tt4 = { -4*eo() }

// Pseudoscalars
IE1 = { e1^e2^e3 }
IE2 = { e6^e7^e8 }
IC1 = { e1^e2^e3^e4^e5 }
IC2 = { e6^e7^e8^e9^e10 }
ID =  { e1^e2^e3^e4^e5^e6^e7^e8^e9^e10 }

// DCGA dualization macro
DD =  { -_P(1).ID() }
// CGA1 and CGA2 dualizations
C1D = { -_P(1).IC1() }
C2D = { -_P(1).IC2() }
// Euclidean 1 and Euclidean 2 dualizations
E1D = { -_P(1).IE1() }
E2D = { -_P(1).IE2() }
// special Dual macro for DCGA dualization operator *
Dual = { DD(_P(1)) }

// Normalize a non-null vector
Normalize = { _P(1)/sqrt(_P(1)._P(1)) }
// Convert degrees angle to radians
Deg2Rad = { _P(1)*acos(-1)/180 }

// CGA1_Point(x,y,z) representing point at (x,y,z) in Euclidean 3D
CGA1_Point = {
 _P(1)*e1 + _P(2)*e2 + _P(3)*e3 +
 (1/2)*(_P(1)*_P(1) + _P(2)*_P(2) + _P(3)*_P(3))*ei1() + eo1()
}
// CGA2_Point(x,y,z) representing point at (x,y,z) in Euclidean 3D
CGA2_Point = {
 _P(1)*e6 + _P(2)*e7 + _P(3)*e8 +
 (1/2)*(_P(1)*_P(1) + _P(2)*_P(2) + _P(3)*_P(3))*ei2() + eo2()
}
// special DCGA createPoint macro
createPoint = { CGA1_Point(_P(1),_P(2),_P(3))^CGA2_Point(_P(1),_P(2),_P(3)) }
DCGA_Point = { createPoint(_P(1),_P(2),_P(3)) }
Normalize_CGA1_Point = { _P(1)/(-_P(1).ei1()) }
Normalize_CGA2_Point = { _P(1)/(-_P(1).ei2()) }
Normalize_DCGA_Point = { _P(1)/(-_P(1).ei()) }
// Embed vector as CGA1 point (shorter name for CGA1_Point)
EV1 = { CGA1_Point(_P(1),_P(2),_P(3)) }
// Embed vector as CGA2 point (shorter name for CGA2_Point)
EV2 = { CGA2_Point(_P(1),_P(2),_P(3)) }
// Embed vector as DCGA point (shorter name for DCGA_Point)
EV = { DCGA_Point(_P(1),_P(2),_P(3)) }
// Project CGA1 point back to a vector in Euclidean 1 space
PV1 = { -(Normalize_CGA1_Point(_P(1)).IE1()).IE1() }
// Project CGA2 point back to a vector in Euclidean _2_ space
PV2 = { -(Normalize_CGA2_Point(_P(1)).IE2()).IE2() }
// Project DCGA point back to a vector in Euclidean 1 space
PV = { PV1(_P(1).ei2()) }

// Rotor(x,y,z,a) with axis (x,y,z) and rotation angle a in _degrees_
CGA1_Rotor = { t = Deg2Rad(_P(4));
 cos(t/2) + sin(t/2)*E1D(Normalize(_P(1)*e1 + _P(2)*e2 + _P(3)*e3))
```

```
}
CGA2_Rotor = { t = Deg2Rad(_P(4));
 cos(t/2) + sin(t/2)*E2D(Normalize(_P(1)*e6 + _P(2)*e7 + _P(3)*e8))
}
Rotor = {
 CGA1_Rotor(_P(1),_P(2),_P(3),_P(4))^CGA2_Rotor(_P(1),_P(2),_P(3),_P(4))
}
// Dilator(d) with scalar dilation factor d
CGA1_Dilator = { (1/2)*(1+_P(1)) + (1/2)*(1-_P(1))*(ei1()^eo1()) }
CGA2_Dilator = { (1/2)*(1+_P(1)) + (1/2)*(1-_P(1))*(ei2()^eo2()) }
Dilator = { CGA1_Dilator(_P(1))^CGA2_Dilator(_P(1)) }
// Translator(x,y,z) for translation by displacement vector (x,y,z)
CGA1_Translator = { 1 - (1/2)*(_P(1)*e1+_P(2)*e2+_P(3)*e3)*ei1() }
CGA2_Translator = { 1 - (1/2)*(_P(1)*e6+_P(2)*e7+_P(3)*e8)*ei2() }
Translator = {
 CGA1_Translator(_P(1),_P(2),_P(3))^CGA2_Translator(_P(1),_P(2),_P(3))
}

