General *n*-Dimensional Rotations

Antonio Aguilera   Ricardo Pérez-Aguila
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas, Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir
México 72820, Cholula, Puebla
aguilera@mail.udlap.mx   sp104378@mail.udlap.mx

ABSTRACT

This paper presents a generalized approach for performing general rotations in the *n*-Dimensional Euclidean space around any arbitrary (*n*-2)-Dimensional subspace. It first shows the general matrix representation for the principal *n*-D rotations. Then, for any desired general *n*-D rotation, a set of principal *n*-D rotations is systematically provided, whose composition provides the original desired rotation. We show that this coincides with the well-known 2D and 3D cases, and provide some 4D applications.

Keywords
Geometric Reasoning, Topological and Geometrical Interrogations, 4D Visualization and Animation.

1. BACKGROUND

The *n*-Dimensional Translation

In this work, we will use a simple *n*D generalization of the 2D and 3D translations. A point $\mathbf{x} = (x_1, x_2, ..., x_n)$ can be translated by a distance vector $\mathbf{d} = (d_1, d_2, ..., d_n)$ and results $\mathbf{x}' = (x_1', x_2', ..., x_n')$, which, can be computed as $\mathbf{x}' = \mathbf{x} \cdot T(\mathbf{d})$, or in its expanded matrix form in homogeneous coordinates, as shown in Eq.1.

$$\begin{bmatrix} x_1' & x_2' & \cdots & x_n' & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ d_1 & d_2 & \cdots & d_n & 1 \end{bmatrix} \quad \text{(Eq. 1)}$$

The arctan2 function

If $\tan(\theta) = m$, then $\theta = \arctan(m)$. Moreover, if we know that $\tan(\theta) = y/x$, then a better well-known solution is $\theta = \arctan 2(y, x)$, where

$$\arctan2(y,x) = \begin{cases} \arctan(y/x) & x > 0 \\ \arctan(y/x) + \pi & x < 0 \\ \pi/2 & x = 0, y \geq 0 \\ -\pi/2 & x = 0, y < 0 \end{cases}$$

In this work we will only use arctan2.

2. INTRODUCTION

The rotation plane

[Ban92] and [Hol91] have identified that if in the 2D space a rotation is given around a point, and in the 3D space it is given around a line, then in the 4D space, in analogous way, it must be given around a plane.

[Hol91] considers that rotations in 3D space must be considered as rotations parallel to a 2D plane instead of rotations around an axis. [Hol91] 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 the 3D space coincides 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, [Hol91] constructs all six basic 4D rotation matrices around the main planes in 4D space.

Main *n*-Dimensional Rotations

We know that in the 3D space, rotations are defined in terms of the axis around they take place. However, we know 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, [Duf94] genera-

WSCG SHORT Communication papers proceedings
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.

lizes the concept of a main rotation in an nD 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 [Hol91] defined as rotation plane. [Duf94] presents the following general matrix for main rotations:

$$R_{a,b}(\theta) = \left[r_{i,j} \;\middle|\; \begin{array}{l} r_{a,a} = \cos(\theta) \\ r_{b,b} = \cos(\theta) \\ r_{a,b} = -\sin(\theta) \\ r_{b,a} = \sin(\theta) \\ r_{j,j} = 1, \quad j \neq a, \quad j \neq b \\ r_{i,j} = 0 \quad elsewhere \end{array} \right]$$

For an n-dimensional rotation, this is an $n \times n$ matrix, or an $(n+1)\times(n+1)$ matrix if homogeneous coordinates are used. It can easily be verified that in the 2D space, $R_{1,2}$ corresponds to the positive (counter clockwise) rotation around the origin, while $R_{2,1}$ corresponds to the negative (clockwise) rotation around the origin, moreover $R_{1,2}{}^{-1} = R_{2,1}$. Similarly in the 3D space, $R_{1,2}$, $R_{2,3}$ and $R_{3,1}$ respectively correspond to a positive rotation around axes X_3, X_1, and X_2 (ie., axes Z, X, and Y) while the corresponding negative rotations are $R_{2,1}$, $R_{3,2}$ and $R_{1,3}$, which, by the way, are the corresponding inverse matrices of $R_{1,2}$, $R_{2,3}$ and $R_{3,1}$.

Observation 1. Matrix $R_{a,b}(\theta)$ is almost an identity matrix except in the intersection of columns a and b with rows a and b, which means that only the coordinates a and b of a point will change after a $R_{a,b}(\theta)$ rotation, which is consistent with the 3D and 2D cases.

Observation 2. Since there are $C\binom{n}{2}$ main planes in a nD space, this is precisely the number of main rotations for such space.

