Chapter 2
Geometry of the Four-Dimensional Space

This chapter will focus on three important topics relative to 4D space: (section 2.1) Basic definitions, (section 2.2) revision of four families of polytopes and (section 2.3) geometric transformations. Although it is suggested by the title, this chapter doesn't concentrate exclusively on 4D space's aspects, because these same aspects will serve as support for presenting some formulations which are applicable to n-Dimensional Space.

2.1 Definitions

2.1.1 Polyhedra

A **polyhedron** is a bounded subset of the 3D Euclidean Space enclosed by a finite set of plane polygons such that every edge of a polygon is shared by exactly one other polygon (adjacent polygons) [Preparata, 85]. Additionally, it can be established that the polygons that are incident to a vertex must compose a single circuit [Coxeter, 63].

[Coxeter, 63] established that the polyhedron's boundary is a simple and closed surface of a volume (therefore, "dangling" faces and edges are not accepted). When the volume's surface (i.e. the polyhedron's boundary) is considered without aspects like areas, distances or angles but only taking the surface's aspects not affected by deformation, then we will work with the **surface's topology** [Weeks, 02]. In this context, the whole surface is called a two-dimensional manifold or a 2-manifold.

A 2-manifold has either local and global properties. Local properties are those observable inside a manifold's small region, while global properties require to consider the manifold entirely [Weeks, 02]. For example, a 2-manifold defines a 2D space with a local topology for a plane, however its global topology can correspond to a sphere's surface or a torus' surface (or any other).

The polyhedron's surface must fulfill two characteristics cited by [Coxeter, 63]: to be simple and closed. The surface is closed, or in other words, the manifold is closed when it decomposes the space where it is embedded (3D space) in two regions, one of them called the interior, is finite [Coxeter, 63]. The surface is simple and closed, or the manifold is oriented and closed, when it is possible to distinguish both its opposite sides in the 3D space, that is to say, it is clearly possible to distinguish between the interior and the outside (Klein's bottle is an example of a closed not oriented manifold) [Hansen, 93].

Edges and vertices, as boundary elements for polyhedra, are classified as 2-Manifold (or just Manifold) elements. A Manifold edge is adjacent to exactly two faces, and a Manifold vertex is the apex (i.e., the common vertex) of only one **cone of faces** (the faces compose a single circuit) [Rossignac, 91].

2.1.2 Pseudo-Polyhedra

A **pseudo-polyhedron** is a bounded subset of the 3D Euclidean Space enclosed by a finite collection of planar faces such that every edge has at least two adjacent faces, and if any two faces meet, they meet at a common edge [Tang, 91]. From this definition we have that polyhedra are a special case (a subset) of pseudo-polyhedra when exactly two faces are incident to all their edges. The boundary's Pseudo-Polyhedra also must fulfill to be a closed surface ("dangling" faces and edges are not accepted).

From the topological's point of view, only some regions of the pseudo-polyhedron's surface are 2-manifold. This is because, contrary to polyhedra, the pseudo-polyhedra's interior is composed by more than two regions. An interior region can be seen as limited by a surface. In a pseudo-polyhedron, at least two interior regions' surfaces have common points, which compose the regions, from the surface seen globally (the pseudo-polyhedron's whole surface) that are not 2-manifold, or in other words, these regions are non-manifold. For example, see the pseudo-polyhedron presented in **Figure 2.1.a**. That pseudo-polyhedron can be seen as two cubes sharing a vertex (**Figure 2.1.b**). The boundary of each cube is topologically equivalent to a sphere's surface (**Figure 2.1.c**). Finally, both surfaces will have a common point, that is to say, the shared vertex (**Figure 2.1.d**). All the surface's regions are 2-manifold, except the point that correspond to the shared vertex, which is non-manifold.

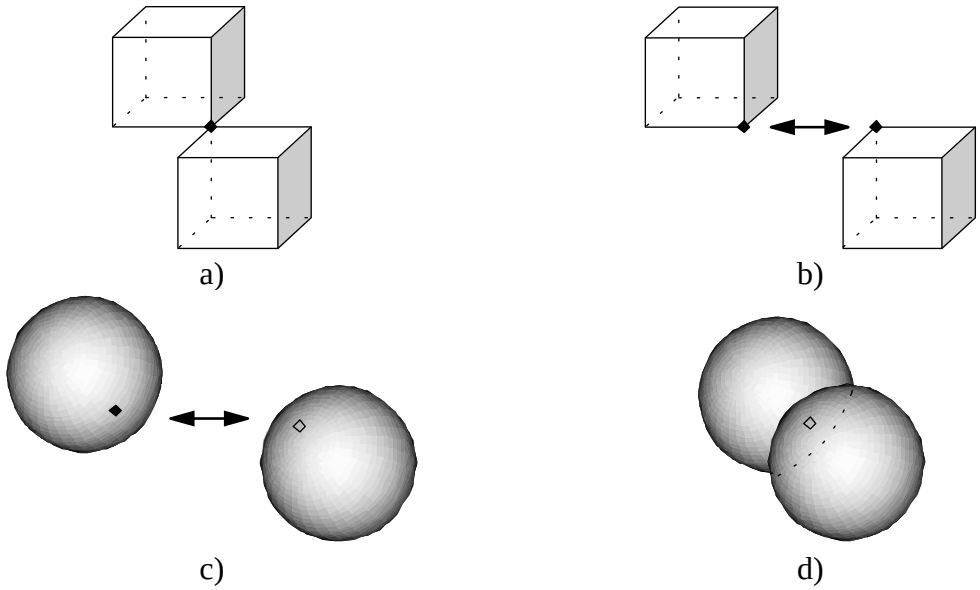


FIGURE 2.1
A pseudo-polyhedron (a) and its topological considerations (b, c & d).
See text for details (Own elaboration).

Edges and vertices, as boundary elements for pseudo-polyhedra, may be either two-manifold (or just manifold) or non-manifold elements. In the case of edges, they are (non) manifold elements when every points of it is also a (non) manifold point, except that either or both of its ending vertices might be a point of the opposite type [Aguilera, 98]. A manifold edge is adjacent to exactly two faces, and a manifold vertex is the apex (i.e., the common vertex) of only one **cone of faces**. Conversely, a non-manifold edge is adjacent to more than two faces, and a non-manifold vertex is the apex (i.e., the common vertex) of more than one **cone of faces** [Rossignac, 91].

2.1.3 Four-Dimensional Polytopes

We will extend the polyhedron's definition presented by [Preparata, 85] (see section 2.1.1) for defining the 4D Polytopes: A 4D Polytope is a closed subset from the 4D

Euclidean Space, which is delimited by finite set of three-dimensional cells or volumes (polyhedra) such that every volumes' face is shared only with another volume (adjacent volumes). In the same way, it can be established that a 4D Polytope's boundary will be a simple and closed hyper-surface of a hyper-volume, therefore, "dangling" volumes, faces and edges are not accepted (it has been extended, in analogous way, one of the most important properties that must fulfill a polyhedron, which has been cited by [Coxeter, 63] and also mentioned in section 2.1.1).

[Weeks, 02] points out that the definitions related to the 2-manifolds' topology (surfaces) can be extended for defining the 3-manifolds' topology. [Weeks, 02] defines a 2-manifold as a space with a plane's local topology on its boundary, and analogously, defines a 3-manifold as a space with our "ordinary" three-dimensional space's local topology on its boundary.

We can appeal to *Flatland* for a better understanding. *Flatland* is a two-dimensional universe, therefore, it is a surface or a 2-manifold, which is inhabited by polygonal beings. *A.Square*'s interaction with his universe will allow him to determine that *Flatland* is a plane, however, this conclusion is topologically valid from a local point of view. *Flatland*'s global topology could belong to a sphere's surface (as described in *Sphereland* [Burger,83]) or a torus' surface, for example. Our three-dimensional universe, which we inhabit, can be seen as *Spaceland* [Abbott, 84]. However, since Einstein, our universe is seen as a hyper-surface, or better, a 3-manifold [Sagan, 80]. Our universe can be the hyper-sphere's boundary (as Einstein believed), or a 4D torus' boundary. Because we are embedded in a 3-manifold, our universe, we can not perceive the effects by its curvature, or in other

words, by its global topology. Since the 4D Polytopes' boundary is composed by three-dimensional cells, its topology will be related with a 3-manifold.

Faces, edges and vertices, as boundary elements for 4D polytopes, will be manifold. [Hansen, 93] has established that a manifold face is adjacent to exactly two volumes, and [Pérez & Aguilera, 03] have suggested that a manifold edge is the common edge of only one **hyper-cone of volumes** (the faces and edges' characterizations will be analyzed with more detail in chapter 4).

2.1.4 Four-Dimensional Pseudo-Polytopes

We will extend the pseudo-polyhedron's definition presented by [Tang, 91] for defining the 4D Pseudo-Polytopes: A **4D Pseudo-Polytope** is a bounded subset of the 4D Euclidean Space enclosed by a finite collection of volumes such that every face has <u>at least</u> two adjacent volumes, and if any two volumes meet, they meet at a common face. From this definition we have that 4D Polytopes are a special case (a subset) of Pseudo-Polytopes when <u>exactly</u> two volumes are incident to all their faces. The boundary's Pseudo-Polytopes also must fulfill to be a hyper-volume's closed hyper-surface ("dangling" volumes, faces and edges are not accepted).

