# Model reduction in geometric tolerancing by polytopes

Vincent Delos, Santiago Arroyave-Tobón, Denis Teissandier

# Model reduction in geometric tolerancing by polytopes

Vincent Delos, Santiago Arroyave-Tobón, Denis Teissandier

*Univ. Bordeaux, I2M, UMR 5295, F-33400 Talence, France*

**Abstract**

There are several models used in mechanical design to study the behavior of mechanical systems involving geometric variations. By simulating defects with sets of constraints it is possible to study simultaneously all the configurations of mechanisms, whether over-constrained or not. Using this method, the accumulation of defects is calculated by summing sets of constraints derived from features (toleranced surfaces and joints) in the tolerance chain. These sets are usually unbounded objects ($\mathbb{R}^6$-polyhedra, 3 parameters for the small rotation, 3 for the small translation), due to the unbounded displacements associated with the degrees of freedom of features. For computational and algorithmic reasons, cap facets are introduced into the operand polyhedra to obtain bounded objects ($\mathbb{R}^6$-polytopes) and facilitate computations. However, the consequence is an increase in the complexity of the models due to the multiplication of caps during the computations. In response to this situation, we formalized and tested a method for controlling the effects of cap facets. Based on the combinatorial properties of polytopes, we propose to trace the operand faces during the different operations. An industrial case is solved and discussed in order to show the significant gain in computational time when applying the new method. This example has been chosen to be as general as possible to illustrate the genericity of the method.

*Keywords:* Tolerance analysis, Degrees of freedom, Polytopes, Traceability, Cap half-space, Minkowski sum

## 1. Introduction

In mechanical design, a tolerance zone represents the limits of manufacturing defects for a given surface. When the surface is considered as a discrete set of points, this restriction is transferred from each point to a given 3D point $M$. This point is assumed rigidly linked to the toleranced surface. Considering manufacturing defects as small displacements [1], each constraint can be modelled as a half-space in the 6-dimensional space of deviations [2]:

$$\bar{H}_k^+ = \left\{ \boldsymbol{x} \in \mathbb{R}^6 : b_k + a_{k_1} x_1 + ... + a_{k_6} x_6 \geq 0 \right\} \qquad (1)$$

where $x_1$, $x_2$, $x_3$ are the rotation variables, $x_4$, $x_5$, $x_6$ are the translation variables, the second member $b_k$ is related to the size of tolerance zone and $a_{k_j}$ $(1 \leq j \leq 6)$ are scalar parameters dependent on the geometry of the toleranced feature and the location of the point $M$ where the constraints are defined.

When a set of $m$ points is considered, a set of $k_{max} = 2m$ half-spaces is obtained. They, in turn, define a convex $\mathcal{H}$-polyhedron (where $\mathcal{H}$ stands for half-space) in $\mathbb{R}^6$ [2]:

$$\Gamma = \bigcap_{k=1}^{k_{max}} \bar{H}_k^+ \qquad (2)$$

Let us take, for example, the toleranced surface depicted at the left-hand side in Figure 1. For illustrative purposes, let us consider it as a 2D model: only the rotation along the $z$-axis and the translations along the $x$-axis and the $y$-axis (hereafter $r_z$, $t_x$ and $t_y$ respectively) are taken into account. The restriction on position imposed by the tolerance

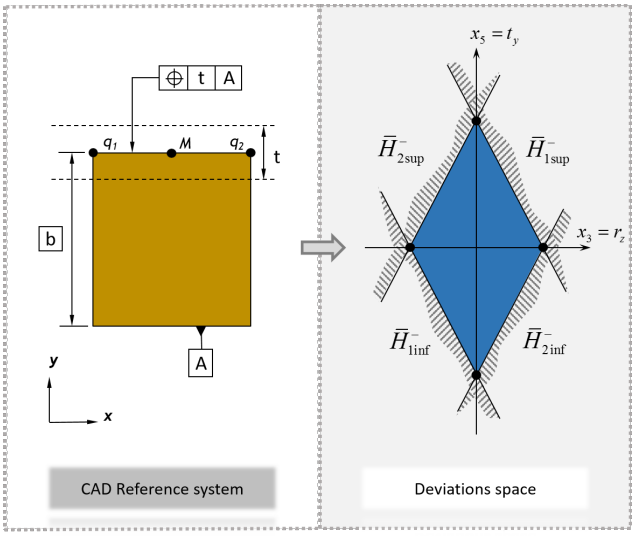



Figure 1: From a toleranced feature in the CAD reference system (left) to its respective polytope in the deviations space (right) considering the constraints expressed at point M.