Observation 3. Let e_k $(1 \leq k \leq n)$ be the unit vector along axis X_k, in the nD space, then $e_a \cdot R_{a,b}(\pi/2) = e_b$, i.e., $R_{a,b}$ moves a point on X_a towards X_b.

3. FROM MAIN TO GENERAL ROTATIONS

Although the following discussions can be found in any text book (see Hearn & Baker [Hea96] for example), they are included here to underline some key points that will be very useful when extending them to the nD case. Moreover, the 3D case will be presented in a way that facilitates its generalization to nD.

General 2D Rotations

Since the main 2D rotation $R_{1,2}$ is around the origin, a general rotation of an angle θ around a fixed point $\mathbf{a} = (a_1, a_2)$ can be obtained by the following composition: $\mathbf{x}' = \mathbf{x} \cdot T(-\mathbf{a}) \cdot R_{1,2}(\theta) \cdot T(\mathbf{a})$, which is expanded as shown in Eq. 2 Note that the inverse matrix of $T(\mathbf{a})$ is $T(-\mathbf{a})$.

General 3D Rotations

A general 3D rotation is a rotation of an angle θ around a general axis. This axis, in this work, will be represented by the supporting line of the directed segment $S = \overrightarrow{\mathbf{ab}}$ (a 1D simplex), where $\mathbf{a} = (a_1^{(0)}, a_2^{(0)}, a_3^{(0)})$ and $\mathbf{b} = (b_1^{(0)}, b_2^{(0)}, b_3^{(0)})$ are two non-coincident 3D points which we will refer as the vertices of S. See Fig 1.(i). The positive direction of the rotation is given by the right hand rule (pointing with your right thumb from **a** to **b**, the remaining four curved fingers describe the positive direction). This rotation can be achieved as the composition of a number of transformations, which firstly aligns segment S with **any** of the main axis, secondly performs the desired rotation around that main axis, and thirdly returns the rotation axis to its original position by performing the inverse of those transformations in reverse order.

Let $\mathbf{v}^{(0)} = \begin{bmatrix} a_1^{(0)} & a_2^{(0)} & a_3^{(0)} \\ b_1^{(0)} & b_2^{(0)} & b_3^{(0)} \end{bmatrix}$ be the matrix for the original vertices' coordinates, let M_k be the corresponding matrix for the k^{th} transformation. Then $\{\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots\}$, where $\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} \cdot M_k$, is a series of matrices holding the vertices coordinates at each step, $\mathbf{v}^{(k)} = \begin{bmatrix} a_1^{(k)} & a_2^{(k)} & a_3^{(k)} \\ b_1^{(k)} & b_2^{(k)} & b_3^{(k)} \end{bmatrix}$, or simply $\mathbf{v}^{(k)} = \begin{bmatrix} \mathbf{a}^{(k)} \\ \mathbf{b}^{(k)} \end{bmatrix}$.

In this work, the chosen main axis where the segment S will be directed to, is X_1. The first step is to move point **a** to the origin, therefore, $M_1 = T(-\mathbf{a})$ which makes $\mathbf{v}^{(1)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ b_1^{(1)} & b_2^{(1)} & b_3^{(1)} \end{bmatrix}$ where $b_j^{(1)} = b_j^{(0)} - a_j^{(0)}$, $j = 1,2,3$. See Fig. 1.(ii).

$$[x_1' \quad x_2' \quad 1] = [x_1 \quad x_2 \quad 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a_1 & -a_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a_1 & a_2 & 1 \end{bmatrix} \qquad \text{(Eq. 2)}$$

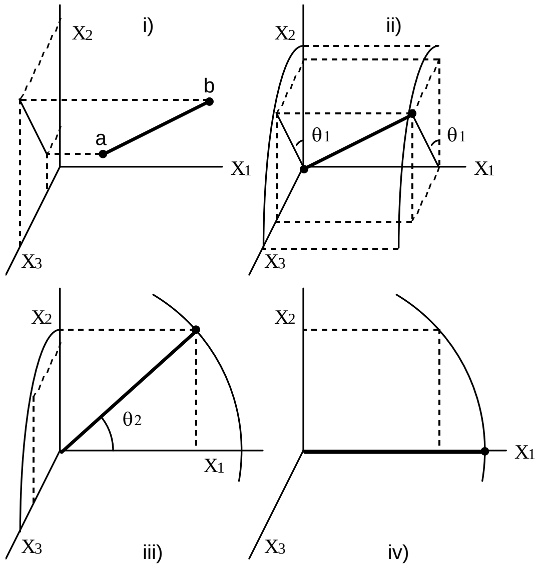


Figure 1. Set of transformations needed to align segment $S = \overrightarrow{\mathbf{ab}}$ to axis X_1.