// Ellipsoid(px,py,pz,rx,ry,rz) with center (px,py,pz), radii rx ry rz
Ellipsoid = {
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*_P(1)*Tx()/rxSq + -2*_P(2)*Ty()/rySq + -2*_P(3)*Tz()/rzSq +
 Txx()/rxSq + Tyy()/rySq + Tzz()/rzSq +
 (pxSq/rxSq + pySq/rySq + pzSq/rzSq - 1)*T1()
}
// Toroid(R,r) with major radius R and minor radius r
Toroid = {
 R = _P(1); r = _P(2); dSq = R*R - r*r;
 Tt4() + 2*Tt2()*dSq + T1()*dSq*dSq - 4*R*R*(Txx()+Tyy())
}
// Sphere(x,y,z,r) with center point (x,y,z) and radius r
CGA1_Sphere = { CGA1_Point(_P(1),_P(2),_P(3)) - (1/2)*_P(4)*_P(4)*ei1() }
CGA2_Sphere = { CGA2_Point(_P(1),_P(2),_P(3)) - (1/2)*_P(4)*_P(4)*ei2() }
Sphere = {
 CGA1_Sphere(_P(1),_P(2),_P(3),_P(4))^CGA2_Sphere(_P(1),_P(2),_P(3),_P(4))
}
// Plane(nx,ny,nz,d) with normal (nx,ny,nz) at distance d from origin
CGA1_Plane = { Normalize(_P(1)*e1 + _P(2)*e2 + _P(3)*e3) + _P(4)*ei1() }
CGA2_Plane = { Normalize(_P(1)*e6 + _P(2)*e7 + _P(3)*e8) + _P(4)*ei2() }
Plane = {
 CGA1_Plane(_P(1),_P(2),_P(3),_P(4))^CGA2_Plane(_P(1),_P(2),_P(3),_P(4))
}
// Line(px,py,pz,dx,dy,dz) through (px,py,pz) in direction of (dx,dy,dz)
CGA1_Line = {
 d = Normalize(_P(4)*e1 + _P(5)*e2 + _P(6)*e3); D = E1D(d);
 p = _P(1)*e1 + _P(2)*e2 + _P(3)*e3; D - (p.D)*ei1()
}
CGA2_Line = {
 d = Normalize(_P(4)*e6 + _P(5)*e7 + _P(6)*e8); D = E2D(d);
 p = _P(1)*e6 + _P(2)*e7 + _P(3)*e8; D - (p.D)*ei2()
}
Line = {
 CGA1_Line(_P(1),_P(2),_P(3),_P(4),_P(5),_P(6))^
 CGA2_Line(_P(1),_P(2),_P(3),_P(4),_P(5),_P(6))
}

// Cylinder(px,py,pz,rx,ry,rz) with center (px,py,pz) and radii rx ry rz
CylinderX = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*py*Ty()/rySq + -2*pz*Tz()/rzSq + Tyy()/rySq + Tzz()/rzSq +
 (pySq/rySq + pzSq/rzSq - 1)*T1()
```

```
}
CylinderY = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq + -2*pz*Tz()/rzSq + Txx()/rxSq + Tzz()/rzSq +
 (pxSq/rxSq + pzSq/rzSq - 1)*T1()
}
CylinderZ = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq + -2*py*Ty()/rySq + Txx()/rxSq + Tyy()/rySq +
 (pxSq/rxSq + pySq/rySq - 1)*T1()
}