Basically, the topological differences between 4D Polytopes and Pseudo-Polytopes are analogous to the Polyhedra and Pseudo-Polyhedra's case. Certain regions from the 4D Pseudo-Polytopes' boundary can be considered as not belong exclusively to just one 3-manifold, because, as analogously to Pseudo-Polyhedra, these regions (faces, edges or

vertices) can be seen as shared by several hyper-surfaces. Therefore, faces, edges and vertices, as boundary elements for 4D polytopes, may be either manifold or non-manifold elements. We have stated that a manifold face is adjacent to exactly two volumes, and a manifold edge is the common edge of only one **hyper-cone of volumes**. [Pérez & Aguilera, 03] have proposed that a non-manifold face is adjacent to more than two volumes, and a non-manifold edge is the common edge of more than one **hyper-cone of volumes** (chapter 4 will describe the methodologies that lead to these characterizations).

2.1.5 The n-Dimensional Polytopes

[Coxeter, 63] defines an Euclidean polytope Π_n as a finite region of n-dimensional Euclidean space enclosed by a finite number of (n-1) dimensional hyperplanes. The finiteness of the region implies that the number N_{n-1} of bounding hyperplanes satisfies the inequality $N_{n-1}>n$. The part of the polytope that lies on one of these hyperplanes is called a cell. Each cell of a Π_n is an (n-1)-dimensional polytope, Π_{n-1}. The cells of a Π_{n-1} are Π_{n-2}'s, and so on; we thus obtain a descending sequence of elements Π_{n-3}, Π_{n-4}, ... , Π_1 (an edge), Π_0 (a vertex).

The way that the cells Π_{n-1}, Π_{n-2}, Π_{n-3}, Π_{n-4}, ... , Π_1, Π_0 are related is given by the following [Sommerville, 58]'s observations:

- The Π_{n-1}'s share Π_{n-2}'s, in that way, it is defined a Polytope Π_n when two and only two Π_{n-1}'s share a Π_{n-2}; when more that two Π_{n-1}'s share a Π_{n-2} then it is defined a Pseudo-Polytope Π_n (the notation Π_n is common to polytopes and pseudo-polytopes).
- Three or more Π_{n-1}'s will have a common Π_{n-3}.
- *p* or more Π_{n-1}'s will have a common Π_{n-p}.
- n or more Π_{n-1}'s will have a common Π_0 (a vertex).

We know that a Π_3 (a 3D Euclidean polytope) is a polyhedron. The polyhedron's cells are Π_2. A Π_2 (a 2D Euclidean polytope) is a polygon. The polygon's cells are Π_1. A Π_1 (a 1D Euclidean polytope) is a segment. Finally, the segment's cells are Π_0, a set of vertices. The cells of a Π_4 (a 4D Euclidean polytope) are Π_3 (polyhedra, also called volumes in the context of Π_4).

From the topological's point of view, n-dimensional Polytopes are considered by [Hansen, 93] as a closed set of n-manifolds, one for each cell Π_{n-1}. In the previous sections, the 3D and 4D Polytopes' boundary was entirely considered as a manifold. For example, a cube's boundary is topologically equivalent to a sphere, but from [Hansen, 93]'s point of view, each face of the cube will be topologically equivalent to a plane, or in other words, each one will be a 2-manifold. Furthermore, [Hansen, 93] states that each element on a cell Π_{n-1}'s boundary will have its respective topologic equivalence. In this way, the edges of a cube's face will be topologically equivalent to a 1-manifold (a space with the local topology of a line [Weeks, 02]) and so forth. By representing the nD Polytopes by this way, [Hansen, 93] presents the following properties (which will be reconsidered in chapter 4):

1. A 0-manifold is a point, and it has no boundary.
2. All boundary elements of an n-manifold are (n-1)-manifold elements.
3. All (n-1)-dimensional elements belong to exactly two n-manifold elements (or twice to the same element).
4. Manifold elements may not intersect each other except at common boundary elements.

2.2 Some Polytopes' Families

In the following sections, three of the main families of polytopes will be described. These polytopes' families exist in all hyperdimensional spaces [Aichholzer, 00]: the *parallelotopes*, the *simplexes* and the *cross polytopes* (sections 2.2.1, 2.2.2 and 2.2.3 respectively). Furthermore, it will be described the *0/1-Polytopes*, which are closely related to the *parellelotopes* (section 2.2.4).

2.2.1 The Hypercube

2.2.1.1 Obtaining a Segment, a Square, a Cube and a Hypercube

[Rucker, 77] presents Claude Bragdon's method to define a series of figures which are called the ***parallelotopes*** [Coxeter, 63] or the ***orthotopes*** [Sommerville, 58]. First a 0D point is taken and moved one unit to the right. The path between the first and the second new point produces a 1D segment. The first dimension, represented by the X_1-axis (X), has appeared (**Figure 2.2**).

FIGURE 2.2
Generation and final 1D unit segment C_1 (own elaboration).

The new segment is then moved one unit upward. The path between the first and the second new segment produces a 2D square (a parallelogram). The second dimension, represented by the X_2-axis (Y), has appeared (**Figure 2.3**).

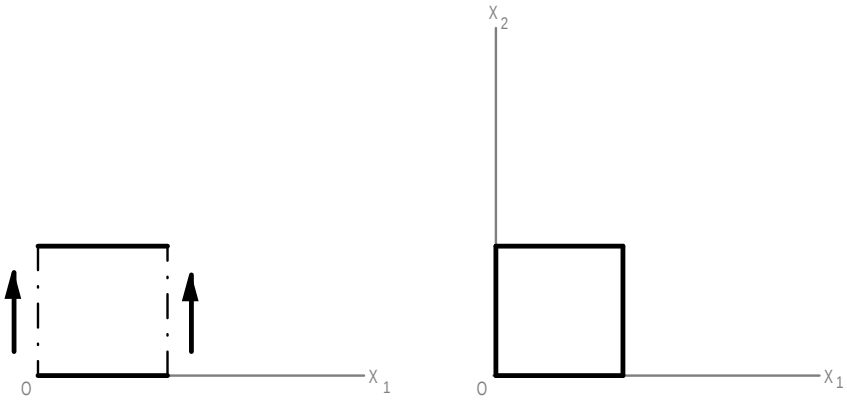


FIGURE 2.3
Generation and final 2D unit square C_2 (own elaboration).

The new square is then moved one unit forward out this paper. The path between the first and the second new square produces a 3D cube (a parallelepiped). The third dimension, represented by the X_3-axis (Z), has appeared (**Figure 2.4**). Because we are working on a 2D surface (this paper or the computer's screen), a diagonal between X_1 (X) and X_2-axis (Y) represents the X_3-axis (Z), *however it should be interpreted as a line perpendicular to this 2D surface*.

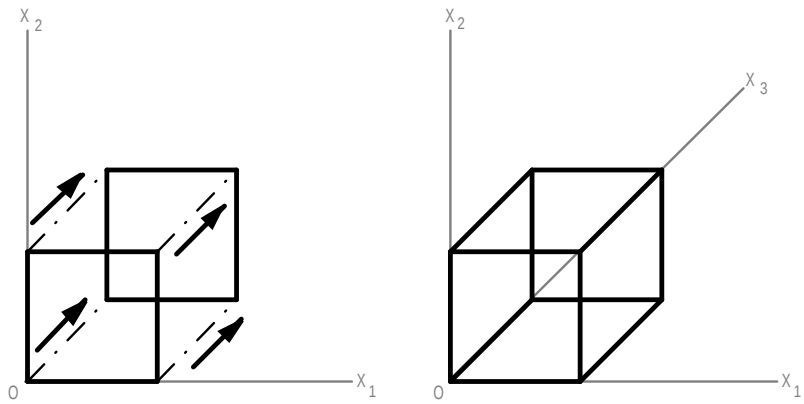


FIGURE 2.4
Generation and final 3D unit cube C_3 (own elaboration).

We know that the fourth dimension has a direction perpendicular to the other three dimensions, in this case the X_4-axis (W) is presented as a perpendicular line to the W_3-axis (Z). Then the cube is moved one unit in direction of the X_4-axis (W). The path (six cubes perpendicular to the first one) between the first and the second new cube produces the 3D boundary of a 4D hypercube (a 4D parallelotope). The fourth dimension has appeared (**Figure 2.5**).

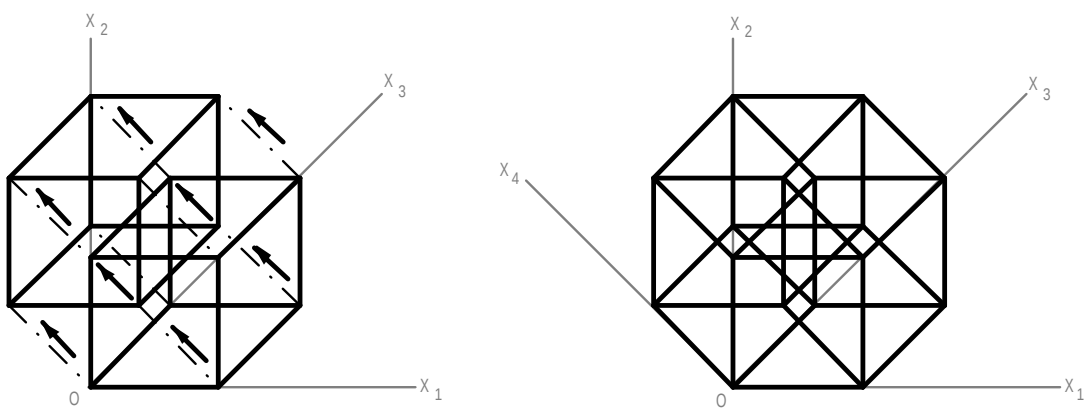
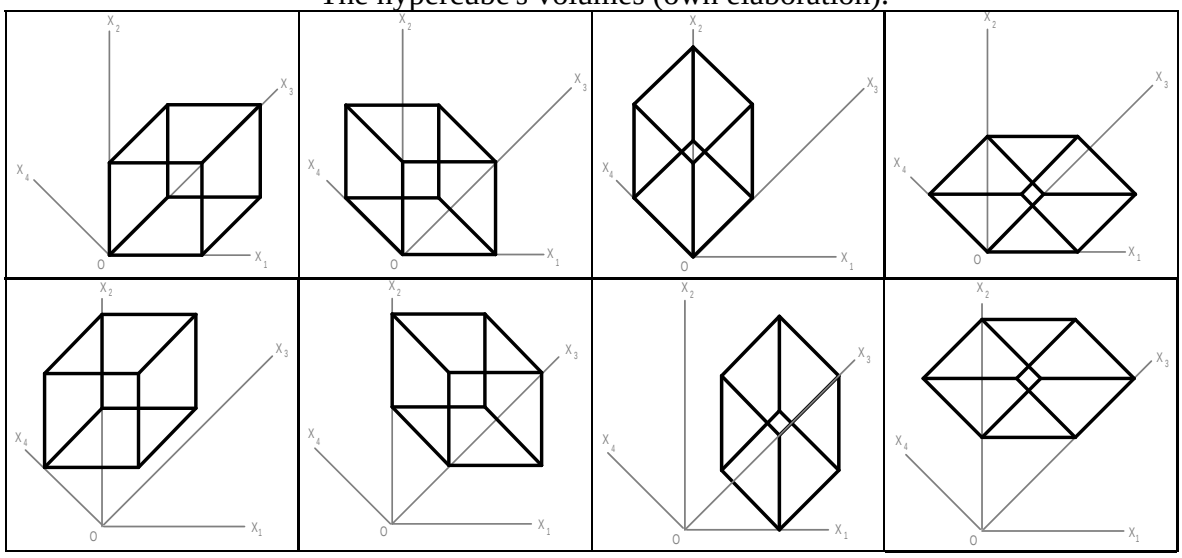