zone on the toleranced feature can be modelled with four half-spaces. For point $q_1$, the half-spaces $\{\bar{H}^+_{1sup}, \bar{H}^+_{1inf}\}$ are obtained. Similarly, $\{\bar{H}^+_{2sup}, \bar{H}^+_{2inf}\}$ are derived from the restriction of $q_2$. These constraints are expressed at point $M$ (see Figure 1). In the space spanned by $[r_z, t_y]$ the intersection of these half-spaces defines a bounded polyhedron, i.e. a polytope (see right-hand side of Figure 1). As the tolerance zone does not impose limits on $t_x$, the intersection of the half-spaces generates an unbounded object in the 3D space defined by $[r_z, t_x, t_y]$.

Similarly, the displacements of a couple of surfaces potentially in contact can be modelled with a polyhedron.

When a unilateral contact (e.g. a planar contact) is considered, just one half-space is obtained for each point on the nominal surface, i.e. $k_{max} = m$ in Eq. 2. For further details about the definition of geometric constraints the reader can refer to [2, 3, 4]. Similar models for geometric tolerancing based on sets of constraints are presented in [5, 6]. The main difference between these models is the type of manipulated constraints. While polytopes handle only linear constraints, the other models handle quadratic constraints. A review of some of these methods is presented in [7, 8] and a comparison with the parametric approaches is presented in [9].

In short, a polyhedron represents all the possible relative displacements between two features of a mechanical system. It may be a feature regarding its nominal definition (geometric constraint), two features of two distinct parts sharing a mating condition (contact constraint).

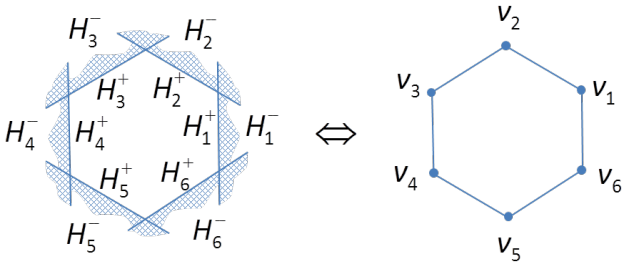



Figure 2: Equivalence of the $\mathcal{H}$ and $\mathcal{V}$-representations. We can map each half-space with a facet as none of the half-spaces is redundant.

The vertices of a polyhedron represent the extreme positions of the respective toleranced feature inside the tolerance zone or between two parts potentially in contact. While the native input data used for defining polyhedra in geometric tolerancing is the $\mathcal{H}$-representation, the $\mathcal{V}$-representation (where $\mathcal{V}$ stands for vertex) is also required for computing some operations. The Minkowski-Weyl theorem states that both definitions are equivalent [10] and this is illustrated in Figure 2.

Fleming [11] established the correlation between cumulative defect limits on parts in contact and the Minkowski sum of sets of constraints. In other words, if several parts are mated in a serial configuration, the stack-up of geometrical deviations can be calculated by summing the geometric and contact polyhedra involved in the toleranced chain (see Figure 3).

Similarly, the interaction of the deviations when parts are mated in parallel, i.e. sharing multiple contacts, can be modelled as the intersection of their respective sets of constraints (see Figure 4). It can be verified that the functional requirements of a mechanical system have been satisfied by checking the inclusion of a calculated polytope (the one representing the whole stack-up of deviations) in a functional one. A functional polytope can be created with the same procedure that defines a geometric specification between any two surfaces of the mechanism.

