



HAL
open science

Geometric transformation of a constrained object using a non-Cartesian method

Mireille Moinet, Philippe Serré

► **To cite this version:**

Mireille Moinet, Philippe Serré. Geometric transformation of a constrained object using a non-Cartesian method. *International Journal of Product Development*, 2014, 19 (1/2/3), <10.1504/IJPD.2014.060047>. <hal-01903195>

HAL Id: hal-01903195

<https://hal.science/hal-01903195v1>

Submitted on 8 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Geometric transformation of a constrained object using a non-Cartesian method

Mireille Moinet* and Philippe Serré

Supmeca-LISMMA, EA 2336,
Mechanical and Manufacturing Engineering School,
3 rue Fernand Hainaut, 93407 Saint Ouen cedex, France
Email: mireille.moinet@supmeca.fr
Email: philippe.serre@supmeca.fr
*Corresponding author

Abstract: This paper presents a new approach to transform a constrained geometric object. The geometric model is based on the vectorisation of the object's shape. Thus, elementary geometric entities and constraints are represented by a set of vectors or bivectors in order to establish the object specifications. The metric tensor is the mathematical tool used to define the vector space generated by the set of vectors and bivectors. The transformation of the initial object is realised thanks to a transformation matrix associated with a connection matrix. Based on the point of displacements of the initial object, this new method gives the final object satisfying the geometric specifications required by the designer.

Keywords: geometric constraint; non-Cartesian geometry; tensorial model; perturbation; declarative modelling.

1 Introduction

In this paper, we focus on geometric modellers. Various types of geometric modellers can be found: parametric modellers, which include B-REP (Boundary Representation), CSG (Constructive Solid Geometry) and hybrids; variational modellers have a higher level of abstraction.

Declarative modelling is a well-adapted solution to declare and specify geometric objects and constraints in computer-aided design systems. The aim of declarative modelling in mechanical design is to provide high-level tools and methods to assist the designer to solve technical problems and find suitable solutions.

In such modellers, the elementary objects and related constraints are described rather than the objects creation procedure. This type of approach can be found today in modules of some 2D sketcher software (Lesage, 2002) and/or for defining part assembly. The systems of geometric constraints to be solved in CAD are often very large. Consequently, they are often decomposed. The decomposition methods reduce the systems of constraints to be solved into several sub-systems easier to solve. All solutions found for the sub-systems are merged in order to provide solutions of the initial system.

There are two ways for decomposing a system of constraints according to the solving processing phase. First, during the problem analysis, the structural decomposition methods help to organise the data of the problem to build a more robust solution. These methods are also called directed graph methods because they use the properties of graph to identify the independent soluble subsystems. They can be decomposed into two sub-methods: the structural methods, such as downward approaches of Owen (1991) and the ascending approaches of Hoffmann et al. (1997) and Bouma et al. (1995); and the rules-based methods. Second, in the digital resolution phase, which occurs after the generation of the equations, the system decomposition consists in an equational decomposition of the system to be solved. This paper focuses more on the resolution of equations than on the reducing the complexity of the system of constraints.

This research uses the variational model already presented by Serré (2000) and Moinet and Serré (2011). This model allows describing 2D geometric objects using the point and the straight line as basic geometric entities. For modelling 3D objects, the basic geometric entities are the point, the straight line, the plane, the cylinder and the sphere. To complete any object modelisation, geometric constraints are added between basic entities. These constraints may be one of the following types: coincidence, orthogonality, parallelism, distance and angle (Rivière, 1993; Clément et al., 1994). The geometric model is based on a vectorial representation of the geometric entities and the constraints. Section 2 presents briefly the vectorisation of the geometry and the non-Cartesian model. Mathematical tools used are also detailed in this section. Then, Section 3 presents the proposed approach based on the transformation of the initial geometry thanks to a connection matrix and a perturbation matrix. This method is particularly appropriate for CAD as the user always starts by 'drawing' an initial shape that is gradually modified to obtain the desired object. Section 4 details the setting equations specifications. These equations involve geometrical specifications like angle line–line equations, angle plane–plane equations, length equations and area equations. Each type of specification imposed by the designer will be reflected in an algebraic equation determined from the metric tensor. Section 5 describes the iterative method used to solve the algebraic system and an application is detailed to illustrate the method. Finally, Section 6 permits to discuss on the method whereas Section 7 concludes this paper.

2 Geometric model

2.1 Vectorisation of the geometry

First step of the declarative modelling is the *description*. This step defines the interaction language that allows the designer to provide properties and relation between the design objects.

The geometric entities handled are points, straight lines and planes – termed the MRGEs for Minimum Geometric Reference Elements (Rivière, 1993; Clément et al., 1994). Authors apply a related vectorial model to this data model. As a result:

- the point is represented by a point;
- the straight line by a vector and a point;
- the plane by a vector and a point.

Let us take an example, given e_1 and e_2 two elements in a vectorial space E . They are chosen as directional vectors of the straight lines running through points M and N .