FIGURE 2.5
Generation and final 4D unit hypercube C_4 (own elaboration).

Definition 2.1: *Let* $\mathbf{C_n}$ *be the n-dimensional parallelotope, then* C_0 *is a point and* **Figures 2.2 to 2.5** *correspond to* C_1 *to* C_4.

TABLE 2.1

The hypercube's volumes (own elaboration).



2.2.1.2 The 4D Hypercube Properties

The analysis of the hypercube is also interesting because it can be done using the analogy with the 3D cube and the method presented in section 2.2.1.1. [Hilbert, 52] identified that the boundary of a hypercube is composed by eight three-dimensional regions called cubes, volumes or cells (**Table 2.1**), and call the hypercube an 8-cell polytope. To better illustrate this, let's see its analogy with its 3D counterpart. The 3D cube's boundary faces can be grouped into three pairs of parallel faces, where their supporting planes define two 2D-spaces parallel to each other. Each pair can be obtained by ignoring all those edges parallel to main axes X_1, X_2 and X_3 [Aguilera, 02c], see **Figure 2.6**.

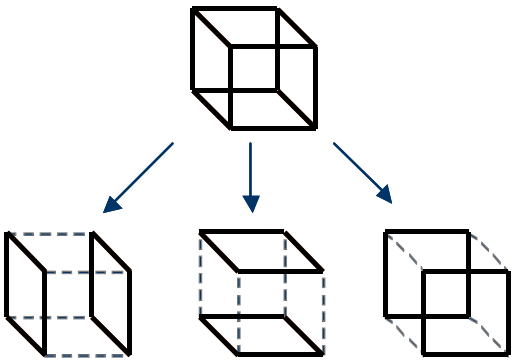


FIGURE 2.6
Viewing the cube's boundary faces (taken from [Aguilera & Pérez, 02]).

Similarly, and as shown in **Figure 2.7**, all the hypercube's boundary volumes can be grouped into four pairs of parallel cubes, furthermore, their supporting hyper-planes define two 3D-spaces parallel to each other.

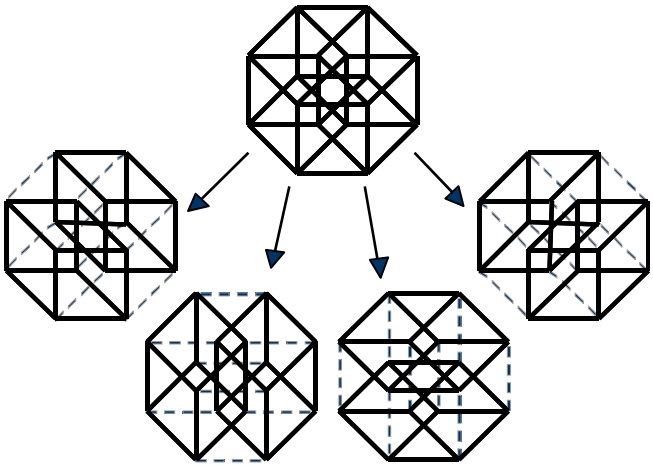
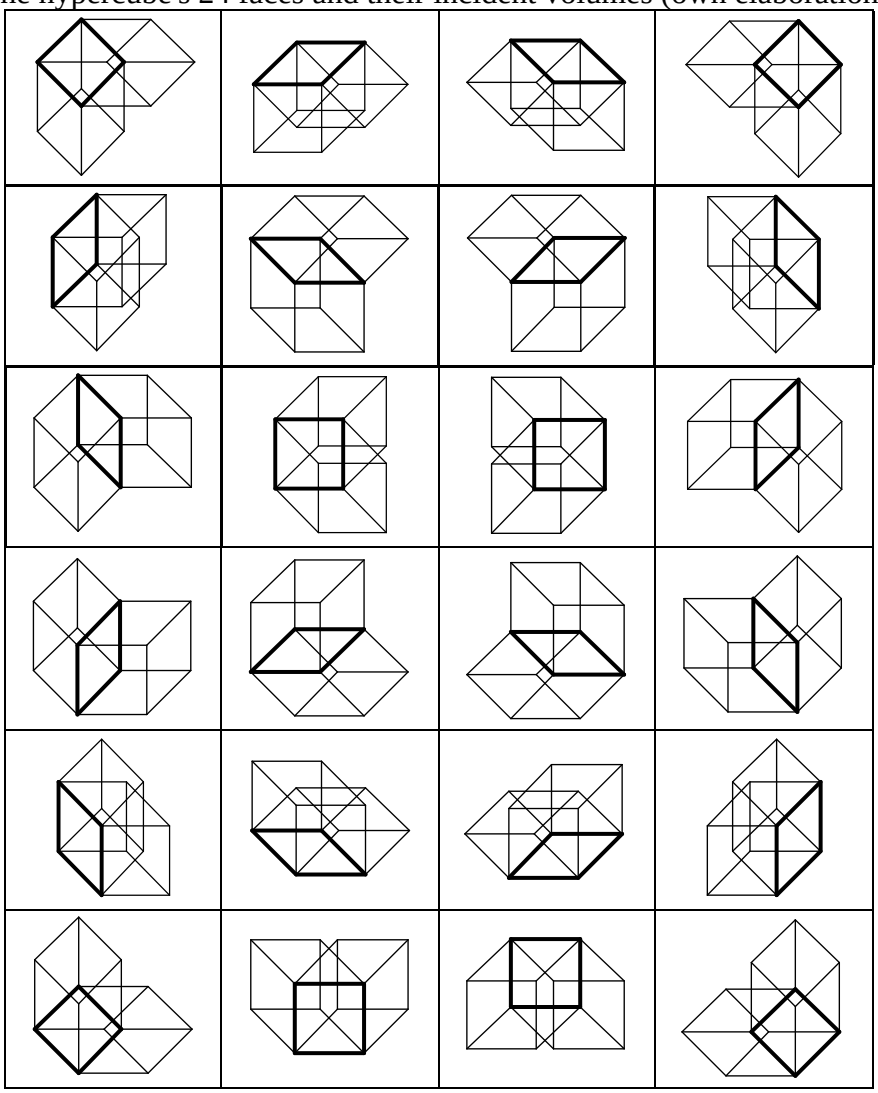


FIGURE 2.7
Viewing the hypercube's boundary volumes (taken from [Aguilera & Pérez, 02]).

[Coxeter, 84] also establishes that besides these eight volumes the hypercube's boundary is composed by 24 faces, 32 edges and 16 vertices. Every face is shared by two cubes that don't lie on the same three-dimensional space, but rather both have rotated about the plane represented by the common face until the two three-dimensional spaces represented by the cubes form a right angle (**Table 2.2**)

TABLE 2.2

The hypercube's 24 faces and their incident volumes (own elaboration).



If we position the hypercube where one of its vertices is at the origin and six of its faces coincide each one with some of X_1X_2 (XY), X_1X_4 (XW), X_2X_3 (YZ), X_2X_4 (YW), X_3X_1 (ZX), and X_3X_4 (ZW) planes, then we have the positive coordinates presented in **Table 2.3**.

TABLE 2.3

The hypercube's coordinates (own elaboration).

X_1 (X)	***X_2 (Y)***	***X_3 (Z)***	***X_4 (W)***	***Binary Representation***	***Vertices' decimal representation***
0	0	0	0	0000	0
1	0	0	0	0001	1
0	1	0	0	0010	2
1	1	0	0	0011	3
0	0	1	0	0100	4
1	0	1	0	0101	5
0	1	1	0	0110	6
1	1	1	0	0111	7
0	0	0	1	1000	8
1	0	0	1	1001	9
0	1	0	1	1010	10
1	1	0	1	1011	11
0	0	1	1	1100	12
1	0	1	1	1101	13
0	1	1	1	1110	14
1	1	1	1	1111	15

We can observe that all vertices' coordinates presented in **Table 2.3** can be used to describe a binary number, where X_1 coordinate is the less significant digit and X_4 coordinate is the most significant digit. In this way, we have that using the decimal representation for those binary numbers, we can refer, for example, to vertex 14 as that whose coordinates are (0,1,1,1). In this work, we will refer to the hypercube's vertices using their decimal representation.