The second step is to rotate $\mathbf{b}^{(1)}$ around axis X_1, by a suitable angle θ_1, so that the resulting $\mathbf{b}^{(2)}$ lies on main plane X_1X_2. In this way we can get rid of the third dimension by making the third coordinate equal to zero, resulting a 2D problem. It can be seen from Fig.1.(ii) that the needed rotation is $R_{3,2}(\theta_1)$, where $\tan(\theta_1) = b_3^{(1)} / b_2^{(1)}$, thus $\theta_1 = \arctan2(b_3^{(1)}, b_2^{(1)})$.

Therefore $M_2 = R_{3,2}\left(\arctan2\left(b_3^{(1)}, b_2^{(1)}\right)\right)$, which makes $\mathbf{v}^{(2)} = \begin{bmatrix} 0 & 0 & 0 \\ b_1^{(2)} & b_2^{(2)} & \mathbf{0} \end{bmatrix}$, see Fig.1.(iii).

Similarly, $M_3 = R_{2,1}\left(\arctan2\left(b_2^{(2)}, b_1^{(2)}\right)\right)$, which rotates $\mathbf{b}^{(2)}$ onto main axis X_1 making the second coordinate also zero, i.e., $\mathbf{v}^{(3)} = \begin{bmatrix} 0 & 0 & 0 \\ b_1^{(3)} & \mathbf{0} & 0 \end{bmatrix}$, see Fig.1.(iv).

These last two steps show that if we want to make the j^th coordinate equal to zero we have to make a rotation

$$M_k = R_{j,j-1}\left(\arctan2\left(b_j^{(k-1)}, b_{j-1}^{(k-1)}\right)\right) \qquad \text{(Eq. 3)}$$

At this point we have segment $S^{(3)}$ (following this notation) aligned to axis X_1, so the next step is a positive rotation of the desired angle θ around axis X_1, that is $M_4 = R_{2,3}(\theta)$. The final stage returns the rotation axis to its original position by applying M_5, M_6, and M_7, which are the inverse matrices of M_3, M_2, and M_1, respectively.

Therefore, given a general 3D rotation defined by segment $S = \overrightarrow{\mathbf{ab}}$ (the rotation linear axis) and an angle θ, every point $\mathbf{x}$ will rotate to the point $\mathbf{x}'$ defined by $\mathbf{x}' = \mathbf{x} \cdot M$, where $M = M_1 \cdot M_2 \cdot M_3 \cdot M_4 \cdot M_5 \cdot M_6 \cdot M_7$ and it is usually computed in advance.

General 4D Rotations

A general 4D rotation is a rotation of an angle θ around a general plane. This plane, in this work, will be represented by the supporting plane of a triangle $T = \mathbf{abc}$ (a 2D simplex), where $\mathbf{a} = (a_1^{(0)}, a_2^{(0)}, a_3^{(0)}, a_4^{(0)})$, $\mathbf{b} = (b_1^{(0)}, b_2^{(0)}, b_3^{(0)}, b_4^{(0)})$ and $\mathbf{c} = (c_1^{(0)}, c_2^{(0)}, c_3^{(0)}, c_4^{(0)})$ are three non-collinear 4D points which we will refer as the vertices of *T*. This rotation can be achieved as the composition of a number of transformations, which firstly takes triangle *T* onto main plane X_1X_2, so that edge **ab** of *T* is aligned with axis X_1, secondly it performs the desired rotation around main plane X_1X_2, and thirdly it returns triangle *T* to its original position by performing the inverse of those transformations in reverse order.

Let $\{\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots\}$ be the series of matrices holding the vertices' coordinates at each step, where $\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} \cdot M_k$, and $\mathbf{v}^{(0)}$ is the matrix for the original vertices' coordinates:

$$\mathbf{v}^{(0)} = \begin{bmatrix} a_1^{(0)} & a_2^{(0)} & a_3^{(0)} & a_4^{(0)} \\ b_1^{(0)} & b_2^{(0)} & b_3^{(0)} & b_4^{(0)} \\ c_1^{(0)} & c_2^{(0)} & c_3^{(0)} & c_4^{(0)} \end{bmatrix}$$

Then, and proceeding in a similar way as the 3D case, we find $M_1 = T(-\mathbf{a})$, which makes

$$\mathbf{v}^{(1)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ b_1^{(1)} & b_2^{(1)} & b_3^{(1)} & b_4^{(1)} \\ c_1^{(1)} & c_2^{(1)} & c_3^{(1)} & c_4^{(1)} \end{bmatrix}$$

From now on, we can not rely on a picture for figuring out the needed 4D rotations, therefore we must rely on Eq. 3 which rotates points in the j^th dimension towards the (j-1)^th dimension, thus we find $M_2 = R_{4,3}\left(\arctan2\left(b_4^{(1)}, b_3^{(1)}\right)\right)$, which makes

$$\mathbf{v}^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ b_1^{(2)} & b_2^{(2)} & b_3^{(2)} & \mathbf{0} \\ c_1^{(2)} & c_2^{(2)} & c_3^{(2)} & c_4^{(2)} \end{bmatrix}$$