Among the operations required in tolerance analysis with sets of constraints, the Minkowski sum is the most expensive. Several studies have been published on the subject (not limited to tolerance analysis), but most of them focus on 3D polytopes [13, 14, 15, 16]. The algorithms described in [17, 18, 19] work in any n-dimensional space. However, they do not provide the $\mathcal{H}$-description of the calculated polytope, required in tolerance analysis for computing subsequent intersections. In [20] and [21] algorithms are presented for summing $\mathcal{HV}$-polytopes (i.e. polytopes defined by both their $\mathcal{H}$-representation and $\mathcal{V}$-representation) in $\mathbb{R}^n$ taking advantage of the dual property of polytopes.

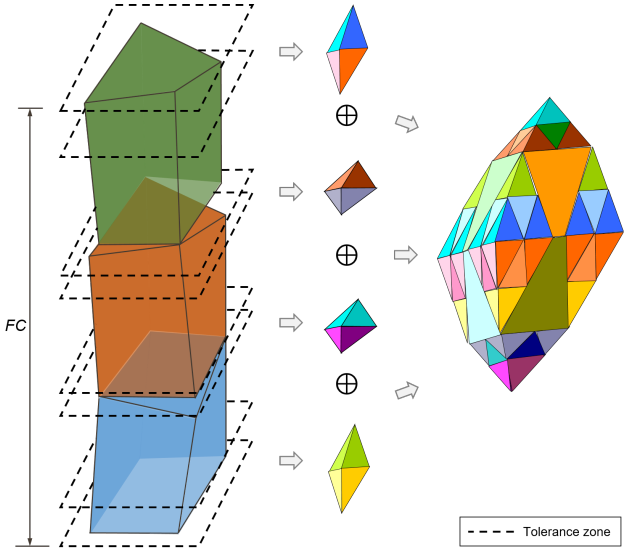



Figure 3: Modelling stack-up of deviations as the sum of polytopes

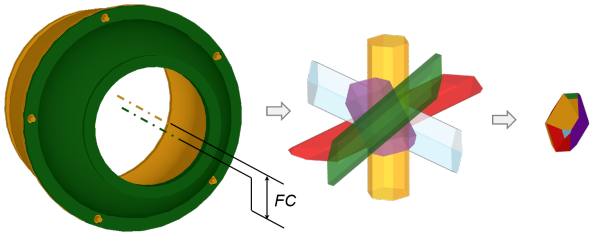



Figure 4: Modelling stack-up of deviations as the intersection of polytopes [12]

Generally, sets of constraints derived from toleranced features or joints (Eq. (2)) define unbounded sets in $\mathbb{R}^6$. Only in the very rare case of complex surfaces the obtained sets are bounded. This is a consequence of the degrees of invariance of the toleranced features or the degrees of freedom of the joints which define unbounded displacements [22, 23]. For this reason, the algorithms for summing polytopes discussed previously are not directly applicable.

In the case of Minkowski sums of polyhedra, only a few studies have been published in a n-dimensional space, a lot of examples in the literature mention algorithms specifically 2D or 3D which are not portable in higher dimensional spaces. Fukuda [18] presented an algorithm to compute Minkowski sums of $\mathbb{R}^n$-polytopes, mentioning the possibility of applying the same procedure to the case of polyhedra with at least one vertex (pointed polyhedra) by treating infinite rays as points at infinity. However, most of the polyhedra manipulated in tolerancing do not have vertices. Homri et al. [3] suggested turning polyhedra into polytopes by introducing virtual boundaries, called cap half-spaces. This strategy has to cope with the multiplication of cap half-spaces during the computation of Minkowski sums. As a consequence, the time taken for computing caps is in general far greater than that needed for computing facets repre-