2.2.1.3 Counting the Number of Lower Dimensional Elements in a nD Hypercube

From the Bragdon's method presented in section 2.2.1.1, it is easy to observe that each time we move C_n to generate C_{n+1} the number of vertices doubles, because we have an initial and a final position. From this analysis, we can conclude that the number of vertices in a C_n is 2^n.

Definition 2.2: *Let* **Q(n,k)** *be the number of kD cubes in a nD hypercube, i.e. the number of C_k's in a C_n for* $0 \le k \le n$.

To compute Q(n,k) we must first calculate how many C_k's are incident to each vertex in a C_n. There are n incident edges to each vertex in C_n and we get a C_k for each subset of k distinct edges taken from these n incident edges (this property can be visualized observing the sequence of **Figures 2.2 to 2.5**). For instance, the number of kD cubes at each vertex of a nD hypercube is:

$$C(n,k) = \frac{n!}{k!(n-k)!}$$

Because we have $C(n,k)$ C_k's in each one of the 2^n C_n's vertices, we get $2^n \cdot C(n,k)$ C_k's. However, each C_k is counted 2^k times, for consequence, we must divide the intermediate formula by this number to get the final formula (presented in [Coxeter, 63] and [Banchoff,96]):

$$Q(n,k) = \frac{2^n \cdot C(n,k)}{2^k} = 2^{n-k} \cdot C(n,k)$$

See **Table 2.4** for the formula's application.

TABLE 2.4

Obtaining the properties of a point, a segment, a square, a cube and a hypercube (own elaboration).

nD hypercubes (C_n) kD cubes (C_k)	**0 (point)**	**1 (segment)**	**2 (square)**	**3 (cube)**	**4 (hypercube)**
0 (vertices)	1	2	4	8	16
1 (edges)		1	4	12	32
2 (faces)			1	6	24
3 (volumes)				1	8
4 (hypervolumes)					1
kD cubes sum	**1**	**3**	**9**	**27**	**81**

[Banchoff, 96] points that the sum of the C_k's in each column (in **Table 2.4**) provides a power of 3. Furthermore, we present the following

Theorem 2.1: $\sum_{k=0}^{n} Q(n,k) = 3^n \quad \forall n \in \mathrm{N}$

Proof: By substituting Q(n,k) with its formula we obtain

$$\sum_{k=0}^{n} Q(n,k) = \sum_{k=0}^{n} 2^{n-k} \cdot C(n,k)$$

where the right hand side is a particular case of the well known Binomial Theorem:

$$(a+b)^n = \sum_{k=0}^{n} C\binom{n}{k} a^{n-k} b^k$$

when $a = 2$ and $b = 1$, which evaluates to 3^n. ◪

Corolary 2.1: *Each term in the sum* $\sum_{k=0}^{n} C\binom{n}{k} \cdot 2^{n-k}$ *represents the number of* C_k*'s in a* C_n.

For example, for the cube (C_3) we have:

$$\sum_{0}^{3} C\binom{3}{k} \cdot 2^{3-k} = C\binom{3}{0} \cdot 2^3 + C\binom{3}{1} \cdot 2^2 + C\binom{3}{2} \cdot 2^1 + C\binom{3}{3} \cdot 2^0 = 8 + 12 + 6 + 1$$

which corresponds to the number of vertices (8), edges (12), faces (6) and cubes (1) respectively.

Corolary 2.2: *The total number of all lower dimensional boundary elements in C_n is 3^n-1.*

Proof: By **Corolary 2.1** each term in the sum

$$\sum_{k=0}^{n} Q(n,k) = \sum_{k=0}^{n} 2^{n-k} \cdot C(n,k)$$

corresponds to the number of C_k's on the C_n's boundary, except the last term (when k = n) which evaluates to 1 (one) and corresponds to C_n itself. Therefore the number of all bounding elements in C_n is

$$\sum_{k=0}^{n-1} Q(n,k) = 3^n - 1 \qquad \square$$

For example, a cube (C_3) has $3^n - 1 = 26$ bounding elements (8 vertices, 12 edges and 6 faces). Moreover, the above corollary with n = 4 satisfies the Coxeter's counting of the hypercube's (C_4) boundary (see section 2.2.1.2).

2.2.1.4 Coordinates for the nD Hypercube

[Coxeter, 63] establishes that the coordinates for a nD hypercube with edges of length 2 can be described in general as:

$$(\underbrace{\pm 1,\ldots,\pm 1}_{n})$$

For example, using the above description, the coordinates for a square (n = 2) are:

$$(+1,+1)$$
$$(+1,-1)$$
$$(-1,+1)$$
$$(-1,-1)$$

If we apply the translation (1,…,1), and the scaling (0.5,…,0.5) we obtain the general set of coordinates for a unit C_n:

$$(\underbrace{0,0,...0,0}_{n}),\ (\underbrace{1}_{1},\underbrace{0,...,0,0}_{n-1}),\ ...,\ (\underbrace{1,...,1}_{i},\underbrace{0,...,0}_{n-i}),\ ...,\ (\underbrace{1,1,...,1}_{n-1},\underbrace{0}_{1}),\ (\underbrace{1,1,...,1,1}_{n}) =$$

$$(1^0,0^n),\ (1^1,0^{n-1}),\ ...,\ (1^i,0^{n-i}),\ ...,\ (1^{n-1},0^1),\ (1^n,0^0)$$

where the coordinates must be permuted in base of the following distribution:

$$C\binom{n}{0},\ C\binom{n}{1},\ ...,\ C\binom{n}{i},\ ...,\ C\binom{n}{n-1},\ C\binom{n}{n}$$

where $C\binom{n}{i} = \frac{n!}{i!(n-i)!}$ defines the number of those coordinates that have i ones and $n-i$ zeros. Then we can evaluate and relate the previous distribution with the number of vertices in the C_n:

$$1 + n + ... + \frac{n!}{i!(n-i)!} + ... + n + 1 = \sum_{i=0}^{n} C\binom{n}{i} = 2^n$$

Table 2.5 shows the application of the procedure on the 4D hypercube.

TABLE 2.5

Defining the 4D hypercube's vertices coordinates (own elaboration).

Value of i	Number of Combinations	Coordinates	Vertex (Decimal representation)
0	1	(0,0,0,0)	0
1	$C\binom{4}{1} = 4$	(1,0,0,0) (0,1,0,0) (0,0,1,0) (0,0,0,1)	1 2 4 8
2	$C\binom{4}{2} = 6$	(1,1,0,0) (1,0,1,0) (0,1,1,0) (1,0,0,1) (0,1,01) (0,0,1,1)	3 5 6 9 10 12
3	$C\binom{4}{3} = 4$	(1,1,1,0) (1,1,0,1) (1,0,1,1) (0,1,1,1)	7 11 13 14
4	1	(1,1,1,1)	15

2.2.2 The Simplex

In [Coxeter, 63] is presented a method for obtaining a family of polytopes called the ***simplexes***. Basically, these polytopes are the simplest that can be generated in their respective spaces. First, consider a 0D point which in fact (and obviously) describes a 0D space (**Figure 2.8**). Now, select any point not embedded in this 0D space. The two points' union will generate a segment, the 1D simplex (In **Figure 2.9** we are considering the special case when the new point is on the X_1-axis).

FIGURE 2.8

The 0D simplex (own elaboration).

FIGURE 2.9
Generation and final 1D simplex, a segment (own elaboration).

Now, a third point is selected in the way that it is not embedded in the straight line (a 1D space) defined by the original segment. The new point is united to the 1D simplex's two vertices, which generates a triangle, a 2D simplex (In **Figure 2.10** we consider the special case when the new third point is on a line parallel to the X_2-axis).

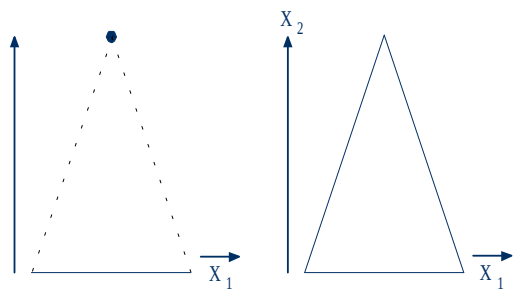


FIGURE 2.10
Generation and final 2D simplex, a triangle (own elaboration).

The next step is consider a fourth point which is not embedded in the plane (a 2D space) defined by the original triangle. The new point is united to the 2D simplex's three vertices, which generates a tetrahedron, a 3D simplex (In **Figure 2.11** we consider the special case when the new fourth point is on a line parallel to X_3-axis).

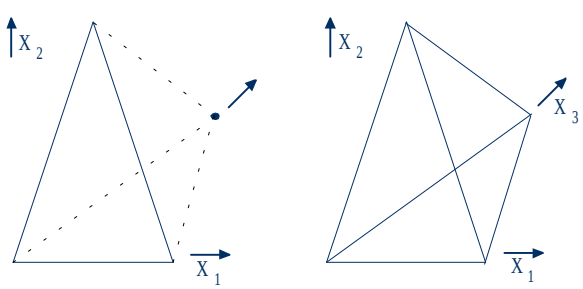


FIGURE 2.11
Generation and final 3D simplex, a tetrahedron (own elaboration).

We select a new fifth point which is not embedded in the hyperplane (a 3D space) defined by the original tetrahedron. The new point is united to the 3D simplex's four vertices, which generates a 4D simplex (In **Figure 2.12** we consider the special case when the new fifth point is on a line parallel to the X_4-axis).