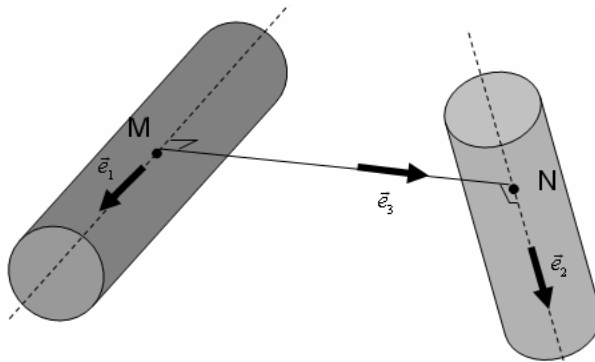
Add a new unit vector to this space: e_3 .

Here l_3 is the length between the two points M and N , thus give:

$$u_{mn} = l_3 e_3$$

The angle between the straight lines running through e_1 and e_2 will be written as α_{12} and can be obtained by the scalar product of these two unit vectors (Figure 1).

Figure 1 Vectorisation



Using this model allows to represent the geometry with vectors. The metric tensor is the mathematical tool chosen to translate the vector relations in algebraic relations.

2.2 Tensorial modelling

The propose of the paper is to use a non-Cartesian model to characterise geometric objects. Contrary to most of geometric models defined by constraints where all elements are represented in a Cartesian frame and the coordinates of all characteristic points are the unknowns of the system of algebraic equations, the non-Cartesian models use

mathematical tools to write algebraic equations between intrinsic characteristic elements of the geometrical object. For instance, the determinant of Cayley–Menger is the base of the method used by Michelucci and Fofou (2004, 2006) to calculate distance between points. Hestenes represents geometry with a unified framework that is geometric algebra. This recent mathematical tool is based on Clifford Algebra (Hestenes, 1966; Hestenes, 1986; Hestenes et al., 1984; Serré et al., 2011).

The approach based on tensor uses a set of vectors describing the objects and constraints. The metric tensor of this set of vectors is constructed. This metric tensor fully defines the metrics of the object and does not depend on a particular Cartesian frame.

An added advantage of this approach is the possibility of ensuring specification consistency by verifying the mathematical properties of the metric tensor (symmetrical, semi-definite positive ...). For example, it can be easier to detect whether the problem has a solution or not, by calculating specific determinants.

2.1.1 Vectors

Any geometry can be simplified as points and vectors. These are the central elements of our modelling. This section recalls some important properties of vectors. More details are found in the work of Moinet (2008).

Any vector x in a vectorial space with n dimensions in which a base e_1, e_2, \dots, e_n is defined as:

$$x = \sum_{i=1}^n e_i x^i$$

where x^1, x^2, \dots, x^n are the *contravariant components* of the vector. In another base E_1, E_2, \dots, E_n it will have the coordinates X^1, X^2, \dots, X^n .

Authors will subsequently use the Einstein convention to simplify writing: if, in an expression, the same index is repeated at the top and bottom, the summation of this index is implicit.

In a Euclidean space, the scalar product of two vectors x and y is written as:

$$x, y = \sum_{i=1}^n \sum_{j=1}^n e_i, e_j x^i y^j$$

The metric tensor is defined as $G_{ij} = e_i, e_j = G_{ji}$

With the metric tensor, the *covariant components* of a vector are defined as:

$$x_i = e_i, x = e_i, e_j x^j = G_{ij} x^j .$$

2.1.2 Bivectors

According to Cartan (1946), a *bivector* is defined as follows: bivector calls the object formed by two vectors x and y arranged in some order, noted $[xy]$. This definition only makes sense if the equality of two bivectors is defined.

Let P is the first bivector defined by vectors x and y ; and the second bivector Q defined by the vectors u and v . z is a vector perpendicular to the plane of the first bivector by:

$$\begin{cases} z_1x^1 + z_2x^2 + \dots + z_nx^n = 0 \\ z_1y^1 + z_2y^2 + \dots + z_ny^n = 0 \end{cases}$$

Note that the components z_1, z_2, \dots, z_n are connected by the following equation, obtained by removing z_1 :

$$\begin{aligned} z_2(x^1y^2 - x^2y^1) + z_3(x^1y^3 - x^3y^1) \\ + \dots + z_n(x^1y^n - x^ny^1) = 0 \end{aligned}$$

Let set P^{ij} as:

$$P^{ij} = x^i y^j - x^j y^i$$

and

$$Q^{ij} = u^i v^j - u^j v^i$$

P^{ij} are the coordinates of the P bivector. P_{ij} called the *covariant components* of the bivector and P^{ij} is the *contravariant components*.

Note that $P^{ij} = -P^{ji}$ (id. $P_{ij} = -P_{ji}$).