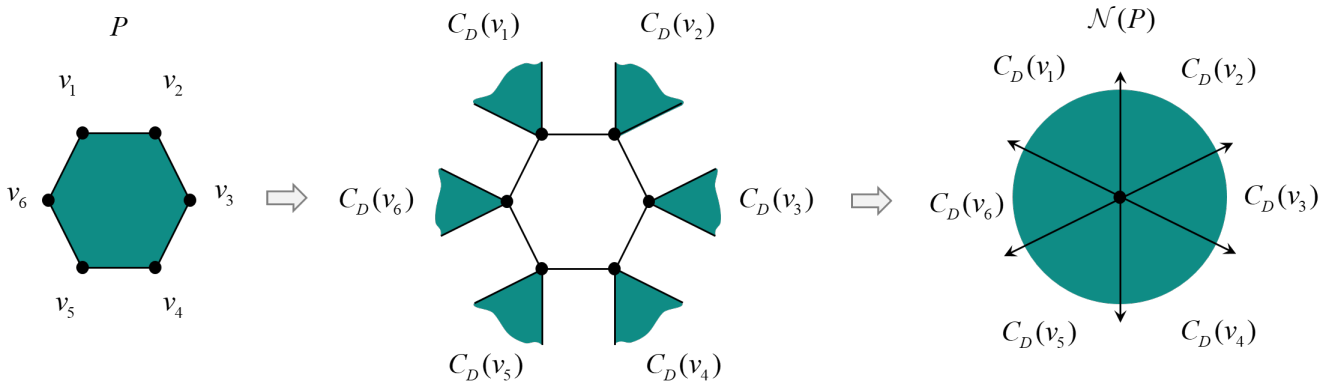



Figure 5: From a polytope $P$ in the primal space (left) to its normal fan $\mathcal{N}(P)$ in the dual space (right)

senting real limits of bounded displacements.

In response to this situation, this paper proposes to reduce the complexity of the models by tracing the cap half-spaces during tolerance simulations to limit the generation of new ones. The main algorithm presented here can be used to sum $\mathbb{R}^n$-polyhedra. The idea is to compute it through the sum of their underlying $\mathbb{R}^n$-polytopes, provided that we can store the double description of the polytopes i.e. the complete lists of both vertices and facets. The size of a polytope defined as $n(k + l)$, where $k$ is the number of vertices and $l$ the number of facets, is the space required to store the whole polytope. As an example, $\mathbb{R}^n$-tetrahedrons have $(n + 1)$ vertices and $(n + 1)$ facets, however $\mathbb{R}^n$-cubes have $2n$ facets but $2^n$ vertices, which makes the last ones unfit for such an algorithm due to the exponential. For more details on families of polytopes in the frame of $\mathbb{R}^n$ computational methods see [24]. The concepts briefly introduced in [25], are here formalized and demonstrated through the use of combinatorial representations and also extended to operations like intersections and inclusion checking which are mandatory to perform a full tolerance analysis. The starting point is the tolerance analysis method based on polytopes, summarized in section 2. When the initial sets of constraints are defined, the caps are identified among the whole set of half-spaces and labeled. Traceability is formalized for intersections, sums and inclusion checking (section 3). The advantages of the proposed method are illustrated later in section 4 by means of a real industrial application. The strengths, limitations and future prospects of our contribution are discussed in section 5.

## 2. Overview: sum of polyhedra as the sum of associated polytopes (cap-based method)

For algorithmic and computational reasons the manipulation of unbounded objects, i.e. polyhedra, is challenging in computational geometry. Among the operations commonly required in geometric tolerancing, summation is the most critical when dealing with this kind of operands.

Thanks to its capability of dealing with $\mathcal{HV}$-polytopes, the approach for summing polytopes based on dual cones is of interest in geometric tolerancing [20, 21]. This method is based on the property proposed by Ziegler [10], which states that the normal fan of a Minkowski sum is the common refinement of the normal fans of its summands. Next, we present a summary of this method as it is the starting point of this work.

**Definition 2.1.** *Let $\Gamma$ be a polytope or a polyhedron, $h$ be a vector of $\mathbb{R}^n$. A face of $\Gamma$ is defined with respect to a vector $h$ as $face_h(\Gamma) = \{x \in \Gamma, x \cdot h \leq y \cdot h, \forall y \in \Gamma\}$. It is the intersection of $\Gamma$ with the hyperplane $\bar{H} = \{x \in \mathbb{R}^n, x \cdot h = min(y \cdot h), \forall y \in \Gamma\}$ whose normal is the vector $h$.*

*The dimension of a face is the dimension of its affine hull: a 0-face is called a vertex, a 1-face is an edge and in $\mathbb{R}^n$ a (n-1)-face is a facet.*

The faces of a polyhedron can be identified as a simple intersection between $\Gamma$ and a hyperplane $\bar{H}$ such that $\Gamma$ is on one side of $\bar{H}$ i.e. $\Gamma$ is not separated by $\bar{H}$. Now let us consider the normal of $\bar{H}$ pointing outwards to $\Gamma$. For a given face $F$, we want to study the set described by all the normals to the hyperplanes identifying $F$.