$M_3 = R_{3,2}\left(\arctan2\left(b_3^{(2)}, b_2^{(2)}\right)\right)$, which makes

$$\mathbf{v}^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ b_1^{(3)} & b_2^{(3)} & \mathbf{0} & 0 \\ c_1^{(3)} & c_2^{(3)} & c_3^{(3)} & c_4^{(3)} \end{bmatrix}$$

and $M_4 = R_{2,1}\left(\arctan2\left(b_2^{(3)}, b_1^{(3)}\right)\right)$, which makes

$$\mathbf{v}^{(4)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ b_1^{(4)} & \mathbf{0} & 0 & 0 \\ c_1^{(4)} & c_2^{(4)} & c_3^{(4)} & c_4^{(4)} \end{bmatrix}$$

At this point edge **ab** of triangle $T^{(4)}$ lies on axis X_1, but the opposite vertex $\mathbf{c}^{(4)}$ can be anywhere in the 4D space. We need to rotate triangle $T^{(4)}$ so that this vertex and the whole triangle result embedded in main plane X_1X_2, however, we have to be careful not to apply any rotation that could move edge **ab** away from axis X_1. According to **observation 1**, any rotation $R_{a,b}$ with $a \neq 1$ and $b \neq 1$ will preserve coordinate in X_1. So we proceed that way by applying Eq. 3 but using vertex **c** instead of vertex **b**, and find $M_5 = R_{4,3}\left(\arctan 2\left(c_4^{(4)}, c_3^{(4)}\right)\right)$, which makes

$$\mathbf{v}^{(5)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ b_1^{(5)} & 0 & 0 & 0 \\ c_1^{(5)} & c_2^{(5)} & c_3^{(5)} & \mathbf{0} \end{bmatrix}$$

and $M_6 = R_{3,2}\left(\arctan 2\left(c_3^{(5)}, c_2^{(5)}\right)\right)$, which makes

$$\mathbf{v}^{(6)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ b_1^{(6)} & 0 & 0 & 0 \\ c_1^{(6)} & c_2^{(6)} & \mathbf{0} & 0 \end{bmatrix}$$

and provides triangle $T^{(6)}$ embedded in main plane X_1X_2, so the next step is a positive rotation of the desired angle θ around plane X_1X_2, that is $M_7 = R_{3,4}(\theta)$. The final stage returns the rotation plane to its original position by applying M_8 to M_{13}, which are the inverse matrices of M_6 to M_1, respectively.

Therefore, given a general 4D rotation defined by triangle $T = \mathbf{abc}$ (the rotation planar axis) and an angle θ, every point $\mathbf{x}$ will rotate to the point $\mathbf{x}'$ defined by $\mathbf{x}' = \mathbf{x} \cdot M$, where $M = \prod_{k=1}^{13} M_k$ and it is usually computed in advance.

General *n*D Rotations

For the general case let us rename the elements of matrix $\mathbf{v}^{(k)}$ as

$$\mathbf{v}^{(k)} = \begin{bmatrix} v_{1,1}^{(k)} & v_{1,2}^{(k)} & v_{1,3}^{(k)} & \cdots & v_{1,n}^{(k)} \\ v_{2,1}^{(k)} & v_{2,2}^{(k)} & v_{1,3}^{(k)} & \cdots & v_{2,n}^{(k)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{n-1,1}^{(k)} & v_{n-1,2}^{(k)} & v_{n-1,3}^{(k)} & \cdots & v_{n-1,n}^{(k)} \end{bmatrix}$$

Then, a general nD rotation is a rotation of an angle θ around a general (n-2)-Dimensional subspace. This subspace will be represented by a (n-2)D simplex, whose vertices are $\mathbf{v}^{(0)}$. Then, and proceeding in a similar way as the 3D and 4D cases, we find $M_1 = T(-\mathbf{a})$, where **a** is the first row of $\mathbf{v}^{(0)}$. Then we see that in the k^{th} step, $2 \leq k \leq \frac{n(n-1)}{2}$, we can make $\mathbf{v}_{r,c}^{(k)}$=0 in certain row r and column c, using $M_k = R_{c,c-1}\left(\arctan 2\left(v_{r,c}^{(k-1)}, v_{r,c-1}^{(k-1)}\right)\right)$ in a rather straightforward sequence, which is an adaptation to a more general case of Eq. 3. Therefore in the Algorithm 1 is presented our procedure which we named as the **Aguilera-Perez Algorithm**.