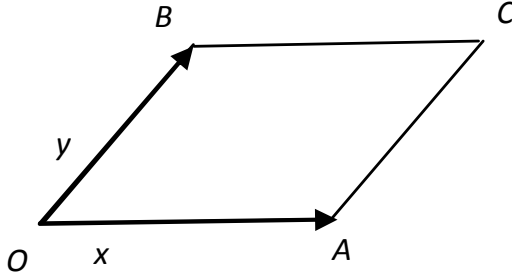
Calculate the area A of a parallelogram OACB associated with the bivector P , as shown in Figure 2:

$$\begin{aligned} A^2 &= \|OA\|^2 \|OB\|^2 \sin^2(\widehat{AOB}) \\ &= \|OA\|^2 \|OB\|^2 - (OA \cdot OB \cos(\widehat{AOB}))^2 \\ &= x^2 y^2 - (x \cdot y)^2 \\ &= \begin{vmatrix} x^2 & x \cdot y \\ y \cdot x & y^2 \end{vmatrix} \\ &= \begin{vmatrix} x_i & x_j \\ y_i & y_j \end{vmatrix} \\ &= P_{ij} x^i y^j \end{aligned}$$

As $P_{ij} x^i y^j = P_{ji} x^j y^i = -P_{ij} x^j y^i$, we deduce that

$$A^2 = \frac{1}{2} P_{ij} (x^i y^j - x^j y^i) = \frac{1}{2} P_{ij} P^{ij} \quad (1)$$

Figure 2 Bivector



Relations between contravariant components P^{ij} and covariant components P_{ij} of bivector P are the following:

$$P_{ij} = \begin{vmatrix} x_i & x_j \\ y_i & y_j \end{vmatrix} = \begin{vmatrix} g_{ih}x^h & g_{jk}x^k \\ g_{ih}y^h & g_{jk}y^k \end{vmatrix} = g_{ih}g_{jk}P^{hk}$$

Formula (1) is rewritten as:

$$\begin{aligned} A^2 &= \frac{1}{2}P_{ij}P^{ij} = \frac{1}{4}P^{ij}(P_{ij} - P_{ji}) = \frac{1}{4}P^{ij}(g_{ih}g_{jk}P^{hk} - g_{jh}g_{ik}P^{hk}) \\ &= \frac{1}{4}(g_{ih}g_{jk} - g_{jh}g_{ik})P^{ij}P^{hk} \end{aligned}$$

This formula is similar to that giving the square of the length of a vector with the difference: a bivector has $\frac{n(n-1)}{2}$ components P^{ij} and the coefficients of the quadratic form are for two composed indices ij and hk .

Like the metric tensor with simple indices, note that the coefficients are always symmetrical:

$$G_{ij,hk} = G_{ih}G_{jk} - G_{ik}G_{jh} \quad (2)$$

Any bivector can be represented by a vector of $\frac{n(n-1)}{2}$ dimensions Euclidean space whose metric is defined by the coefficients of $G_{ij,hk}$.

Scalar product: The above results are used to define the dot product of two bivectors by one of two expressions:

$$\frac{1}{2}P_{ij}Q^{ij} = \frac{1}{2}P^{ij}Q_{ij}$$

This scalar product has a numerical value independent of the choice of coordinates. In the Euclidean space, the dot product of two bivectors equals the product of their measures by the cosine of the angle of their planes (analogous to the scalar product of two vectors).

Angle φ is defined as angle between plans of two bivectors P and Q , by formula:

$$\cos \varphi = \frac{P \cdot Q}{\sqrt{P^2} \sqrt{Q^2}}$$

3 Geometric transformation

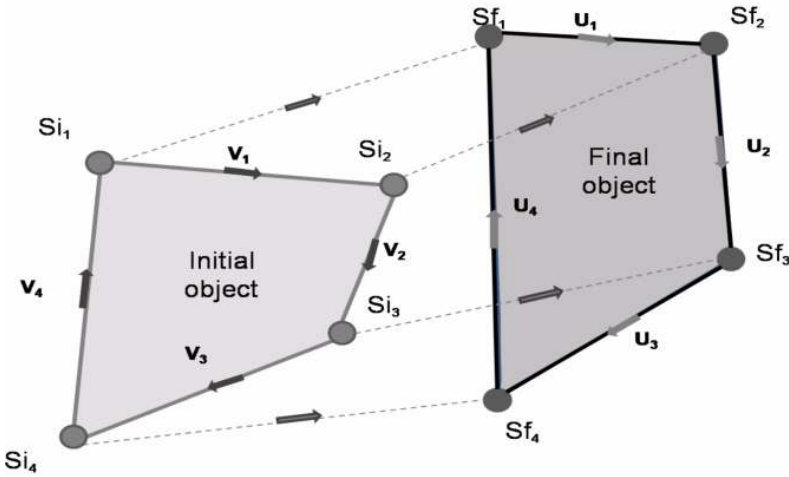
3.1 The connection matrix

A geometric object is represented by a list of vectors. Given V a set of n non-normed vectors as: $V = (v_1, v_2, \dots, v_n)$. Each vector is an oriented bipoint. For example, v_1 has Si_1 as origin and Si_2 as target (see Figure 3). The matrix $G(V)$ is defined as:

$$G_{ij}(V) = v_i \cdot v_j$$

This matrix is also noted G_{ij} .