**Definition 2.2.** *Let $F$ be a face of polyhedron $\Gamma$, the normal cone of $\Gamma$ at $F$ is $C_D(F) = \{h \in \mathbb{R}^n, face_h(\Gamma) = F\}$.*

The rays of the dual cone $C_D(F)$ are the outward normals to the facets containing $F$. Figure 5 provides an example of such cones. Let us write as $\mathcal{L}_\Gamma$ the list of all the faces of $\Gamma$ and as $\mathcal{L}_\Gamma^j$ the list of faces of dimension $j$ only. Then $\mathcal{L}_\Gamma^0 = \mathcal{V}_\Gamma$ is the list of vertices of $\Gamma$, $\mathcal{L}_\Gamma^1$ its list of edges and $\mathcal{L}_\Gamma^{n-1}$ its list of facets. $\mathcal{L}_\Gamma = \bigcup_{j=0}^{n} \mathcal{L}_\Gamma^j$

**Definition 2.3.** *The normal fan of $\Gamma$ is the collection of all the normal cones of $\Gamma$*

$$\mathcal{N}(P) = \left\{C_D(F), \forall F \in \mathcal{L}_\Gamma\right\}$$

*In the particular case of a polytope, the fan is complete and it forms a partition of the whole space $\mathbb{R}^n$.*

An example of the construction of a normal fan of a 2D polytope is presented in Figure 5.

**Definition 2.4.** *Let $P_A$ and $P_B$ be two full-dimensional polytopes in $\mathbb{R}^n$, $\mathcal{L}_A$ and $\mathcal{L}_B$ their respective lists of faces and $\mathcal{N}(P_A)$ and $\mathcal{N}(P_B)$ their normal fans. The common refinement of $\mathcal{N}(P_A)$ and $\mathcal{N}(P_B)$ can be defined as [10]:*

$$\mathcal{N}(P_A) \wedge \mathcal{N}(P_B) = \left\{ \begin{array}{l} C_D(F_A) \cap C_D(F_B) : \\ C_D(F_A) \in \mathcal{N}(P_A)\ \forall F_A \in \mathcal{L}_A, \\ C_D(F_B) \in \mathcal{N}(P_B)\ \forall F_B \in \mathcal{L}_B \end{array} \right\}$$

**Definition 2.5.** *The Minkowski sum of two sets is defined as:*

$$P_A \oplus P_B = \{a + b, a \in P_A, b \in P_B\}$$

**Theorem 2.6.** *The normal fan of a Minkowski sum is the common refinement of the individual fans (proof in [10]):*

$$\mathcal{N}(P_A \oplus P_B) = \mathcal{N}(P_A) \wedge \mathcal{N}(P_B)$$

An example of a Minkowski sum of two polytopes is presented in Figure 6a. As they are full-dimensional polytopes, their normal fans $C_{P_A}$ and $C_{P_B}$ partition $\mathbb{R}^2$. In turn, their intersection $C_{P_A} \cap C_{P_B}$ is also a complete fan, which is in fact the normal fan of the sum $P_A \oplus P_B$.

Figure 6b presents an example of a polytope derived from a polyhedron $\Gamma_A$ bounded by a couple of cap half-spaces $\bar{H}^+_{C_1}$ and $\bar{H}^+_{C_2}$. In geometric tolerancing, these half-spaces virtually limit the unbounded displacements of the related feature or joint. Therefore, the required number and their placement are determined according to the joint type or the class of the surface. The idea behind this strategy is to generate an initial set of operand polytopes belonging to $\mathbb{R}^6$ and introducing the required set of cap half-spaces to the set of geometric constraints. Moreover, Homri et al. [3] demonstrated that the sum of two polyhedra can be calculated as the sum of two associated polytopes, as depicted in Figure 6c.