Procedure ComputeM($\mathbf{v}^{(0)}$, θ, n)
$\quad M_1 := T(-\mathbf{a})$
$\quad \mathbf{v}^{(1)} := \mathbf{v}^{(0)} \cdot M_1$
$\quad M := M_1$
$\quad$ k := 1
$\quad$ **for** r := 2 **to** n-1 **do**
$\quad\quad$ **for** c := n **downto** r **do**
$\quad\quad\quad$ k := k+1
$\quad\quad\quad M_k := R_{c,c-1}\left(\arctan 2\left(v_{r,c}^{(k-1)}, v_{r,c-1}^{(k-1)}\right)\right)$
$\quad\quad\quad \mathbf{v}^{(k)} := \mathbf{v}^{(k-1)} \cdot M_k$
$\quad\quad\quad M := M \cdot M_k$
$\quad\quad$ **endFor**
$\quad$ **endFor**
$\quad M := M \cdot R_{n-1,n}(\theta) \cdot M^{-1}$
endProcedure

Algorithm 1. The Aguilera-Perez Algorithm for Computing a General nD Rotation Matrix.

Note that this procedure, at the end of its two loops, produces matrix $\mathbf{v}^{(k)}$ with zeros in its last two columns. This means that simplex $\mathbf{v}^{(0)}$ has been transformed into simplex $\mathbf{v}^{(k)}$ which is embedded in a (n-2)-dimensional subspace, because every vertex has a zero in its last two coordinates. Moreover, $\mathbf{v}^{(k)}$ is a lower triangular matrix, i.e., with n zeros in its first row, n-1 zeros in the second, and so forth, until row n-1 with exactly two zeros.

$$\begin{bmatrix} v_{1,1}^{(0)} & v_{1,2}^{(0)} & v_{1,3}^{(0)} & \cdots & v_{1,n}^{(0)} \\ v_{2,1}^{(0)} & v_{2,2}^{(0)} & v_{2,3}^{(0)} & \cdots & v_{2,n}^{(0)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{n-1,1}^{(0)} & v_{n-1,2}^{(0)} & v_{n-1,3}^{(0)} & \cdots & v_{n-1,n}^{(0)} \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ v_{2,1}^{(k)} & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ v_{n-1,1}^{(k)} & \cdots & v_{n-1,n-2}^{(k)} & 0 & 0 \end{bmatrix}$$

This means that the embedding has taken place in a way that vertex 1 of $\mathbf{v}^{(k)}$ is at the origin, vertex 2 is on main axis X_1, vertex 3 is on main plane X_1X_2, and so forth, until vertex n-1 which is inside an (n-2)-dimensional hyperspace, which is exactly the subspace where $\mathbf{v}^{(k)}$ is embedded. At that point of the algorithm, M is a single matrix that transforms simplex $\mathbf{v}^{(0)}$ into simplex $\mathbf{v}^{(k)}$, that is, $\mathbf{v}^{(k)} = \mathbf{v}^{(0)} \cdot M$.

This leaves just enough room for performing our next step, the desired rotation, which will be a positive rotation of an angle θ, done around the hyperplane $X_1X_2...X_{n-2}$, a (n-2)-dimensional hyperspace, that is, $R_{n-1,n}(\theta)$. The final stage returns the rotation plane to its original position, which can be done with M^{-1}. These last two stages are performed at the end of the algorithm, computing a single matrix M for the whole n-Dimensional rotation.

Therefore, given a general n-Dimensional rotation defined by simplex S (the rotation hyper axis) and an angle θ, every point $\mathbf{x}$ will rotate to the point $\mathbf{x}'$ defined by $\mathbf{x}' = \mathbf{x} \cdot M$, where M is provided by our algorithm.

Note that this procedure acts somewhat like the Gaussian elimination for a system of linear equations.

4. APPLICATIONS

In this section we will present an interesting application of our Aguilera-Perez Algorithm for performing the general n-dimensional rotations under the context of the unraveling of 4D polytopes, specifically the 4D simplex (see Fig. 2).

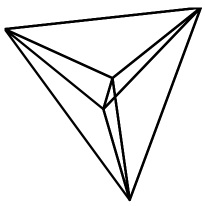


Figure 2. The 4D Simplex.

This topic has been discussed in [Ban96], [Kak94] and [Agu02] (where a methodology for the unraveling of the 4D hypercube is presented). Basically, the unraveling of a n-dimensional polytope implies to embed the (n-1)-dimensional cells that compose its boundary onto a (n-1)-dimensional hyperplane. See Fig. 3.a and Fig 3.b for examples of the unraveling processes of the cube and the tetrahedron (a 3D simplex) respectively.

a)

b)

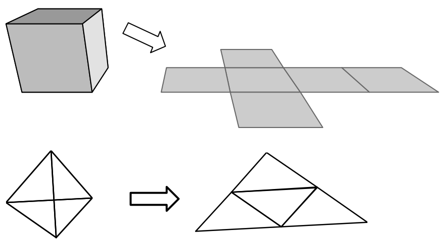


Figure 3. Unraveling the cube (a) and the tetrahedron (b).