Figure 3 Vertex displacement



The two objects are represented by two different matrices which have the same size.

The variation $Sf_i - Si_i$ corresponds to the vertex displacement. Here Si_i and Sf_i are vectors which are the position of the vertex i , before and after displacement, respectively.

An incidence matrix C makes the relation between each points of an object and vectors. It is an $n \times m$ matrix where n and m are the number of vectors and vertices, respectively, such that $C_{ij} = -1$ if the vector v_i leaves vertex Si_j , 1 if it enters vertex Si_j and 0 otherwise.

Then, the edge relation for an initial object is:

$$v_i = C_{ij} \cdot Sj \tag{3}$$

The edge relation for a final object is:

$$u_i = C_{ij} S_j \quad (4)$$

3.2 The perturbation of the initial object model

If V is a set of vectors of ranks r and U is a set of vectors obtained after a transformation from an initial state to a final one, we express the relation between these two states.

Note the vertex variation:

$$Sf_i - Si_i = \Omega_{ij} v_j \quad (5)$$

This matrix Ω is called the perturbation matrix from the initial state of the geometry to the final one. It is an $n \times m$ matrix.

Thanks to equation (5), the relation between initial vectors and final one is uncomplicated:

$$\begin{aligned} u_i &= C_{ij} (\Omega_{jk} v_k + Si_j) \\ u_i &= v_i + C_{ij} \Omega_{jk} v_k \\ u_i &= (\delta_{ik} + C_{ij} \Omega_{jk}) v_k \end{aligned} \quad (6)$$

where δ_{ik} represents the Kronecker symbol which equals 1 if $i = k$ and, if not 0.

The purpose is to find a transformation of an object from its initial shape to its final one. H matrix characterises the position of all geometry's points. Indeed

$$\begin{aligned} H_{ij} &= u_i, u_j \\ H_{ij} &= (\delta_{ik} + C_{ij} \Omega_{jk}) v_k, (\delta_{jm} + C_{jl} \Omega_{lm}) v_m \\ H_{ij} &= (\delta_{ik} + C_{ij} \Omega_{jk}) (\delta_{jm} + C_{jl} \Omega_{lm}) v_k, v_m \\ H_{ij} &= (\delta_{ik} + C_{ij} \Omega_{jk}) (\delta_{jm} + C_{jl} \Omega_{lm}) G_{km} \end{aligned}$$

This expression writes in the matrix form is:

$$H = (I + C\Omega)G'(I + C\Omega) \quad (7)$$

Note T is the vectorial transformation $I + C\Omega$. Thanks to the rank property:

$$\text{rank}(H) \leq \min(\text{rank}(T), \text{rank}(G))$$

H is a matrix of rank r .

Formula (7) above represents the variation of the geometric object. The elements of the perturbation matrix Ω are the unknowns of the problem.

So far, the metric tensor G is only used to characterise the position and orientation of all vectors of the initial sketch. A new mathematical tool is defined: the GG tensor that is linked to G by the transfer equation coming from equation (2):

$$GG_{ij,hk} = \begin{vmatrix} G_{ih} & G_{jh} \\ G_{ik} & G_{jk} \end{vmatrix} \quad (8)$$

This relation is called the transfer relationship between G and GG .

4 Establishing the equations of the problem

The geometric constraints specified by the user lead to a set of equations. These equations can be classified into four types: depending on whether they come from length, angle between vectors and angle between bivectors or area.

The first two types of equations were already discussed in a previous paper of Moinet and Serré (2011). Thus, Section 4.1 and 4.2 recall the significant results.

Later in the document, element ij of matrix $G(V)$ will be written as G_{ij} (i.e. for the final matrix, noted H_{ij}).

4.1 Length equations

By definition, an element of matrix diagonal G_{ii} (H_{ii} for the final one) represents the squared length of an initial vector (final resp.). The length specification of the vector v_i imposed by the designer is denoted L_i . Thus,

$$\sqrt{H_{ii}} = L_i \quad (9)$$

Given that each final tensor element is expressed as follows, according to the elements of perturbation matrix Ω :

$$H_{ij} = \sum_{p=1}^n \sum_{q=1}^n (\delta_{ip} + x_{ip}) G_{pq} (\delta_{jq} + x_{jq}) \quad (10)$$

With x_{ip} the ip -th element of the product matrix $C\Omega$ and δ_{ip} the Kronecker symbol which equals 1 if $i = p$ and, 0 otherwise.

By developing this last expression to the first order, we obtain:

$$H_{ij} \cong G_{ij} + \sum_{p=1}^n (x_{ip} G_{pj} + G_{ip} x_{jp}) \quad (11)$$

In particular, for the elements of the diagonal:

$$H_{ii} \cong G_{ii} + 2 \sum_{p=1}^n x_{ip} G_{pi} \quad (12)$$

And by developing equation (9), it becomes:

$$\frac{1}{\sqrt{G_{ii}}} \sum_{p=1}^n x_{ip} G_{pi} \cong L_i - \sqrt{G_{ii}} \quad (13)$$

Terms L_i , G_{pi} and G_{ii} are known. The unknowns of this equation are x_{ip} .