// Cone(px,py,pz,rx,ry,rz) with center (px,py,pz) and radii rx ry rz
ConeX = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 2*(px*Tx()/rxSq - py*Ty()/rySq - pz*Tz()/rzSq) -
 Txx()/rxSq + Tyy()/rySq + Tzz()/rzSq + (pySq/rySq + pzSq/rzSq - pxSq/rxSq)*T1()
}
ConeY = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 2*(py*Ty()/rySq - pz*Tz()/rzSq - px*Tx()/rxSq) +
 Txx()/rxSq - Tyy()/rySq + Tzz()/rzSq + (pxSq/rxSq - pySq/rySq + pzSq/rzSq)*T1()
}
ConeZ = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 2*(pz*Tz()/rzSq - py*Ty()/rySq - px*Tx()/rxSq) +
 Txx()/rxSq + Tyy()/rySq - Tzz()/rzSq + (pxSq/rxSq + pySq/rySq - pzSq/rzSq)*T1()
}

// Paraboloid(px,py,pz,rx,ry,rz) with vertex (px,py,pz) and radii rx ry rz
ParaboloidX = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*pz*Tz()/rzSq - 2*py*Ty()/rySq - Tx()/rx +
 Tzz()/rzSq + Tyy()/rySq + (pzSq/rzSq + pySq/rySq + px/rx)*T1()
}
ParaboloidY = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq - 2*pz*Tz()/rzSq - Ty()/ry +
 Txx()/rxSq + Tzz()/rzSq + (pxSq/rxSq + pzSq/rzSq + py/ry)*T1()
}
ParaboloidZ = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq - 2*py*Ty()/rySq - Tz()/rz +
 Txx()/rxSq + Tyy()/rySq + (pxSq/rxSq + pySq/rySq + pz/rz)*T1()
}

// Hyperbolic paraboloid (px,py,pz,rx,ry,rz) z-axis aligned
// with center point (px,py,pz) and radii rx ry rz
```

```
HParaboloidZ = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq + 2*py*Ty()/rySq - Tz()/rz +
 Txx()/rxSq - Tyy()/rySq + (pxSq/rxSq - pySq/rySq + pz/rz)*T1()
}

// Hyperboloid of one sheet (px,py,pz,rx,ry,rz) z-axis aligned
// with center point (px,py,pz) and radii rx ry rz
Hyperboloid1 = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 2*pz*Tz()/rzSq - 2*px*Tx()/rxSq - 2*py*Ty()/rySq +
 Txx()/rxSq + Tyy()/rySq - Tzz()/rzSq +
 (pxSq/rxSq + pySq/rySq - pzSq/rzSq - 1)*T1()
}
// Hyperboloid of two sheets (px,py,pz,rx,ry,rz) z-axis aligned
// with center point (px,py,pz) and radii rx ry rz
Hyperboloid2 = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 2*px*Tx()/rxSq + 2*py*Ty()/rySq - 2*pz*Tz()/rzSq -
 Txx()/rxSq - Tyy()/rySq + Tzz()/rzSq +
 (pzSq/rzSq - pxSq/rxSq - pySq/rySq - 1)*T1()
}

// Parabolic cylinders (px,py,pz,rx,ry,rz)
// with center point (px,py,pz) and radii rx ry rz
PCylinderX = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*py*Ty()/rySq - Tz()/rz + Tyy()/rySq + (pySq/rySq + pz/rz)*T1()
}
PCylinderY = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq - Tz()/rz + Txx()/rxSq + (pxSq/rxSq + pz/rz)*T1()
}
PCylinderZ = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq - Ty()/ry + Txx()/rxSq + (pxSq/rxSq + py/ry)*T1()
}

// Hyperbolic cylinders (px,py,pz,rx,ry,rz)
// with center point (px,py,pz) and radii rx ry rz
HCylinderX = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*py*Ty()/rySq + 2*pz*Tz()/rzSq + Tyy()/rySq - Tzz()/rzSq +
 (pySq/rySq - pzSq/rzSq - 1)*T1()
}
HCylinderY = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 2*px*Tx()/rxSq - 2*pz*Tz()/rzSq - Txx()/rxSq + Tzz()/rzSq +
```