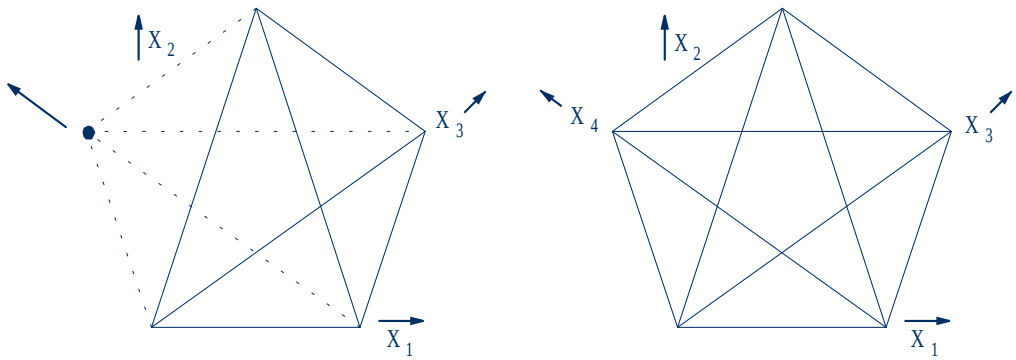


FIGURE 2.12
Generation and final 4D simplex (own elaboration).

Any n+1 lineally independent points (or in other words, all the points don't lie in a [n-k]-dimensional hyperplane) will be the n-dimensional simplex's vertices. The elements in the nD simplex's boundary will be all (n-1), (n-2), …, 1, 0-dimensional simplexes composed by the possible subsets of the n+1 points. Then, a nD simplex will have:

$$C\binom{n+1}{1} \text{ vertices, } C\binom{n+1}{2} \text{ edges, } C\binom{n+1}{3} \text{ faces, } C\binom{n+1}{4} \text{ volumes, ...}$$

Let N_k^n be the number of the k-dimensional elements on the n-dimensional simplex's boundary. Its formula is then ([Sommerville, 58] & [Coxeter, 63]):

$$N_k^n = C\binom{n+1}{k+1}$$

By applying the formula on the 4D simplex, it is found that it has 5 vertices (0D), 10 edges (1D), 10 (triangular and 2D) faces and 5 (tetrahedrical and 3D) volumes (see **Figure 2.13**).

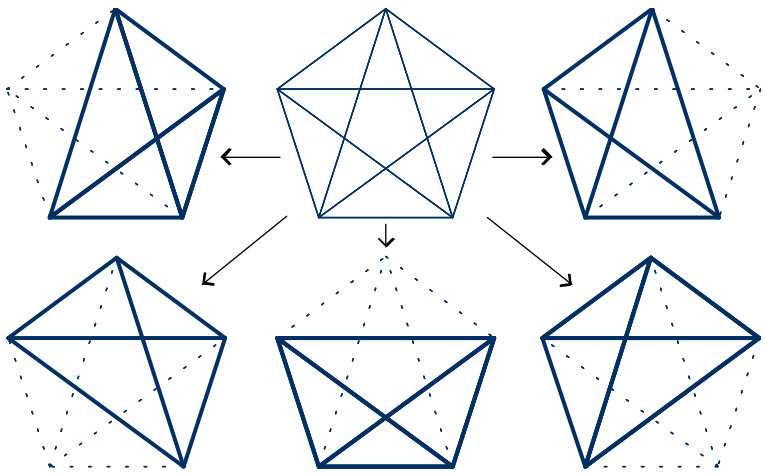


FIGURE 2.13
Viewing the 4D simplex's five boundary volumes (Own elaboration).

[Coxeter, 63] points out that the well known relation

$$C\binom{n+1}{k+1} = C\binom{n}{k+1} + C\binom{n}{k}$$

represents the simplex's construction as a "piramid", with any cell as "base" and the vertex, which is outside the (n-1)D hiperplane described by the "base", as its "apex" (see **Figures 2.8 to 2.12**). The number of k-dimensional elements that compose the "base" are indicated by the first term (when n = k + 1 the base counts itself), while the number of k-dimensional elements that are incident to the apex are counted by the second term (when n = k + 1 the apex counts itself). **Table 2.6** shows the application of the relation on the tetrahedron.

TABLE 2.6

The kD elements of the tetrahedron built as a "pyramid" (own elaboration).

kD elements	**kD elements that compose the base** $C\begin{pmatrix} n \\ k+1 \end{pmatrix}$	**kD elements incident to the apex** $C\begin{pmatrix} n \\ k \end{pmatrix}$
Vertices – 0D	3	1
Edges – 1D	3	3
Faces – 2D	1	3

The coordinates of the nD simplex's first n vertices are given by permuting:

$$(\underbrace{1,0,0,...,0}_{n})$$

While the vertex n+1 is the origin [Cohen, 79]. For example, the vertices' coordinates for the 4D simplex are presented in the **Table 2.7**.

TABLE 2.7

The vertices' coordinates for the 4D simplex (own elaboration).

Vertices (n = 4)	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5 (n+1)	0	0	0	0

2.2.3 The Cross Polytope

The methodology presented by [Coxeter, 63] will be used again for generating a third family of polytopes, the **Cross Polytopes**. Just as in both previous methodologies (hypercube and simplex) we start with a 0D point (**Figure 2.14**). Besides the original point it will be generated other two which are translated in opposite directions along the new first

dimension, represented by the X_1 axis. Both points are joined to the first one to compose the 1D cross polytope (**Figure 2.15**), that is, a segment (this is the only case where the original first point is eliminated from the final segment, because it is the common point of the two segments composed by the union of the two new points on the X_1 axis and it).

FIGURE 2.14
The 0D cross polytope (own elaboration).

FIGURE 2.15
Generation and final 1D cross polytope, a segment (own elaboration).

In the 1D space are generated two new points in addition to the existent two. The new points are translated in opposite directions along the new second dimension, the X_2 axis. Both points then are joined with the original two to compose the 2D cross polytope, a square (**Figure 2.16**).

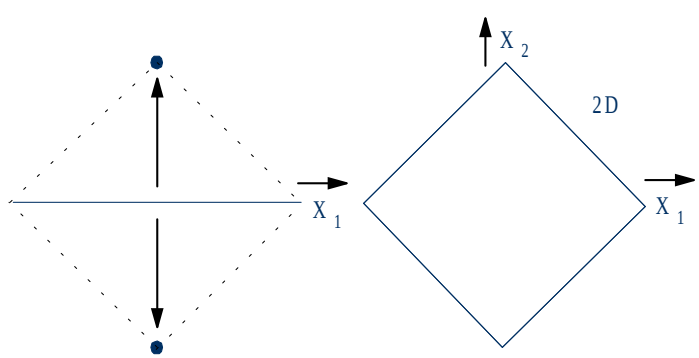


FIGURE 2.16
Generation and final 2D cross polytope, a square (own elaboration).

Again, two new points, additionally to the existent four, are generated in the 2D space. Both points are translated in opposite directions along the new third dimension, which is represented by X_3 axis. The new points are joined to the original four to compose the 3D cross polytope (**Figure 2.17**), or in other words, a octahedron.

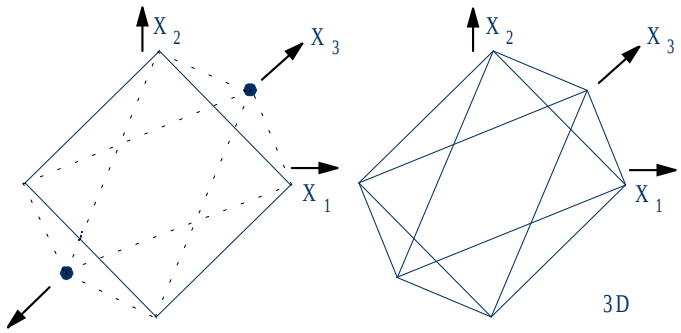


FIGURE 2.17
Generation and final 3D cross polytope, an octahedron (own elaboration).

Finally, two new points are generated, additionally to the six original points in the 3D space, and translated in opposite directions along the new fourth dimension, that is, the X_4 axis. As in the previous steps, both points are joined to the original six, in this way, it is obtained the 4D cross polytope (**Figure 2.18**).

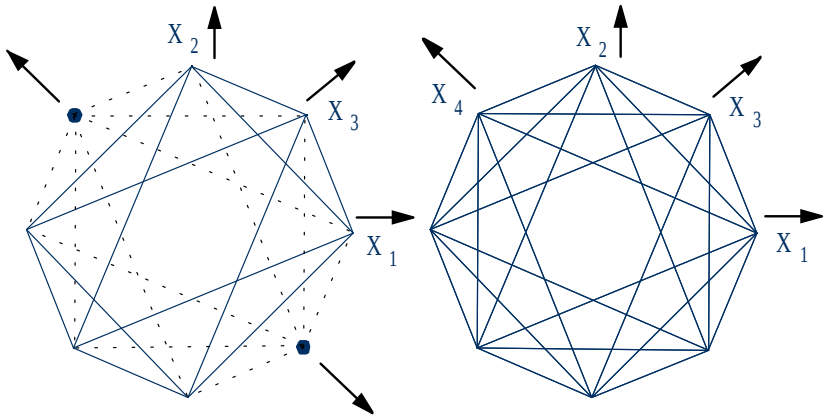


FIGURE 2.18
Generation and final 4D cross polytope (own elaboration).

[Coxeter, 63] points out that the nD cross polytope can be considered as a dipiramid which is based in the (n-1)D cross polytope, where there are two apexes in both directions of the new dimension. See the cross polytope's building sequence in **Figures 2.14 to 2.18**. For example, octahedron is a dipiramid based in the square (**Figure 2.17**) while the 4D cross polytope is a 4D "dipiramid" based in the octahedron (**Figure 2.18**). Both cross polytopes have their pair of apexes in both directions of the third and four dimensions respectively.