4.2 Line–line angle equations

By definition, an element G_{ij} represents the scalar product between vectors v_i and v_j .

The designer requires the cosine of the angle between v_i and v_j to satisfy specification α_{ij} , resulting in the following equation:

$$H_{ij} = \sqrt{H_{ii}} \sqrt{H_{jj}} \cos(\alpha_{ij}) \quad (14)$$

Re-write equation (14) to move all the terms of H to the left of the equals sign gives:

$$H_{ij} (H_{ii})^{-1/2} (H_{jj})^{-1/2} = \cos(\alpha_{ij})$$

By developing H_{ij} , H_{ii} and H_{jj} using equation (11) and by linearisation to the first order, expression of an angular specification translates as follows:

$$\sum_{p=1}^n \left(G_{ip} - \frac{G_{ij}}{G_{jj}} G_{pj} \right) x_{jp} + \left(G_{pj} - \frac{G_{ij}}{G_{ii}} G_{pi} \right) x_{ip} \cong \sqrt{G_{ii}} \sqrt{G_{jj}} \cos(\alpha_{ij}) - G_{ij} \quad (15)$$

Terms G_{ip} , G_{pj} , G_{ii} , G_{ij} , G_{jj} and $\cos(\alpha_{ij})$ are known. The unknowns are elements of $C\Omega$, x_{jp} and x_{ip} .

4.3 Area equations

As mentioned in the paper, the dot product of two bivectors equals the product of their measures by the cosine of the angle of their planes. Thus

$$HH_{ij,kl} = A_{ij} A_{kl} \cos\left(\widehat{[ij]}, \widehat{[kl]}\right) \quad (16)$$

where $HH_{ij,kl}$ expresses the final position of the bivectors $[v_i v_j]$ and $[v_k v_l]$. A_{ij} (resp. S_{kl}) is the surface area swept by the bivector $[v_i v_j]$ (resp. $[v_k v_l]$)

$$HH_{ij,ij} = A_{ij}^2$$

The area A_{ij} can be specified by the following relation:

$$A_{ij} = \sqrt{HH_{ij,ij}} \quad (17)$$

where A_{ij} is the value of the area specified by the designer.

Thanks to equation (8),

$$HH_{ij,ij} = \begin{vmatrix} H_{ii} & H_{ij} \\ H_{ji} & H_{jj} \end{vmatrix} = H_{ii} H_{jj} - H_{ij}^2$$

Involving a second-order Taylor expansion in the root, it can be written as:

$$A_{ij} \cong \frac{1}{\sqrt{GG_{ij,ij}}} \left(GG_{ij,ij} + \begin{pmatrix} G_{jj} \sum_{p=1}^n x_{ip} G_{pi} \dots \\ + G_{ii} \sum_{p=1}^n x_{jp} G_{pj} \dots \\ - G_{ij} \sum_{p=1}^n x_{ip} G_{pj} + G_{ip} x_{jp} \end{pmatrix} \right) \quad (18)$$

where $GG_{ij,ij} = G_{ii}G_{jj} - G_{ij}^2$.

Terms A_{ij} , G_{pj} , G_{pi} , G_{ii} , G_{jj} and G_{ij} are known. The unknowns of this equation are the elements x .

4.4 Plane-plane angle equations

The previous section has shown the way to specify the area of a surface generated by a bivector. Using equation (16) also allows the specification of the angle between two planes [see equation (19)]:

Note $\Phi_{ij,kl}$ is the angle between two bivectors $[v_i v_j]$ and $[v_k v_l]$

$$\cos(\Phi_{ij,kl}) = HH_{ij,kl} \cdot A_{ij}^{-1} A_{kl}^{-1} \quad (19)$$

Moreover, the final position of the bivectors is characterises by tensor HH as:

$$HH_{ij,kl} = \begin{vmatrix} H_{ik} & H_{il} \\ H_{jk} & H_{jl} \end{vmatrix} = H_{ik} H_{jl} - H_{jk} H_{il} \quad (20)$$

Thanks to equation (20), the relationship between the final tensor HH given information between bivectors and the tensor H given information between vectors is expressed.