```
 (-pxSq/rxSq + pzSq/rzSq - 1)*T1()
}
HCylinderZ = {
 px = _P(1); py = _P(2); pz = _P(3); rx = _P(4); ry = _P(5); rz = _P(6);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); pzSq = _P(3)*_P(3);
 rxSq = _P(4)*_P(4); rySq = _P(5)*_P(5); rzSq = _P(6)*_P(6);
 -2*px*Tx()/rxSq + 2*py*Ty()/rySq + Txx()/rxSq - Tyy()/rySq +
 (pxSq/rxSq - pySq/rySq - 1)*T1()
}

// Parallel planes pair (p1,p2) with planes a=p1, a=p2 perpendicular to a-axis
PPlanesX = { p1 = _P(1); p2 = _P(2); Txx() - (p1+p2)*Tx() + p1*p2*T1() }
PPlanesY = { p1 = _P(1); p2 = _P(2); Tyy() - (p1+p2)*Ty() + p1*p2*T1() }
PPlanesZ = { p1 = _P(1); p2 = _P(2); Tzz() - (p1+p2)*Tz() + p1*p2*T1() }
// Non-parallel planes pairs (also types of cylinders)
// XPlanesX(y,z,ry,rz) x-axis aligned, with vertex (y,z) and slope (+|-)rz/ry
XPlanesX = {
 py = _P(1); pz = _P(2); ry = _P(3); rz = _P(4);
 pySq = _P(1)*_P(1); pzSq = _P(2)*_P(2); rySq = _P(3)*_P(3); rzSq = _P(4)*_P(4);
 -2*py*Ty()/rySq + 2*pz*Tz()/rzSq + Tyy()/rySq - Tzz()/rzSq +
 (pySq/rySq - pzSq/rzSq)*T1()
}
// XPlanesY(x,z,rx,rz) y-axis aligned, with vertex (x,z) and slope (+|-)rz/rx
XPlanesY = {
 px = _P(1); pz = _P(2); rx = _P(3); rz = _P(4);
 pxSq = _P(1)*_P(1); pzSq = _P(2)*_P(2); rxSq = _P(3)*_P(3); rzSq = _P(4)*_P(4);
 -2*pz*Tz()/rzSq + 2*px*Tx()/rxSq + Tzz()/rzSq - Txx()/rxSq +
 (pzSq/rzSq - pxSq/rxSq)*T1()
}
// XPlanesZ(x,y,rx,ry) z-axis aligned, with vertex (x,y) and slope (+|-)ry/rx
XPlanesZ = {
 px = _P(1); py = _P(2); rx = _P(3); ry = _P(4);
 pxSq = _P(1)*_P(1); pySq = _P(2)*_P(2); rxSq = _P(3)*_P(3); rySq = _P(4)*_P(4);
 -2*px*Tx()/rxSq + 2*py*Ty()/rySq + Txx()/rxSq - Tyy()/rySq +
 (pxSq/rxSq - pySq/rySq)*T1()
}