Bounding polyhedra into polytopes avoids the manipulation of vertices placed at infinity. In addition, the normal fans of the operands are complete, and thus the calculation of the common refinement implies only full-dimensional intersections. As can be seen in Figure 6b, the unbounded sides of the calculated polytope are limited by three cap half-spaces when just one would suffice. We usually call this situation the multiplication of caps. This problem is worst when the dimension of the space increases due to the new connection facets that are generated (this phenomenon does not occur in 2D). The consequences of this multiplication are felt when a calculated polytope is used recursively as an operand in a subsequent sum. This implies that the time for computing cap facets is in general far greater than that for computing significant facets.

## 3. Proposed approach: traceability of half-spaces

### *3.1. Defining cap half-spaces*

When summing two polyhedra using the sum of two associated polytopes, we have to ensure that the main characteristics of the polyhedra are preserved. This means ensuring that the topological structure of each polyhedron $\Gamma$ is included inside that of its associated polytope $P$. To do this, we employ the notion of combinatorial representation, introduced by Fukuda in [26]. He describes an algorithm that computes the list of all the faces $\mathcal{L}_P$ of a polytope $P$ given the list of its facets $\mathcal{L}^{n-1}_P = \{F_1, ..., F_M\}$. The combinatorial representation of a given face $F$ of $P$, is the set of facet indices $j$ of $\mathcal{L}^{n-1}_P$ such that $F$ is a subface of $F_j$. This notion is interesting in the sense that it can identify the same topological objects whether they belong to a polytope or an unbounded polyhedron such as that depicted in Figure 7. Let us note as $\mathcal{CR}(F)$ the combinatorial representation of $F$. Based on this approach, we propose the following definition:

**Definition 3.1.** *Let $\Gamma$ be a polyhedron with the current list of facets $\mathcal{L}^{n-1}_\Gamma = \{F_1, ..., F_w\}$, $\mathcal{H}^{cap} = \{\bar{H}^+_u, u = 1, ..., w'\}$ is a list of cap half-spaces for $\Gamma$ if and only if:*

- *$P = \Gamma \cap (\cap_u \bar{H}^+_u)$ is a polytope with the current facet numbering for $\mathcal{L}^{n-1}_P$, $\{1, ..., w\}$ for the numbers of the polyhedron facets and $\{w+1, \cdots, w+w'\}$ for the cap facets*
- *the combinatorial representations of the faces of $\Gamma$ are also combinatorial representations of faces of $P$.*

Finding the half-spaces of $\mathcal{H}^{cap} = \{\bar{H}^+_u\}$ to build the polytope $P$ should not be a problem in tolerancing analysis as we work in the space $\mathcal{S}$ of small displacements. We choose the set $\{\bar{H}^+_u\}$ to make sure they include this space and do not interfere with it and that each half-space in $\mathcal{H}^{cap}$ provides a facet in $P$. We usually use a box bounding $\mathcal{S}$, for more details see [3].

### *3.2. Tracing caps in sums*

The definition 3.1 is able to handle efficiently the addition of sets of constraints as we compute the facets of a sum of polyhedra through a sum of polytopes. The question now is: can we recover all the facets of $\Gamma = \Gamma_1 + \Gamma_2$ computing $P = P_1 + P_2$ with $P_1 = \Gamma_1 \cap (\cap_u \bar{H}^+_u)$ and $P_2 = \Gamma_2 \cap (\cap_u \bar{H}^+_u)$?

#### *3.2.1. Basic properties*

The traceability of cap facets in Minkowski sums is based on a property demonstrated by Fukuda [18]: in a sum of polytopes, every face of the result can be decomposed into a sum of faces from the different summands and such a decomposition is unique. We apply this to a facet $F_P$ of the polytope $P = P_1 + P_2$ where $F_P = F_{P_1} + F_{P_2}$. Note that although $F_P$ is a facet, $F_{P_1}$ and $F_{P_2}$ might not be. In most cases in $\mathbb{R}^6$, $F_P$, $F_{P_1}$ and $F_{P_2}$ have different dimensions. For example, in $\mathbb{R}^2$, a facet $F_P$ is a 1-face and we have 3 cases:

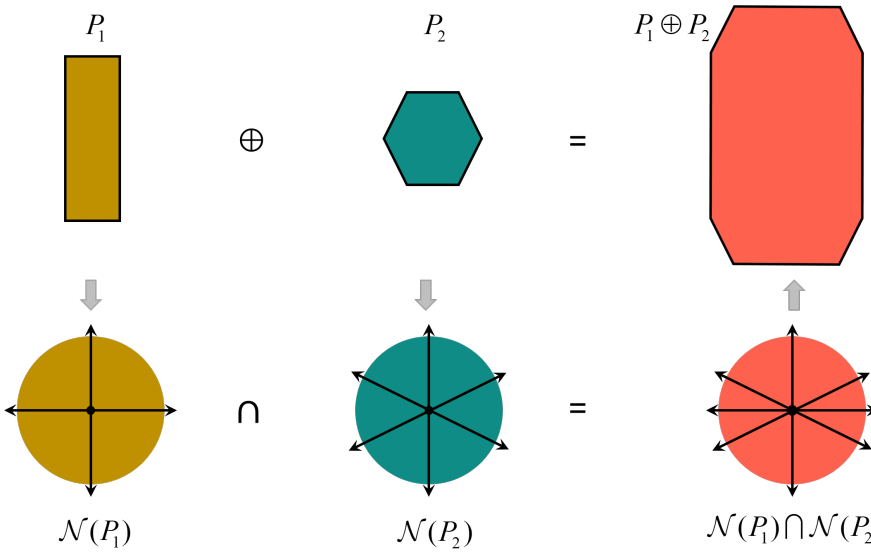
- $F_{P_1}$ is a vertex (0-face) and $F_{P_2}$ is a facet (1-face)
- $F_{P_1}$ is a facet (1-face) and $F_{P_2}$ is a vertex (0-face)
- $F_{P_1}$ is a facet (1-face) and $F_{P_2}$ is a facet (1-face)

The last case occurs when $F_{P_1}$ and $F_{P_2}$ are parallel. We are going to identify $F_{P_1}$ and $F_{P_2}$ with their vertices.

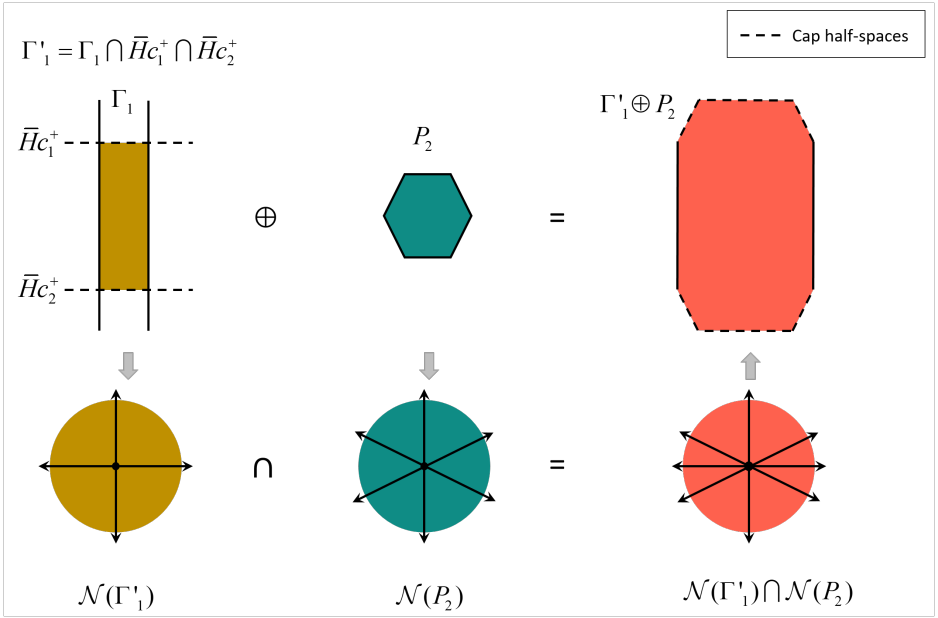
Let $F_\Gamma \in \mathcal{L}^{n-1}_\Gamma$, then the decomposition of $F_\Gamma = F_{\Gamma_1} + F_{\Gamma_2}$ in faces of $\Gamma_1$ and $\Gamma_2$ is unique. Moreover $F_{\Gamma_1}$ and $F_{\Gamma_2}$ have corresponding faces in $P_1$ and $P_2$ such that $\mathcal{CR}(F_{\Gamma_1}) = \mathcal{CR}(F_{P_1})$ and $\mathcal{CR}(F_{\Gamma_2}) = \mathcal{CR}(F_{P_2})$. The dual cones associated to the faces $F_{\Gamma_1}$ and $F_{P_1}$ are the same, because the rays of these cones are the outer normals of the facets that contain $F_{\Gamma_1}$ and $F_{P_1}$. So we have $C_D(F_{\Gamma_1}) = C_D(F_{P_1})$ and $C_D(F_{\Gamma_2}) = C_D(F_{P_2})$.