After calculus,

$$HH_{ij,kl} \cong GG_{ij,kl} \left(1 + \frac{1}{GG_{ij,kl}} \begin{pmatrix} H_{jl} \sum_{p=1}^n x_{ip} H_{pk} + H_{ip} x_{kp} \dots \\ + H_{ik} \sum_{p=1}^n x_{jp} H_{pl} + H_{jp} x_{lp} \dots \\ - H_{il} \sum_{p=1}^n x_{jp} H_{pk} + H_{jp} x_{kp} \dots \\ - H_{jk} \sum_{p=1}^n x_{ip} H_{pl} + H_{ip} x_{lp} \end{pmatrix} \right) \quad (21)$$

where

$$GG_{ij,kl} = G_{ik}G_{jl} - G_{jk}G_{il} \quad (22)$$

By replacing $HH_{ij,kl}$ in equation (19)

$$\cos(\Phi_{ij,kl}) \cong \frac{GG_{ij,kl}}{\sqrt{GG_{ij,ij}}\sqrt{GG_{kl,kl}}} \left(1 + \sum_{p=1}^n \left(\frac{GG_{pj,kl}}{GG_{ij,kl}} + \frac{GG_{jp,ij}}{GG_{ij,ij}} \right) x_{ip} + \dots \right. \\ \left. + \sum_{p=1}^n \left(\frac{GG_{ij,pl}}{GG_{ij,kl}} + \frac{GG_{pl,lk}}{GG_{lk,lk}} \right) x_{kp} + \dots \right. \\ \left. + \sum_{p=1}^n \left(\frac{GG_{ij,kp}}{GG_{ij,kl}} + \frac{GG_{pk,kl}}{GG_{lk,lk}} \right) x_{lp} + \dots \right. \\ \left. + \sum_{p=1}^n \left(\frac{GG_{ip,kl}}{GG_{ij,kl}} + \frac{GG_{ji,ip}}{GG_{ij,ij}} \right) x_{jp} \right) \quad (23)$$

In this expression, $\cos(\Phi_{ij,kl})$ and terms of GG are known. The unknowns are $x_{ip}, x_{kp}, x_{lp}, x_{jp}$, $p \in [1, \dots, n]$ corresponding to elements of $C\Omega$ matrix. Indices i, j, k and l are done.

New equations of specifications are now known. The use of bivectors allows specifying angles between planar faces using the same mathematical tool. The variables of these equations are the elements of the perturbation matrix $C\Omega$. The last step is to solve them.

5 Solving and applications

5.1 Geometric problem solving

The system to be solved consists of non-linear relations (9), (14), (17) and (19). The known terms are the elements of the initial tensors G and GG , line–line angle specifications α , plane–plane angle specification ϕ , length specifications L or area specification A .

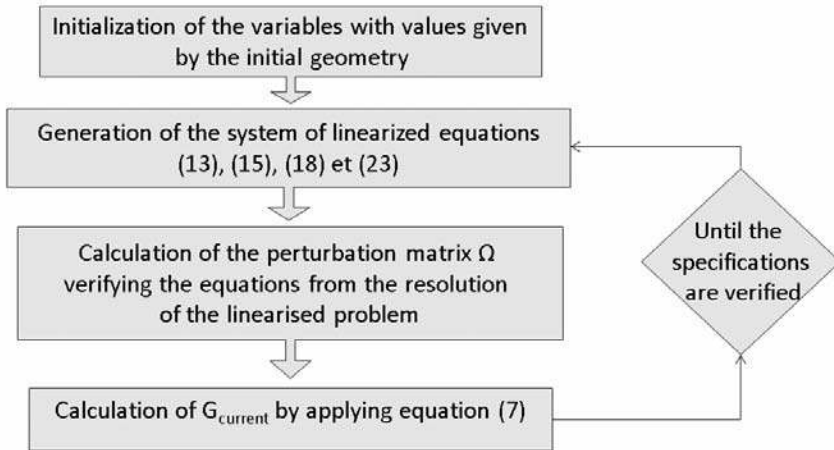
To solve the problem, elements of the perturbation matrix $C\Omega_{ij}$ have to be found. Matrix Ω defines the transformation of the set of vectors from the initial to the final state.

In this section, we are not going to attempt to describe in detail the method used to solve the equations. Previously, this study was carried out by Moinet and Serré (2011) for a different set of equations but well adapted to the new problem presented in this paper.

The method most frequently used to solve this type of system, ($F(X) = 0$), is that of Newton–Raphson. However, the algorithm implemented is a variation of the Newton–Raphson method which is presented in the next paragraph. Finally, this solution is illustrated by an example.

The algorithm can be represented in the form of the diagram shown in Figure 4.

Figure 4 Diagram of the implemented algorithm



5.2 Application on 3D well-constrained example

To illustrate this method, one example in 3D is fully processed: a tetrahedron. Every step of the problem is successively described.

The tetrahedron example will illustrate the four constraints: length equation, line–line angle equation, area equations and plane–plane angle equations.

Tetrahedron is composed of points A, B, C and D. It has six edges: AB, AC, AD, BC, BD and CD and four planar faces ABC, ADB, ACD and BDC.