// DupinCyclide(R,r1,r2) with major radius R and minor radii r1 and r2
DupinCyclide = {
 u = (1/2)*(_P(2) + _P(3)); c = (1/2)*(_P(2) - _P(3)); a = _P(1); bSq = a*a-c*c;
 Tt4() + 2*Tt2()*(bSq-u*u) - 4*a*a*Txx() - 4*bSq*Tyy() +
 8*a*c*u*Tx() + ((bSq-u*u)*(bSq-u*u) - 4*c*c*u*u)*T1()
}
hornedDupinCyclide = {
 u = (1/2)*(_P(2) + _P(3)); c = (1/2)*(_P(2) - _P(3)); a = _P(1); bSq = a*a-u*u;
 Tt4() + 2*Tt2()*(bSq-c*c) - 4*a*a*Txx() - 4*bSq*Tyy() +
 8*a*c*u*Tx() + ((bSq-c*c)*(bSq-c*c) - 4*c*c*u*u)*T1()
}
```

7.3 products.csv

To finish the definition of the DCGA algebra, one more file `algebras/DCGA/products.csv` must be generated using the `ProductTableCreator`, which is an extra program that is available for download at the Gaalop homepage. When this program is run, it will ask for the definition folder `algebras/DCGA` of the algebra and then it generates `products.csv`.

7.4 Gaalop configuration

When Gaalop is run, it has a Configure button to access the configuration tabs of Gaalop.

In the Algebra tab, enter the full path to our `algebras` folder into the box titled additionalBaseDirectory.

It may be necessary to download the latest version of the Lightweight Java Game Library (LWJGL) for use with the Gaalop Visualizer plugin. The latest LWJGL package in the version 2.x series may work. Download and decompress the LWJGL package to somewhere on your computer, then access the Visualizer configuration tab and set the lwjglNativePath to the appropriate folder within the LWJGL installation folder. The Gaalop `plugins` folder may need updated copies of the files `lwjgl.jar` and `lwjgl_util.jar` that can be copied from within the LWJGL installation.

If all other dependencies required for the Visualizer plugin are installed, then it should be possible to proceed with testing the DCGA algebra in Gaalop, and generating Visualizer renderings of the DCGA entities.

7.5 Visualizing DCGA entities

In the preceding subsections, Gaalop was configured for the DCGA algebra, and the Visualizer plugin was configured to use LWJGL for rendering. Assuming that the configuration is successful, the next step is to create your first CLUScript for developing your first Geometric Algebra Algorithms or computations in DCGA using Gaalop.

Start Gaalop, then press the New File button to create a new CLUScript file with any name you like. Select Own - DCGA in the Algebra to use selection box. Select Visual Code Inserter in the VisualCodeInserter selection box. Select Table-Based Approach in the Optimization selection box. Select Visualizer in the CodeGenerator selection box. Now, to get started, the following CLUScript demonstrates how to use the macros that were defined in `macros.clu`.

```
/* Demo of the DCGA algebra
 * Uncomment the lines to be visualized and press Optimize button.
 * Visualizing one line at a time is recommended on slow computers.  */
// Set the drawing color, and display a variable circular toroid
//:White;:T = Toroid(R,r);                    /* variable inputs (R,r) */
//:P1 = createPoint(1,2,3);                           /* or EV(x,y,z) */
//:Red    ;:S = Sphere(1,2,3,2);                 /* (px,py,pz,radius) */
//:Green  ;:P = Plane(1,2,3,sqrt(1+4+9));      /* (nx,ny,nz,distance) */
//:Blue   ;:L = Line(1,2,3,0,0,1);             /* (px,py,pz,dx,dy,dz) */
//:Cyan   ;:C = S^P;                                       /* circle */
//:Magenta;:E = Ellipsoid(1,2,3,4,3,2);        /* (px,py,pz,rx,ry,rz) */
//:Yellow ;:e = E^P/16; /* ellipse (divide by 16 to help find zeros) */

// Geometric outer product null space (GOPNS) bi-CGA entities
// (Visualizer required IPNS entities, so Dual * is taken)
// Sphere, the wedge of four surface points
//:SD = *( EV(-5,0,0)^EV(5,0,0)^EV(0,5,0)^EV(0,0,5) );
// Plane, the wedge of three plane points (not collinear) and infinity
//:PD = *( EV(-5,0,1)^EV(5,0,1)^EV(0,5,1)^ei() );
// Line, the wedge of two line points and infinity
```

```
//:LD = *( EV(-5,0,1)^EV(5,0,1)^ei() );
// Circle, the wedge of three circle points
//:CD = *( EV(-5,0,1)^EV(5,0,1)^EV(0,5,1) );