Unraveling the 4D Simplex

Because the 4D simplex boundary is composed by five tetrahedrons [Cox84], we can expect that the unravelings of the 4D simplex will be a tetrahedron surrounded by four other tetrahedrons and sharing a face with each one (the unravelings of the tetrahedron are a triangle surrounded by the other three triangles and sharing an edge with each one, see Fig. 2.b). We will refer to the unravelings of the 4D simplex as a ***stellated tetrahedron***.

The coordinates of the 4D simplex to unravel are presented in Table 1 (see [Ban96] for a methodology to get the 4D simplex's coordinates).

Vertex	X_1	X_2	X_3	X_4
0	0	0	0	0
1	1	0	0	0
2	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	0	0
3	$\frac{1}{2}$	$\frac{\sqrt{3}}{6}$	$\frac{\sqrt{6}}{3}$	0
4	$\frac{1}{2}$	$\frac{\sqrt{3}}{6}$	$\frac{\sqrt{6}}{12}$	$\frac{\sqrt{10}}{4}$

Table 1. Coordinates of a 4D simplex.

Analogously to the tetrahedron's unraveling, we have to select the 4D simplex's volume which will be surrounded by the other four volumes as commented before. Moreover, the supporting hyperplane of this selected volume will be the hyperplane to which all the 4D simplex's remaining volumes will be directed to. Observing the 4D simplex's coordinates we can see that four of them present their fourth coordinate value (X_4) equal to zero. This fact represents that one of the simplex's volumes (formed by vertexes 0-1-2-3) has X_4=0 as its supporting hyperplane. Selecting the hyperplane W=0 is useful because one of the volumes is "naturally embedded" in the 3D space and it will not require any transformations.

Now, it is also useful to identify the simplex's volumes through their vertices and to label them for future references. Until now we have one identified volume, it is formed by vertexes 0-1-2-3, and it will be called volume A. See Table 2.

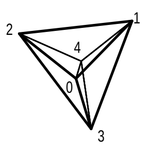
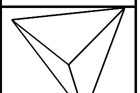
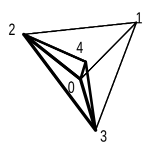
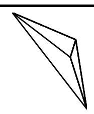
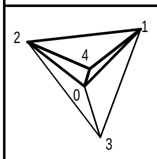
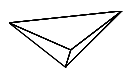
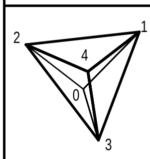
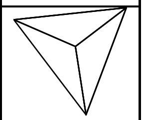
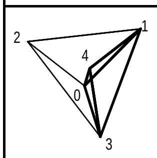
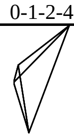
Volume's position in the 4D simplex	Volume and Label	Volume's position in the 4D simplex	Volume and Label
	 Volume A 0-1-2-3		 Volume D 0-2-3-4
	 Volume B 0-1-2-4		 Volume E 1-2-3-4
	 Volume C 0-1-3-4		

Table 2. The 4D simplex's boundary volumes.

We have already described volume A as "naturally embedded" in the 3D space, because it won't require any transformations. Volume A will occupy the central position in the stellated tetrahedron and it will be called the "central volume".

All of the remaining volumes will have face adjacency with the central volume. Due to this characteristic they can easily be rotated toward our space because their rotating plane is clearly identified. Each of these volumes will rotate around the supporting plane of its shared face with central volume. They will be called "adjacent volumes".

Although the rotating planes are clearly identified, the main difference between the 4D simplex and other polytopes' unraveling (as the hypercube, see [Agu02]) is that the rotating planes do not correspond to 4D space main planes (X_1X_2, X_2X_3, X_3X_1, X_1X_4, X_2X_4 and X_3X_4) in the simplex's unraveling.

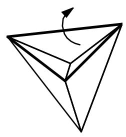
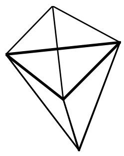
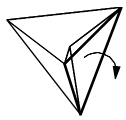
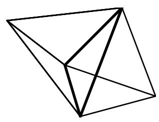
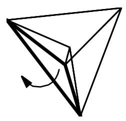
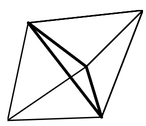
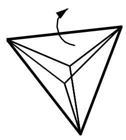
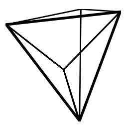
Adjacent volume previous to rotation	**Transformations to apply**	**Position in the stellated tetrahedron after the transformations**
	Volume B Rotation around X_1X_2 plane with $\theta = 104° 29'$	
	Volume C $v^{(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1/2 & \sqrt{3}/6 & \sqrt{6}/3 & 0 \end{bmatrix}$ $\theta = 104° \ 29'$	
	Volume D $v^{(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1/2 & \sqrt{3}/2 & 0 & 0 \\ 1/2 & \sqrt{3}/6 & \sqrt{6}/3 & 0 \end{bmatrix}$ $\theta = -104° \ 29'$	
	Volume E $v^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & \sqrt{3}/2 & 0 & 0 \\ 1/2 & \sqrt{3}/6 & \sqrt{6}/3 & 0 \end{bmatrix}$ $\theta = -104° \ 29'$	