Figure 5 Text file proposed

<pre> #IVERTEX4 Point:A(22.420713;39.068088;0.) Point:B(96.527153;-28.075308;0.) Point:C(-42.065776;-8.276318;0.) Point:D(50.415214;10.09387;48.433486) #EDGE6 Edge:AB(-A;B) Edge:AC(-A;C) Edge:AD(-A;D) Edge:BC(-B;C) Edge:BD(-B;D) Edge:CD(-C;D) #IFACE4 Face:ABC(AB;BC;-AC) Face:ADB(AD;-BD;-AB) Face:ACD(AC;CD;-AD) Face:BDC(BD;-CD;-BC) </pre>	<pre> #!ANGULAR_SPECIFICATION_11 Angle:a1(AB;AC)=0.0 #!LENGTH_SPECIFICATION5 Longueur:l2(AC)=85 #!ANGULAR_SPECIFICATION_22 Angle:a2(ABC;BDC)=-0.633920960614 Angle:a3(ADB;BDC)=-0.63715499106 #!AREA_SPECIFICATION2 Area:S1(ABC)=4250 Area:S2(ADB)=2437.2 </pre>
---	---

Each face is characterised by a bivector:

- Face ABC is represented by $[\overline{AB}, \overline{AC}]$,
- Face ADB is represented by $[\overline{AD}, -\overline{BD}]$,
- Face ACD is represented by $[\overline{AC}, \overline{CD}]$,
- Face BDC is represented by $[\overline{BD}, -\overline{CD}]$.

To specify this object, the user described in the text file presented in Figure 5 all the entities needed for its construction. The bivectors are constructed from the first two vectors making up the face and are reported in the text file.

5.2.1 Specifications

In order to express plane–plane angular specifications or area specifications with the keywords ‘ANGULAR SPECIFICATION’ and ‘AREA SPECIFICATION’ are introduced in the specification text file (see Figure 5). For the angular specification, the faces are designated by their name and the value specified is the cosine of the angle between the two faces. For the area specification, the face is also designated by its name and the value specified is the desired surface.

Therefore, the text file presented in Figure 5 specifies:

- Angle between two edges
 - Angle between edge AB and edge AC is 90° ($\cos 90^\circ = 0$)
- Edge length
 - $AC=85$
- 2 angles between planes:
 - Angle between ABC and BDC faces is 129.34 ($\cos 129.34^\circ = -0.6339$)
 - Angle between ADB and BDC faces is 129.58 ($\cos 129.34^\circ = -0.6372$)
- Areas:
 - Area of the surface ABC is 4250
 - Area of the surface ADB is 2437.2 .

5.2.2 Results

The iterative resolution converges in seven iterations and 0.1320 s. These specifications are met close to 10^{-12} length units.

The obtained tensors G and GG are required to visualise the result geometry which are shown in Figures 6 and 7.

The order of the vectors is as follows: AB, AC, AD, BC, BD and CD .

Both framed values give as a numerical value of $AC = 85$ and the angle between AB and AC is 90° ($\cos(AB, AC) = 0, AC = \sqrt{7225}$).

Figure 6 The final metric tensor associated with the example

$G_n =$

$1.0e+004 *$

0.2500	-0.0000	0.1086	-0.2500	-0.1414	0.1086
-0.0000	0.7225	-0.1881	0.7225	-0.1881	-0.9106
0.1086	-0.1881	0.2848	-0.2967	0.1762	0.4729
-0.2500	0.7225	-0.2967	0.9725	-0.0467	-1.0192
-0.1414	-0.1881	0.1762	-0.0467	0.3176	0.3643
0.1086	-0.9106	0.4729	-1.0192	0.3643	1.3835

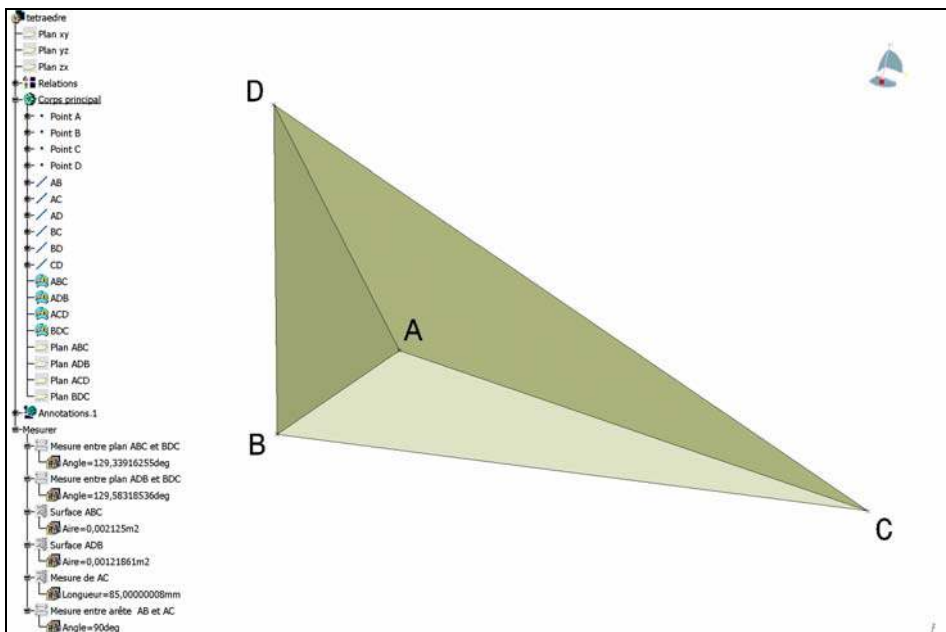
Figure 7 The final metric tensor GG associated with the example