// Elliptic cylinders (px,py,pz,rx,ry,rz)
//:CylX = CylinderX(0,1,0,1,2,4);
//:CylY = CylinderY(1,0,0,2,1,4);
//:CylZ = CylinderZ(1,0,0,4,2,1);
// Cones  (px,py,pz,rx,ry,rz)
//:CneX = ConeX(1,0,0,1,1,1);
//:CneY = ConeY(0,1,0,1,1,1);
//:CneZ = ConeZ(0,0,1,1,1,1);
// Paraboloids (px,py,pz,rx,ry,rz)
//:ParX = ParaboloidX(1,0,0,1,1,1);
//:ParY = ParaboloidY(0,1,0,1,1,1);
//:ParZ = ParaboloidZ(0,0,1,1,1,1);
// Hyperbolic paraboloid z-axis aligned (px,py,pz,rx,ry,rz)
//:HparZ = HParaboloidZ(0,0,1,1,1,1);
// Hyperboloid of one sheet z-axis aligned (px,py,pz,rx,ry,rz)
//:Hyp1 = Hyperboloid1(0,0,1,1,1,1);
// Hyperboloid of two sheets z-axis aligned (px,py,pz,rx,ry,rz)
//:Hyp2 = Hyperboloid2(0,0,1,1,1,1);
// Parabolic cylinders (px,py,pz,rx,ry,rz)
//:PCylX = PCylinderX(0,0,1,1,2,4);
//:PCylY = PCylinderY(0,0,1,2,1,4);
//:PCylZ = PCylinderZ(1,0,0,2,4,1);
// Hyperbolic cylinders (px,py,pz,rx,ry,rz)
//:HCylX = HCylinderX(0,0,1,1,1,1);
//:HCylY = HCylinderY(0,0,1,1,1,1);
//:HCylZ = HCylinderZ(1,0,0,1,1,1);
// Parallel planes pair (p1,p2)
//:PPX = PPlanesX(1,2);
//:PPY = PPlanesY(1,4);
//:PPZ = PPlanesZ(1,6);
// Non-parallel planes pair
//:XPX = XPlanesX(1,1,2,1); /* (y,z,ry,rz) */
//:XPY = XPlanesY(1,1,2,1); /* (x,z,rx,rz) */
//:XPZ = XPlanesZ(1,1,2,1); /* (x,y,rx,ry) */
// Ellipse, parabola, hyperbola
//:e = CylinderZ(0,0,0,1,4,1)^Plane(0,0,1,1);
//:p = PCylinderZ(0,0,0,1,4,1)^Plane(0,0,1,1);
//:h = HCylinderZ(0,0,0,1,4,1)^Plane(0,0,1,1);

// Rotate, dilate, and translate an ellipsoid
//:ER = Rotor(0,0,1,45) * Ellipsoid(0,0,0,1,3,1) * ~Rotor(0,0,1,45);
//:ED = Dilator(2) * Ellipsoid(0,0,0,1,2,3) * ~Dilator(2);
//:ET = Translator(1,2,3) * Ellipsoid(0,0,0,2,3,4) * ~Translator(1,2,3);

// DupinCyclide and hornedDupinCyclide(R,r1,r1), also translated
//:DC = Translator(1,1,1) * DupinCyclide(3,2,1) * ~Translator(1,1,1);
//:HDC = Translator(1,1,1) * hornedDupinCyclide(3,2,1) * ~Translator(1,1,1);

// Parabolic cyclide as inversion of Toroid in Sphere centered on toroid surface
//:PC = Sphere(2,0,0,3) * Toroid(5,3) * ~Sphere(2,0,0,3);
// PC rotated 60deg around the z-axis (this computation can take a minute)
//:RPC = Rotor(0,0,1,60) * PC * ~Rotor(0,0,1,60);