One of the fundamental properties of the Euclidean n-dimensional space is the possibility of configuring n mutually perpendicular lines passing through any point [Coxeter, 63]. When selecting equidistant points from the origin along the main axis in both directions, the cross polytope's 2n vertices are defined (the main axis compose a "cross", that the origin for the polytope's name). Then, the vertices' coordinates for the nD cross polytope with edges of length $\sqrt{2}$ are given by permuting:

$$(\underbrace{\pm 1,0,\ldots,0}_{n})$$

For example, the vertices' coordinates for the 4D cross polytope are presented in **Table 2.8**.

TABLE 2.8
The vertices' coordinates for the 4D cross polytope (own elaboration).

Vertex	X_1	X_2	X_3	X_4
1	1	0	0	0
2	-1	0	0	0
3	0	1	0	0
4	0	-1	0	0
5	0	0	1	0
6	0	0	-1	0
7	0	0	0	1
8	0	0	0	-1

When there are considered only n points which are equidistant from the origin along the main axis, the (n-1)-dimensional simplex's vertices are defined. In fact, this (n-1)D simplex is embedded in a nD space's hyper-octant. For example, in **Figure 2.19** is presented a triangle embedded in the octant which is defined by the positive sides of 3D space's X_1, X_2 and X_3 axis.

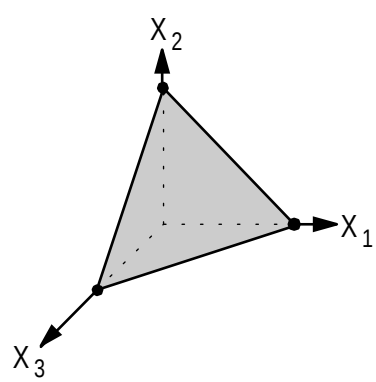


FIGURE 2.19
Triangle embedded in the octant defined by the positive sides of the 3D space's X_1, X_2 and X_3 axis (own elaboration).

Consider the cross polytope whose coordinates are based in permuting $(\pm 1,0,...,0)$. Therefore, due to the existence of 2^n possible hyper-octants in nD space, the number of (n-1)D simplexes on the cross polytope's boundary will be 2^n [Coxeter, 63]. For example,

octahedron is composed by 8 triangular faces (one for each octant in 3D space), while the 4D cross polytope has 16 tetrahedrical volumes on its boundary (**Figure 2.20**), one of each 4D space's hyper-octant.

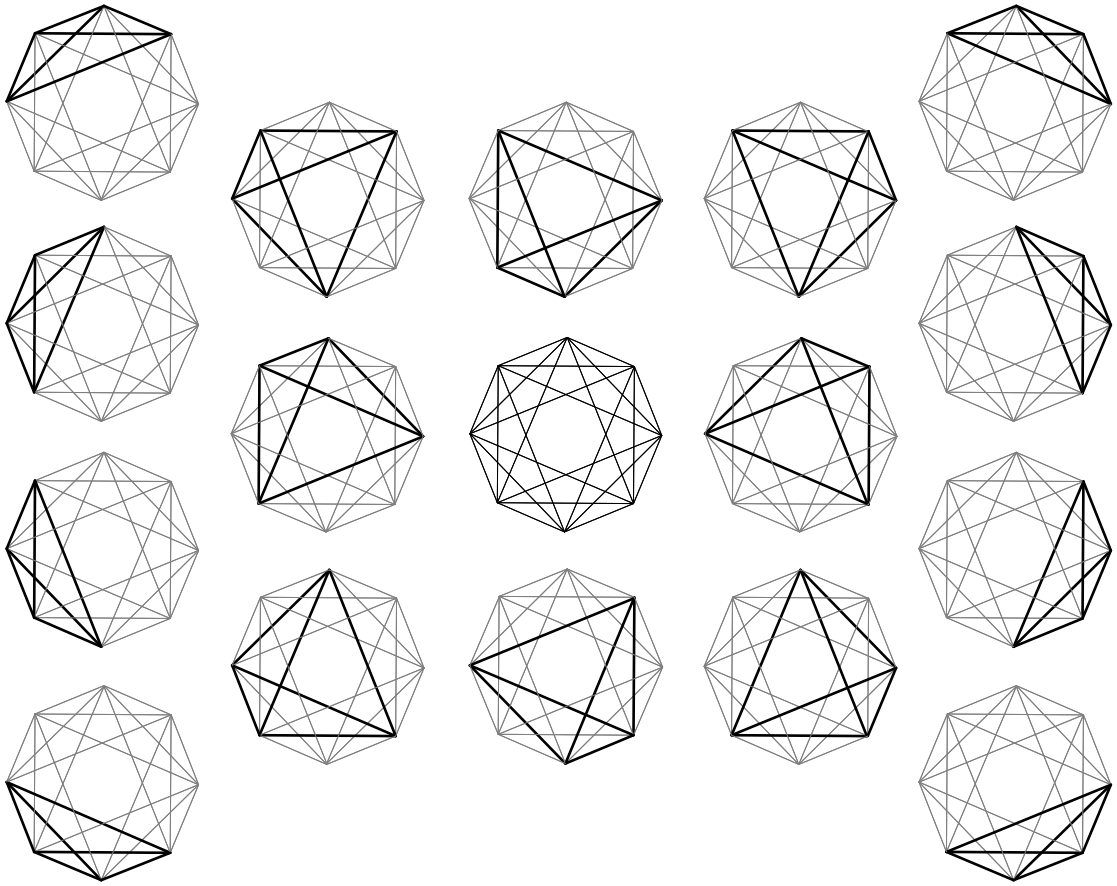


FIGURE 2.20
Viewing the 4D cross polytope's 16 boundary volumes (Own elaboration).

The cross polytope can be obtained, from another point of view, through the Duality Principle [Banchoff, 96]. In brief, for building a polyhedron's dual a vertex in the center of each face is chosen. The vertices' final set obtained in this way, defines a dual polyhedron. As an example, we have the well known duality between the cube and the octahedron.

The Duality Principle can be applied in the same way in the 4D space (and in the subsequent spaces), therefore, the 4D hypercube's eight volumes correspond to the 4D cross polytope's eight vertices. The duality between nD hypercube and cross polytope generates a correspondence between their elements on their respective boundaries. The counting of the elements on the 4D cross polytope's boundary by [Sommerville, 58] establishes that it has 8 vertices, 24 edges, 32 faces and 16 volumes. See **Table 2.9** to appreciate the relation between the 4D hypercube and cross polytope's elements.

TABLE 2.9
Correspondences between elements
on the 4D hypercube and cross polytope's boundaries (own elaboration).

4D Hypercube (k)	**4D Cross Polytope (k)**	**Number of kD elements**
0	3	16
1	2	32
2	1	24
3	0	8

The duality, between hypercubes and cross polytopes, establishes that counting the hypercube's k-dimensional elements is equivalent to counting the cross politope's (n-k-1)-dimensional elements [Sommerville, 58]. It is known that the hypercube's k-dimensional elements counting (see Section 2.2.1.3) is given by:

$$Q(n,k) = 2^{n-k} \cdot C\binom{n}{k}$$

If in the above formula, k (the dimensionality of the elements on the hypercube's boundary) is replaced by n-k-1, or in other words, the k-dimensional elements on the cross polytope's boundary will be counted using the hypercube's formula, then we will have:

$$Q(n,k) = 2^{n-(n-k-1)} \cdot C\binom{n}{n-k-1} = 2^{k+1} \cdot C\binom{n}{n-(k+1)}$$

Be (k+1) = r, then by applying the well known relation:

$$C\binom{n}{r} = C\binom{n}{n-r}$$

The formula will be finally:

$$Q(n,k) = 2^{k+1} \cdot C\binom{n}{k+1}$$

Which coincides with that presented by [Coxeter, 63] for counting the k-dimensional elements on the nD cross polytope's boundary.

2.2.4 The 0/1-Polytopes

Let C_n be a n-dimensional hypercube whose vertices' coordinates hold the decimal representation (see **Table 2.3**). Consider the convex hull of the subset $V \subseteq C_n$ of the n-dimensional hypercube's vertices. The polytope P that represents the convex hull of V is called a 0/1-Polytope or a subpolytope of the hypercube [Aichholzer, 00] (it receives the name of 0/1-Polytope because its vertices' coordinates are precisely 0's and 1's). The subset V will describe a valid 0/1-Polytope under three conditions: its cardinality must be greater than n (remember that n+1 points are required to compose a simplex); all its vertices doesn't lie in a (n-1)-dimensional hyperplane; and finally, it must be convex (this condition is obviously assured by its relation with the convex hull).

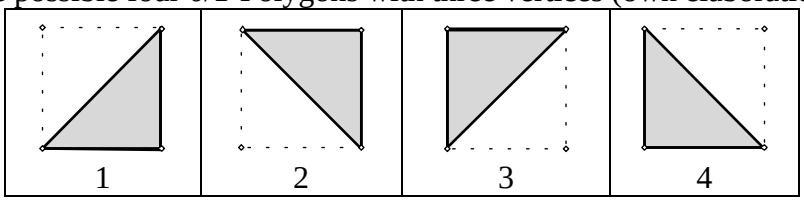
There are $2^{(2^n)}$ possible subsets of the nD hypercube's vertices. The subsets with n vertices or less are not considered. Furthermore, a great part of the subsets are equivalent, in other words, they are reflections or rotations between each other. A "Class of Vertex

Sets” is defined by [Aichholzer, 97] as the one that contains sets of vertices that can be transformed in another set, in the same class, by applying some transformation of reflection or rotation.

There are four sets of 3 vertices taken from the square (in 2D space), which are presented in **Table 2.10**. For example, suitable rotations can be applied to combination 1 for obtaining combinations 2, 3 and 4. Moreover, a suitable reflection can be applied to combination 2 for obtaining combination 3 (in a similar way to combinations 1 and 4). Therefore, all the combinations belong to the same class.