Table 3. Applied transformations to the adjacent volumes

Due to this situation, we will apply the Aguilera-Perez Algorithm with the specific case of $n = 4$, i.e., for performing general 4D-rotations (as a special case, in our example, we have only one volume that will rotate around the X_1X_2 plane, that is, the volume B that shares the face composed by vertices 0, 1 and 2, see Tables 1 and 2). The final objective is to rotate each one of the adjacent volumes an angle of 104° 29' around the supporting plane of the face that share with the central volume. This angle corresponds to the supplement of the simplex's dihedral angle that is 75° 31' [Cox63]. In this way we guarantee that their X_4 coordinate will be equal to zero. The matrices $\mathbf{v}^{(0)}$ (i.e. matrices that contain the points of the shared faces between an adjacent volume and the central volume) and the angles' direction for each adjacent volume are presented in Table 3 (the central volume is also included in each image as a reference for the initial and final position of the volume being analyzed).

Now, all the transformations to unravel the simplex have been determined. To ravel it back, the same process must be applied in an inverse way but only the angles' signs must be changed.

Table 4 presents some snapshots from the 4D simplex's unraveling sequence. From t = 0.00 until t = 0.75, the adjacent volumes (in red) are projected inside the central tetrahedron (in yellow). When t = 1.00, adjacent volumes are projected on the central tetrahedron's faces (they look like planes) -an effect due to the selected 4D-3D projection. From t = 1.25 until t = 5.00, the adjacent volumes are projected outside the central tetrahedron. When t = 3.00 an interesting phenomenon arises, the projected volumes form an hexadron (a cube) –again, an effect due to the selected projection. When t = 5.00 the stellated tetrahedron is finally composed.

5. CONCLUSIONS

In this work we have presented the Aguilera-Perez Algorithm which specifies a methodology to perform general *n*-dimensional rotations. Such methodology coincides with the well known 2D and 3D cases. Moreover, we have discussed an application in the context of the 4D simplex's unraveling.

Future work

Currently we identify two lines of research which are closely related with the current presented work:

- To propose methodologies for general *n*-dimensional rotations based in octonions or in a more general way, based in 2^k-nions (whose theoretical basis are discussed in [Con03]). As a precedent we mention the well known theory related to the representation of general 3D rotations using quaternions (see [Hea96] for example).
- To define the procedures to unravel *n*D polytopes such as the cross-polytope (the analogous to the octahedron in the 3D space, see [Gru03] and [Mcm02] for more details) whose positions can be arbitrary in the *n*D space.

Moreover, our methodology is currently being used as auxiliary didactic material at the Universidad de las Américas-Puebla considerably improving the teaching/learning processes related to *n*-Dimensional Euclidean Spaces.