// Intersection of DupinCyclide and a Plane
//:DC = DupinCyclide(3,2,1); /* try (R,r1,r2) for variable inputs R,r1,r2 */
//:PL = Plane(2,1,0,2);
//:DCPL = DC^PL; /* try large epsilon=4 in Gradient Method to see DCPL points */
```

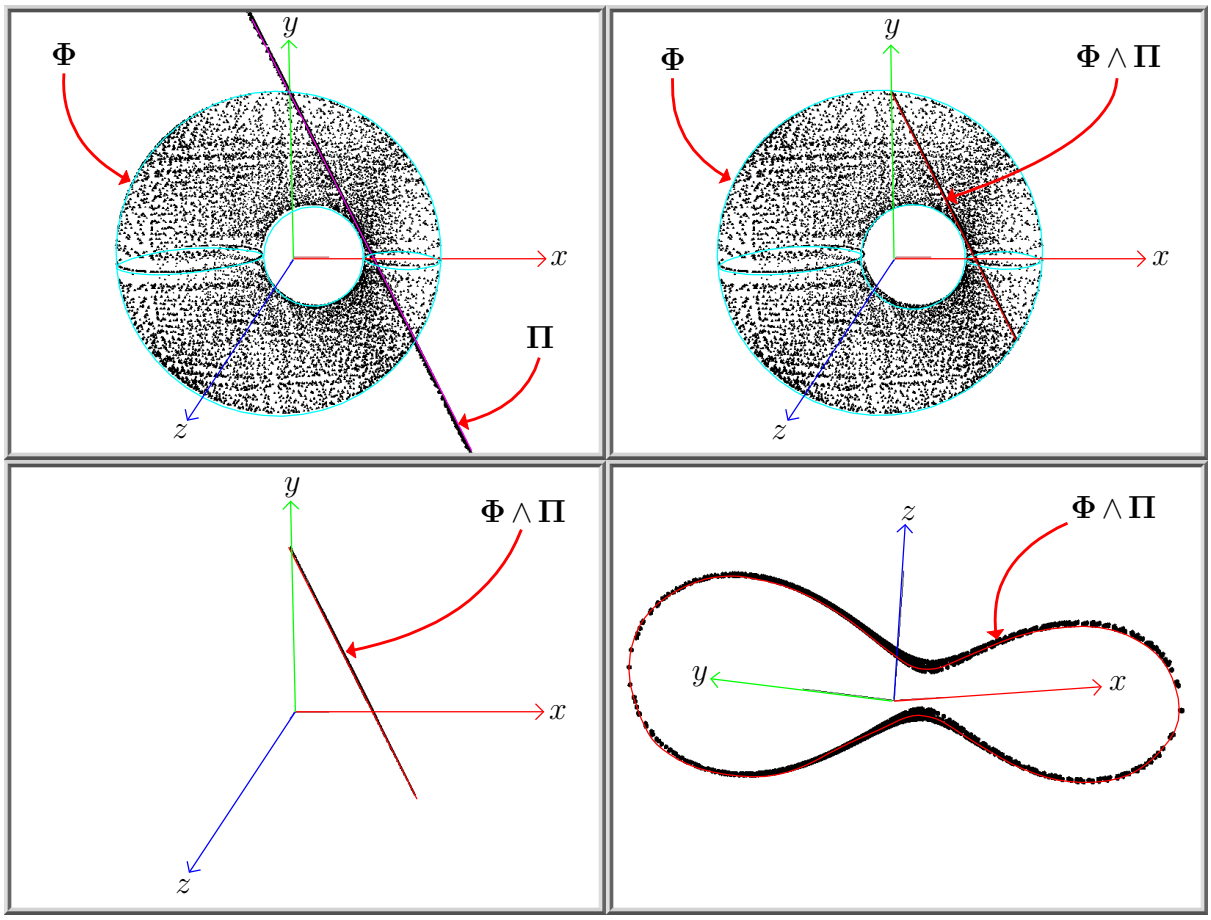

Figure 33. Intersection $\mathbf{\Phi} \wedge \mathbf{\Pi}$ of ring Dupin cyclide $\mathbf{\Phi}$ and plane $\mathbf{\Pi}$

Figure 33 shows the Gaalop Visualizer plugin renderings of a DCGA ring Dupin cyclide $\mathbf{\Phi}$, a standard DCGA plane $\mathbf{\Pi}$, and their intersection entity $\mathbf{\Phi} \wedge \mathbf{\Pi}$. All three entities are rendered in the upper-left image. In the upper-right image, the plane rendering is disabled, showing only the cyclide and the intersection entity, which is seen only on its edge. In the lower-left image, the cyclide rendering is disabled, showing only the intersection entity, which is still seen on its edge. The lower-right image is a rotated view of the intersection entity, which clearly shows that it is a curve that cuts the cyclide exactly where the plane cut it.

8 Conclusion

The $\mathcal{G}_{8,2}$ *Double Conformal / Darboux Cyclide Geometric Algebra* (DCGA) that has been presented, and possibly introduced for the first time, in this paper may be an interesting algebra for future research or for some applications now.

DCGA provides entities for all the surfaces available in CGA, and DCGA also has entities for all quadric surfaces. DCGA also provides a toroid entity which may be a new entity not previously available in $\mathcal{G}_{4,1}$ CGA nor in $\mathcal{G}_{6,3}$ QGA. More generally, DCGA has entities for cyclide surfaces. DCGA has a complete set of entity transformation operations as versor operations that can transform both the GIPNS and GOPNS forms of all entities, including the toroid and cyclide entities. The available versors are rotor, dilator, translator, and motor. DCGA supports the creation of GIPNS intersection entities as the

wedge of intersecting GIPNS entities, but this support is limited to intersecting up to a single quadric, toroid, or cyclide surface entity with some combination of spheres, planes, lines, and circles not exceeding a combined grade of 8.

Although not yet tested by this author, the possible extension of $\mathcal{G}_{8,2}$ DCGA to a $\mathcal{G}_{12,3}$ *Triple* or $\mathcal{G}_{16,4}$ *Quadruple Conformal Geometric Algebra* may be theoretically feasible, and may allow for general cubic and quartic surface entities.