$GG =$

$1.0e+009 *$

1.806250000000000	0.000000000454000	-0.000000000447300	-0.000000000633900
0.000000000454000	0.593994384000000	-0.000000000203100	-0.000000000637200
-0.000000000447300	-0.000000000203100	1.703625625000000	-0.000000000312700
-0.000000000633900	-0.000000000637200	-0.000000000312700	3.066612129000000

Figure 8 Final object (see online version for colours)



The displayed GG is the unit bivectorial tensor, which directly provides the cosine of the angle between two planes. The order of the bivectors is as follows:

$$\left[\overline{AB}, \overline{AC} \right], \left[\overline{AD}, -\overline{BD} \right], \left[\overline{AC}, \overline{CD} \right] \text{ and } \left[\overline{BD}, -\overline{CD} \right]$$

The calculation of the areas (square root of the diagonal elements) confirms the accuracy of results.

Figure 8 shows the result of the last calculation in the CATIA® environment. This model allows verifying, one more time that specifications made by the designer are met.

6 Discussion

This original method to transform an object from its initial shape to its final one gives identical results as the method presented in the previous paper of Moinet and Serré (2011). However, the use of the connection matrix reduces the number of equations. Indeed, the closure equations are useless with this new model. A gain of one iteration (about five with the last method) is observed in the resolution. The original approach permits to treat length and angle between edges. This new approach is also well-adapted to higher level specifications like area or angular between two planar faces.

7 Conclusion

A vectorial representation is useful for characterising objects and can be used to transform an object from an initial state to a final one.

This research proposes a new method, based on point displacement, to compute a final object satisfying all the specifications required by the designer. More specifically, this paper presents an original method for generating equations and solving them. The proposed method was validated by numeric experiments on the same prototype for 2D and 3D cases. The method successfully transforms 2D and 3D geometrics objects from an initial configuration into a final one with respect of the specifications made by the designer. This paper deals with dimensional specification made to the geometry but the method remains applicable using others constraints such as chirality for instance.

In the future, the authors intend to use this approach to model and solve more complex problems with surface or volume type objects linked by constraints, like distances between planes, volume of object, etc.

References

- Bouma, W., Fudos, I., Hoffmann, C-M., Cai, J. and Paige, R. (1995) ‘A geometric constraint solver’, *Computer-Aided Design*, Vol. 27, No. 6, pp.487–500.
- Cartan, E. (1946) *Leçons sur la géométrie des espaces de Riemann* (Cahiers scientifiques 2), 2nd ed., Paris, Gauthier-Villars.
- Clément, A., Rivière, A. and Temmerman, M. (1994) *Cotation tridimensionnelle des systèmes mécaniques – Théorie et pratique*, PYC, France.
- Hestenes, D. (1966) *Space-Time Algebra*, Gordon & Breach, New York.
- Hestenes, D. (1986) *New Foundations for Classical Mechanics*, Reidel, Dordrecht.

- Hestenes, D. and Sobczyk, G. (1984) *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*, Reidel, Dordrecht.
- Hoffmann, C-M., Lomonosov, A. and Sitharam, M. (1997) ‘Finding solvable subsets of constraint graphs’, *Proceedings of Principles and Practice of Constraint Programming*, Linz, Austria, pp.463–477.
- Lesage, D. (2002) *Un modèle dynamique de spécifications d'ingénierie basé sur une approche de géométrie variationnelle*, PhD Thesis, Institute of National Polytechnique de Grenoble.
- Michelucci, D. and Foufou, S. (2004) ‘Using Cayley-Menger determinants for geometric constraint solving’, *ACM Symposium on Solid Modeling*, pp.285–290.
- Michelucci, D. and Foufou, S. (2006) ‘The witness configuration method’, *Computer Aided Design*, Vol. 38, No. 4, pp.284–299.
- Moinet, M. and Serré, P. (2011) ‘Declarative modelling and transformation of a constrained geometric object’, *International Journal of Product Development*, Vol. 13, No. 4, pp.344–361.
- Moinet, M. (2008) *Descriptions non-cartésiennes et résolutions de problèmes géométriques sous contraintes*, PhD Thesis, Ecole Centrale Paris.
- Rivière, A. (1993) *La géométrie du groupe des déplacements appliquée à la modélisation du tolérancement*, PhD Thesis, Ecole Centrale Paris.
- Serré, P. (2000) *Cohérence de la spécification d'un objet de l'espace euclidien à n dimensions*, PhD Thesis, Ecole Centrale Paris.
- Serré, P., Anwer, N. and Yang, J. (2011) *Guide to geometric Algebra in Practice: On the Use of Conformal Geometric Algebra in geometric Constraint Solving*, Springer-Verlag, pp.217–232.