TABLE 2.10

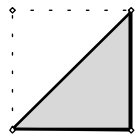
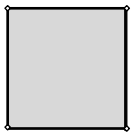
The possible four 0/1-Polygons with three vertices (own elaboration).



In the 2D space there are $2^4 = 16$ subsets of vertices taken from a square. The possible eleven sets with 0, 1 and 2 vertices are not considered. Therefore, it will be considered 4 sets with 3 vertices (see the previous paragraph) and one set with four vertices. Finally, by the application of the transformations of rotation and reflection, there are only two classes of 0/1-Polygons in the 2D space (see **Table 2.11**), one class with three vertices (a triangle) and one class with four vertices (a square).

TABLE 2.11

The two 0/1-Polygons (own elaboration).

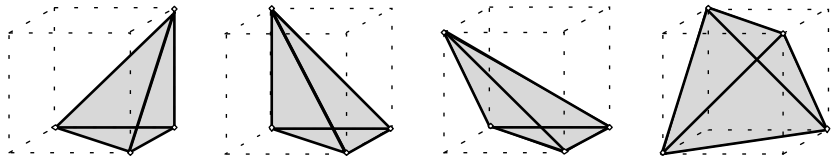
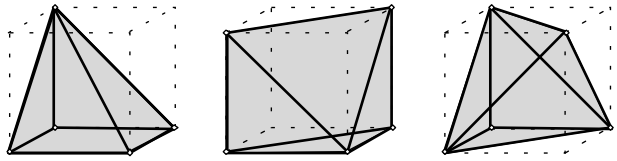
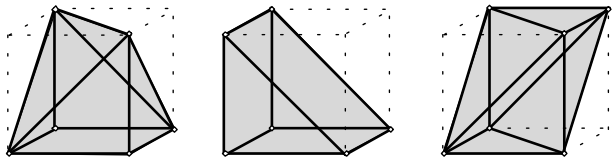
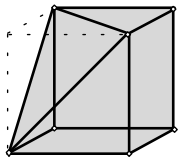
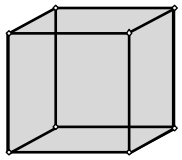
	
3 vertices	4 vertices

There are $2^8 = 256$ possible sets of vertices taken from a cube. Those sets with 0, 1, 2 and 3 vertices won't be considered. By applying rotations and reflections between the remaining sets with the same number of vertices, [Aichholzer, 00] has concluded the existence of 12 (classes) 0/1-Polyhedra in the 3D space (see **Table 2.12**). The distribution for the considered combinations of vertices is the following :

- $C\binom{8}{4} = 70$ sets with four vertices (4 classes).
- $C\binom{8}{5} = 56$ sets with five vertices (3 classes).
- $C\binom{8}{6} = 28$ sets with six vertices (3 classes).
- $C\binom{8}{7} = 8$ sets with seven vertices (1 class).
- $C\binom{8}{8} = 1$ set with one vertex (1 class).

TABLE 2.12

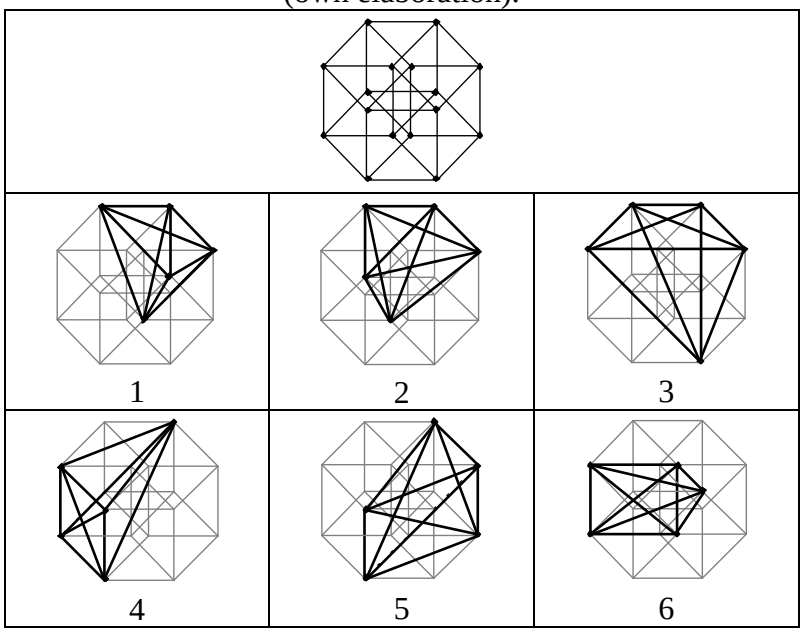
The twelve 0/1-Polyhedra (own elaboration).

Vertices Number	**0/1-Polyhedra**
4	
5	
6	
7	
8	

It is known that the 4D hypercube has 16 vertices on its boundary. From these vertices, a total of 2^{16} = 65,536 sets with 0 to 16 vertices can be formed. The sets with 0, 1, 2, 3 and 4 vertices will be ignored. It must be considered, with the remaining sets, that all their vertices don't lie in a 3D hyperplane. In the **Table 2.13** are shown some 4D 0/1-Polytopes' classes, in fact, six examples of sets with five vertices are presented (all the 4D 0/1-Polytopes with five vertices are simplexes; in general, all the nD

0/1-Polytopes with n+1 vertices are simplexes). [Hill, 98] presents a counting of 402 classes for the 4D 0/1-Polytopes (however, [Hill, 98] considered all the 2^{16} possible sets).

TABLE 2.13
Some 4D 0/1-Polytopes (1 to 6) composed by five vertices: simplexes (own elaboration).



It has been mentioned that the number of sets of vertices taken from the nD hypercube is $2^{(2^n)}$. When n < 5, one of the most common methodologies for finding the classes for 0/1-Polytopes is the exhaustive searching [Aichholzer, 97]. However, for determining the 5D 0/1-Polytopes' classes it must be considered the existence of 2^{32} = 4,294,967,296 possible sets (with 0 to 32 vertices). Moreover, for determining the 6D 0/1-Polytopes' classes it must be considered 2^{64} sets (with 0 to 64 vertices). In [Aichholzer,00] is presented a methodology, that minimizes the complexity imposed by the exhaustive searching, for determining the 5D 0/1-Polytopes' classes (in fact, [Aichholzer,00] reports 1,226,525 classes).

2.3 The 4D Geometric Transformations

2.3.1 The 3D Geometric Transformations as Extension of the 2D Geometric Transformations

[Hearn, 95] considers 3D geometric transformations (translation, rotation, scaling, etc.) as extensions of the 2D geometric transformations for these same operations with the consideration of the X_3 coordinate.

2.3.1.1 Translations

Translating in the 2D space implies a displacement of a polygon in direction of X_1 and X_2-axis, in other words, we apply a translation over a polygon to change its position. A 2D point is converted when the translation distances t_1 and t_2 are added to the original coordinate (x_1,x_2) to move it to the new position (x_1',x_2'):

$$x_1' = x_1 + t_1$$
$$x_2' = x_2 + t_2$$

Or using homogeneous coordinates and the matrix representation:

$$\begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_1 & t_2 & 1 \end{bmatrix}$$

Based in the previous idea, **translating in 3D space** implies a displacement of an object in direction of the X_1, X_2 and X_3-axis. We translate a 3D object when it is moved in

each one of the three directions of the coordinates. We translate a point (x_1,x_2,x_3) to the position (x_1',x_2',x_3') adding the corresponding distances t_1, t_2 and t_3:

$$x_1' = x_1 + t_1$$
$$x_2' = x_2 + t_2$$
$$x_3' = x_3 + t_3$$

or

$$\begin{bmatrix} x_1' & x_2' & x_3' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_1 & t_2 & t_3 & 1 \end{bmatrix}$$

2.3.1.2 Scaling

Scaling in 2D space implies a change of size (and in some cases of shape and position) of an object through two factors each one with relation with X_1 and X_2 coordinates. A 2D point is converted when it is multiplied by the scaling factors S_1 and S_2 to produce the transformed coordinates (x_1', x_2'). The scaling factor S_1 scales objects in the direction parallel to X_1 axis, while the scaling factor S_2 scales objects in the direction parallel to X_2 axis. We have then:

$$x_1' = S_1 \cdot x_1$$
$$x_2' = S_2 \cdot x_2$$

or

$$\begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_1 & 0 & 0 \\ 0 & S_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Again, it is possible to extend the previous 2D concept and to conclude that **scaling in 3D space** implies a change of size of a polyhedron by three factors each one with relation with X_1, X_2 and X_3 coordinates. We convert a 3D point (x_1, x_2, x_3) when it is multiplied by the corresponding scaling factors S_1, S_2 and S_3 to get the coordinates (x_1', x_2', x_3'):

$$x_1' = S_1 \cdot x_1$$
$$x_2' = S_2 \cdot x_2$$
$$x_3' = S_3 \cdot x_3$$

or

$$\begin{bmatrix} x_1' & x_2' & x_3' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & S_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.1.3 Rotations