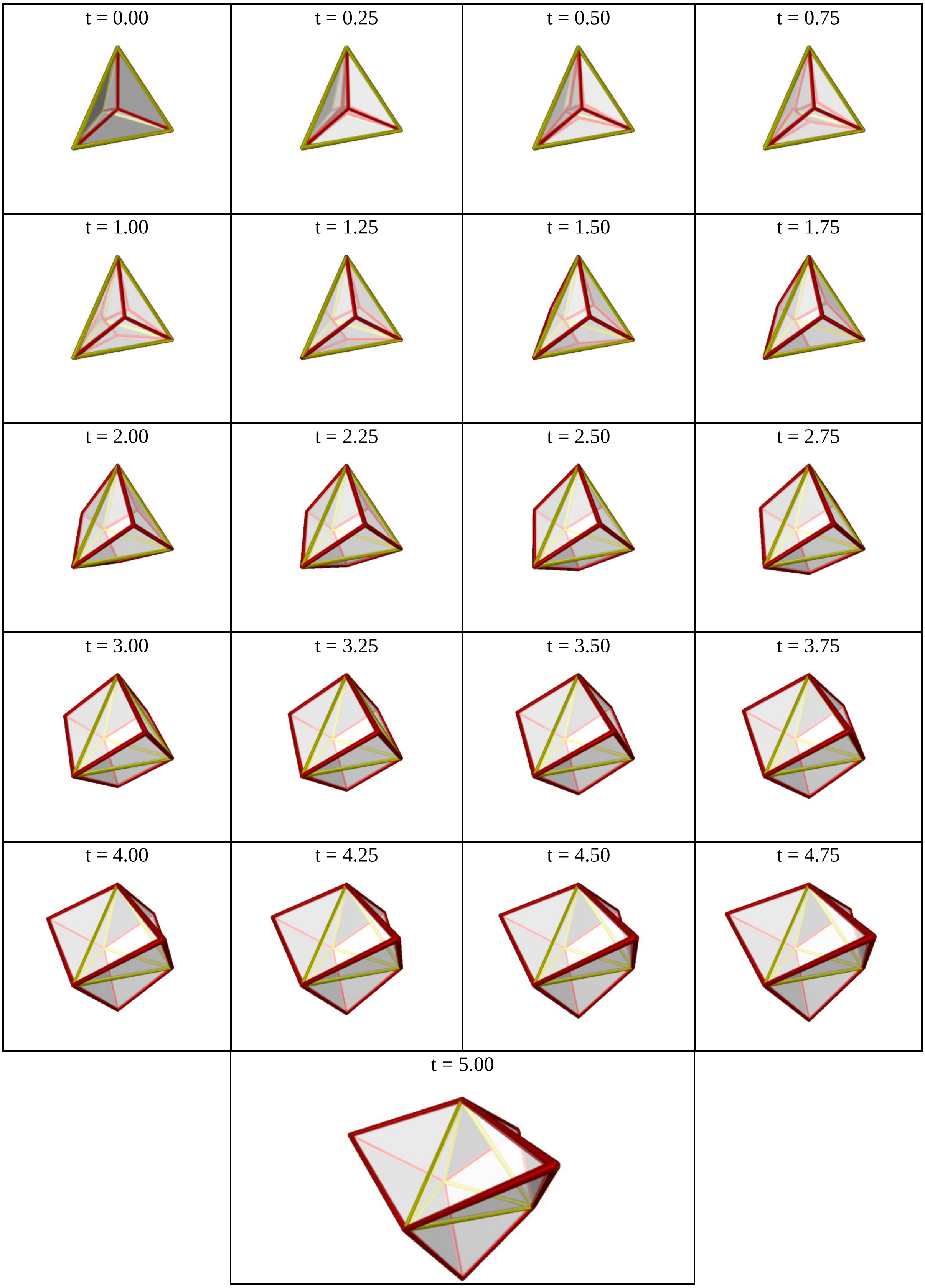


Table 4. Unraveling the 4D simplex.

6. REFERENCES

[Agu02] Aguilera, A., and Pérez-Aguila, R. A Method For Obtaining The Tesseract By Unraveling The 4D Hypercube. Journal of the 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2002. Volume 10, Number 1, pp. 1-8, February 4 to 8, 2002. Plzen, Czech Republic. ISSN: 1213-6972.

[Ban92] Banks, D. Interactive Manipulation and Display of Two-Dimensional Surfaces in Four Dimensional Space. Proceedings of the 21st annual conference on Computer graphics, July 24 - 29, 1994, Orlando, FL USA, Pages 327 - 334.

[Ban96] Banchoff, Thomas F. Beyond the Third Dimension. Scientific American Library, 1996.

[Con03] Conway, J., and Smith, D. On Quaternions and Octonions. A K Peters, Ltd., 2003.

[Cox63] Coxeter, H.S.M. Regular Polytopes. Dover Publications, Inc., New York, 1963.

[Cox84] Coxeter, H.S.M. Fundamentos de Geometría, Editorial Limusa, 1984.

[Duf94] Duffin, K., and Barnett, W. Spiders: A new user interface for rotation and visualization of n-dimensional points sets. Proceedings of the 1994 IEEE Conference on Scientific Visualization, October 17 to 21, 1994, Washington, D.C. USA, Pages 205-211.

[Gru03] Grünbaum, B. Convex Polytopes. Second Edition prepared by Kaibel, V., Klee, V., and Ziegler, M. Graduate Texts in Mathematics, Volume 221. Springer-Verlag New York, Inc., 2003.

[Hea96] Hearn, D., and Baker, P. Computer Graphics, C Version. Second Edition. Prentice Hall, 1996.

[Hol91] Hollasch, S.R. Four-Space Visualization of 4D Objects. Arizona State University, 1991, Thesis for the Master of Science Degree.

[Kak94] Kaku, M. Hyperspace: A Scientific Odyssey Through Parallel Universes, Time Warps, and the Tenth Dimension. Oxford University Press, 1994.

[Mcm02] McMullen, P., and Schulte, E. Abstract Regular Polytopes. Encyclopedia of Mathematics and Its Applications, Volume 92. Cambridge University Press, 2002.