References

[1] Alan Bromborsky. *Geometric Algebra Module for Sympy*. 2015.

[2] Patrick Charrier. Geometric Algebra enhanced Precompiler for C++ and OpenCL. Master's thesis, Technische Universität Darmstadt, Germany, 2012.

[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. *Quaternions and Clifford Geometric Algebras*. ViXra.org, 2015.

[5] Sebti Foufou and Lionel Garnier. Dupin cyclide blends between quadric surfaces for shape modeling. *Comput. Graph. Forum*, 23(3):321–330, 2004.

[6] Dietmar Hildenbrand. Geometric Algebra: A Foundation of Elementary Geometry with possible Applications in Computer Algebra based Dynamic Geometry Systems. *The Electronic Journal of Mathematics and Technology*, 1(1):19, 2015.

[7] David Hestenes. *New Foundations for Classical Mechanics*, volume 99 of *Fundamental Theories of Physics*. Dordrecht: Kluwer Academic Publishers, Second edition, 1999.

[8] 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.

[9] Dietmar Hildenbrand. *Foundations of Geometric Algebra Computing*. Berlin: Springer, 2013.

[10] Daniel Klawitter. *Clifford Algebras, Geometric Modelling and Chain Geometries with Application in Kinematics*. Springer Spektrum, 2015.

[11] Helmut Pottmann, Ling Shi and Mikhail Skopenkov. Darboux cyclides and webs from circles. *Comput. Aided Geom. Design*, 29(1):77–97, 2012.

[12] Christian Perwass. *Geometric Algebra with Applications in Engineering*, volume 4 of *Geometry and Computing*. Springer, 2009.

[13] P. Ramachandran and G. Varoquaux. Mayavi: 3D Visualization of Scientific Data. *Computing in Science & Engineering*, 13(2):40–51, 2011.

[14] Bodo Rosenhahn. *Pose Estimation Revisited*. Christian-Albrechts-Universität zu Kiel, 2003.

[15] Michael Schrott and Boris Odehnal. Ortho-circles of dupin cyclides. *Journal for Geometry and Graphics*, 10(1):73–98, 2006.

[16] Christian Steinmetz. Examination of new geometric algebras including a visualization and integration in a geometric algebra compiler. Master's thesis, Technische Universität Darmstadt, Germany, 2013.

[17] SymPy Development Team. *SymPy: Python library for symbolic mathematics*. 2015.

[18] Gerald Sommer, editor. *Geometric Computing with Clifford Algebras, Theoretical Foundations and Applications in Computer Vision and Robotics*. Berlin: Springer, 2001.

[19] Julio Zamora-Esquivel. G6,3 Geometric Algebra; Description and Implementation. *Advances in Applied Clifford Algebras*, 24(2):493–514, 2014.