The rotation's case is more special than translation and scaling. **Rotation in 2D space** is always given about a point. However, in the 2D space there are an infinite number of points. [Hearn, 95] defines a 2D rotation as the change of position of a figure along a circumference's trajectory in the 2D space (the X_1X_2 plane, for example). The 2D points can rotate an angle θ around the origin, which is the easiest point, then we have that a rotation is defined mathematically as:

$$x_1' = x_1 \cdot \cos\theta - x_2 \cdot \sin\theta$$
$$x_2' = x_1 \cdot \sin\theta + x_2 \cdot \cos\theta$$

or

$$\begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

However, in the 3D space a rotation must be given about a line and there are an infinite number of lines in space. The easiest way to calculate a rotation in the 3D space is when the rotating axis (a line) coincides with the X_1, X_2 or X_3 axis. Moreover, if we consider the 2D space as the X_1X_2 plane where rotations are around an axis that is perpendicular to such plane, then we have the first of the main rotating axis in 3D space, specifically rotation around X_3 axis [Hearn, 95]:

$$\begin{aligned} x_1' &= x_1 \cdot \cos\theta - x_2 \cdot \sin\theta \\ x_2' &= x_1 \cdot \sin\theta + x_2 \cdot \cos\theta \\ x_3' &= x_3 \end{aligned}$$

or

$$R_3(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

[Hearn, 95] points that equations for rotations around X_1 and X_2 axis can be obtained with the following cyclic substitutions:

$$x_3 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3$$

Then, applying the indicated substitutions over the equations for rotation around X_3 axis, we have the rotation around X_1 axis:

$$\begin{aligned} x_2' &= x_2 \cdot \cos\theta - x_3 \cdot \sin\theta \\ x_3' &= x_2 \cdot \sin\theta + x_3 \cdot \cos\theta \\ x_1' &= x_1 \end{aligned}$$

or

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And now, applying the indicated substitutions over the equations for rotation around X_1 axis, we have the rotation around X_2 axis:

$$x_3' = x_3 \cdot \cos\theta - x_1 \cdot \sin\theta$$
$$x_1' = x_3 \cdot \sin\theta + x_1 \cdot \cos\theta$$
$$x_2' = x_2$$

or

$$R_2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.2 The 4D Geometric Transformations as Extension of the 3D Geometric Transformations

Probably the simplest transformation operations to be derived from 3D space to 4D space are translation and scaling. We saw before how the definition of 2D translation and scaling were easily adapted for working inside the 3D space. Now, we will define these geometric transformations in the 4D and nD spaces.

2.3.2.1 Translations in the 4D and nD Spaces

The **translation in the 4D space** implies the displacement of a 4D polytope in direction of X_1, X_2, X_3 and X_4-axis with the following equations and transformation matrix (which are a simple extension of translation in the 3D space):

$$x_1' = x_1 + t_1$$
$$x_2' = x_2 + t_2$$
$$x_3' = x_3 + t_3$$
$$x_4' = x_4 + t_4$$

or

$$\begin{bmatrix} x_1' & x_2' & x_3' & x_4' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ t_1 & t_2 & t_3 & t_4 & 1 \end{bmatrix}$$

Finally, the **translation in the nD space** will imply to position a nD polytope in a new location by adding translation values to each one of its points' n coordinates. Each point (x_1, x_2, x_3, x_4, …,x_n) will be moved t_1 units parallel to the X_1-axis, t_2 units parallel to the X_2-axis, and so forth until it is moved t_n units parallel to the X_n-axis. In this way, the new point (x_1', x_2', x_3', x_4', …, x_n') is obtained. This operation is described through the following matrix operation:

$$[\underbrace{x_1' \quad x_2' \quad x_3' \quad x_4' \quad \cdots \quad x_n' \quad 1}_{n+1}] = [\underbrace{x_1 \quad x_2 \quad x_3 \quad x_4 \quad \cdots \quad x_n \quad 1}_{n+1}] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ t_1 & t_2 & t_3 & t_4 & \cdots & t_n & 1 \end{bmatrix}_{(n+1)\times(n+1)}$$

The translation matrix will have n+1 columns and n+1 rows because we are considering that the points will have the homogeneous representation. All the elements in

the matrix's main diagonal will be 1's. The translation values are located in the last row each one positioned in the column that corresponds to their respective axis. The matrix's remaining elements are 0's.

2.3.2.2 Scaling in the 4D and nD Spaces

Scaling in 4D space will imply a change of size of a polytope by four factors each one with relation with X_1, X_2, X_3 and X_4 axes. We convert a 4D point (x_1 ,x_2, x_3, x_4) when it is multiplied by the corresponding scaling factors S_1, S_2, S_3 and S_4 to get the coordinates (x_1', x_2', x_3', x_4'):

$$x_1' = S_1 \cdot x_1$$
$$x_2' = S_2 \cdot x_2$$
$$x_3' = S_3 \cdot x_3$$
$$x_4' = S_4 \cdot x_4$$

or

$$\begin{bmatrix} x_1' & x_2' & x_3' & x_4' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_1 & 0 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 & 0 \\ 0 & 0 & S_3 & 0 & 0 \\ 0 & 0 & 0 & S_4 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The scaling in the nD space will refer to change the size of a nD polytope through the factors S_1, S_2, S_3, S_4, … and S_n along the X_1, X_2, X_3, X_4, … and X_n axes respectively. The matrix representation will show that all the scaling factors are located in the main diagonal each one positioned in the column that corresponds to their respective axis:

$$[\underbrace{x_1' \quad x_2' \quad x_3' \quad x_4' \quad \cdots \quad x_n' \quad 1}_{n+1}] = [\underbrace{x_1 \quad x_2 \quad x_3 \quad x_4 \quad \cdots \quad x_n \quad 1}_{n+1}] \cdot \begin{bmatrix} S_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & S_2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & S_3 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & S_4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & S_n & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}_{(n+1)\times(n+1)}$$

2.3.3 Rotations in 4D Space

[Banks, 92] and [Hollasch, 91] have identified that if in 2D space a rotation is given around a point, and in 3D space is given around a line, then in 4D space, in analogous way, it must be given around a plane.

[Hollasch, 91] considers that rotations in 3D space must be considered as rotations parallel to a 2D plane instead of rotations around an axis. [Hollasch, 91] supports this idea considering that given an origin of rotation and a destination point in the 3D space, the set of all rotated points for a given rotation matrix lie in a single plane, which is called the rotation plane. Moreover, the rotation axis in 3D space coincide with the normal vector of the rotation plane. The concept of rotation plane is consistent with the 2D space because all the rotated points lie in the same and only plane. Finally, with the above ideas, [Hollasch,91] constructs the six basic 4D rotation matrices around the main planes in 4D space, namely X_1X_2, X_1X_3, X_1X_4, X_2X_3, X_2X_4 and X_3X_4 planes, based in the fact that only two coordinates change for a given rotation (these changing coordinates correspond to the rotation plane):

$$R_{1,2}(\theta)=\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&\cos\theta&-sen\theta&0\\0&0&sen\theta&\cos\theta&0\\0&0&0&0&1\end{bmatrix}\qquad R_{2,3}(\theta)=\begin{bmatrix}\cos\theta&0&0&sen\theta&0\\0&1&0&0&0\\0&0&1&0&0\\-sen\theta&0&0&\cos\theta&0\\0&0&0&0&1\end{bmatrix}$$

$$R_{1,3}(\theta)=\begin{bmatrix}1&0&0&0&0\\0&\cos\theta&0&-sen\theta&0\\0&0&1&0&0\\0&sen\theta&0&\cos\theta&0\\0&0&0&0&1\end{bmatrix}\qquad R_{1,4}(\theta)=\begin{bmatrix}1&0&0&0&0\\0&\cos\theta&sen\theta&0&0\\0&-sen\theta&\cos\theta&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$$

$$R_{2,4}(\theta)=\begin{bmatrix}\cos\theta&0&-sen\theta&0&0\\0&1&0&0&0\\sen\theta&0&\cos\theta&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}\qquad R_{3,4}(\theta)=\begin{bmatrix}\cos\theta&sen\theta&0&0&0\\-sen\theta&\cos\theta&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$$

2.3.4 Rotations in the nD Space

We know that in 3D space the rotations are defined in terms of the axis around they take place. However, we know from section 2.3.3 that it is more appropriate to consider that 3D rotations take place in a plane embedded in the 3D space (the plane's normal vector coincides with the rotation axis). Using these ideas, [Duffin, 94] generalize the concept of rotation in a *n*D space ($n \geq 2$) as the rotation of an axis X_a in direction to an axis X_b. The plane described by axis X_a and X_b is what [Hollasch, 91] defined as rotation plane. [Duffin,94] presents the following general rotation matrix:

$$R_{ab}(\theta) = \begin{bmatrix} r_{ij} \left| \begin{array}{ll} r_{ii} = 1 & i \neq a, i \neq b \\ r_{aa} = \cos\theta & \\ r_{bb} = \cos\theta & \\ r_{ab} = -sin\theta & \\ r_{ba} = sin\theta & \\ r_{ij} = 0 & elsewhere \end{array} \right. \end{bmatrix}$$

The matrix $R_{ab}(\theta)$ is an identity matrix except in the intersection of columns a and b and rows a and b. Because in a *n*D space there are $C\binom{n}{2}$ main planes, this is precisely the number of main rotations for such space.

From these concepts, we must consider that a rotation can be referenced by using two notations: using the axis that describe the rotation plane or using the axis that describe the (n-2)D subspace that is fixed during the rotation. In this work we will refer to rotations using the second notation. In **Table 2.14**, we present the application of both notations on the rotations for 2D, 3D and 4D space.

TABLE 2.14

Defining the rotation plane for rotations in 2D, 3D and 4D space (own elaboration).

nD Space	**Number of possible main rotations**	**(n-2)D subspace fixed during rotation**	**Main axis that describe the fixed subspace in each possible main rotation**	**Main axis that describe the rotation plane in each possible main rotation**
2D	$C\binom{2}{2} = 1$	0D - Point	-	X_1X_2
3D	$C\binom{3}{2} = 3$	1D - Edge (axis)	X_1 X_2 X_3	X_2X_3 X_1X_3 X_1X_2
4D	$C\binom{4}{2} = 6$	2D - Plane	X_1X_2 X_2X_3 X_1X_3 X_1X_4 X_2X_4 X_3X_4	X_3X_4 X_1X_4 X_2X_4 X_2X_3 X_1X_3 X_1X_2