If $\gamma$ is the vector orthogonal to the hyperplane of $F_\Gamma$ then $\gamma = C_D(F_{\Gamma_1}) \cap C_D(F_{\Gamma_2}) = C_D(F_{P_1}) \cap C_D(F_{P_2})$. This proves that for each facet $F_\Gamma \in \mathcal{L}^{n-1}_\Gamma$, there is a corresponding facet $F_P$ in $P = P_1 + P_2$ such that $F_P \parallel F_\Gamma$. We can therefore obtain all the facets of $\Gamma$ through the polytope $P$, which justifies the definition 3.1.

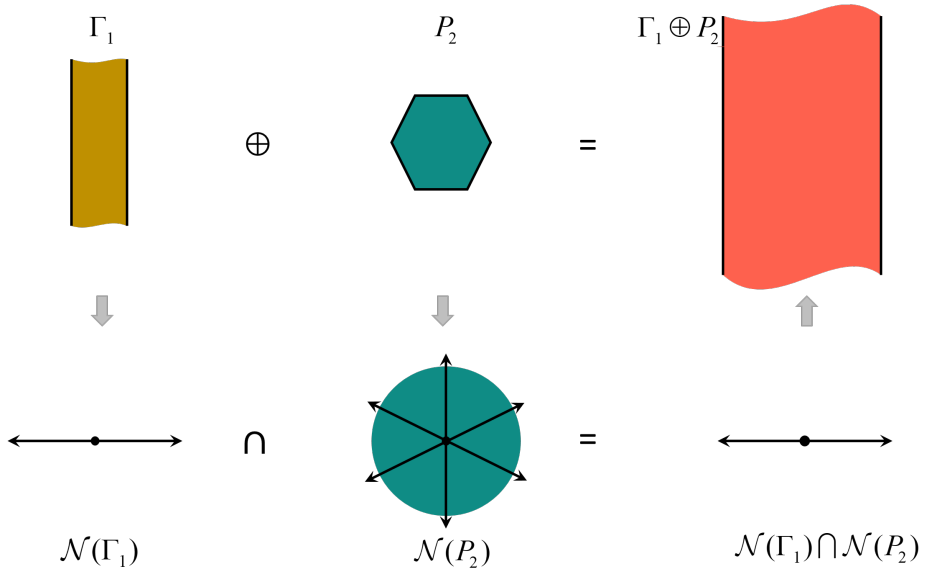
We trace the facets and vertices of the operands through the different operations to know if they are caps or not. The fact that we work on polytopes, i.e. bounded objects, enables us to identify each face, whatever its dimension, with its vertices. For example, in $\mathbb{R}^3$, each 0-face or vertex can be identified by a single vertex, each 1-face or edge by two vertices, each 2-face or facet by at least three vertices, and the 3-face or polytope by the whole set of vertices.




(a) Sum of polytopes: 1) compute the normal fans 2) intersect all the dual cones 3) build the sum in the primal space




(b) Sum of capped polyhedra




(c) Sum of polyhedra

Figure 6: Sum of polyhedra through the sum of associated polytopes

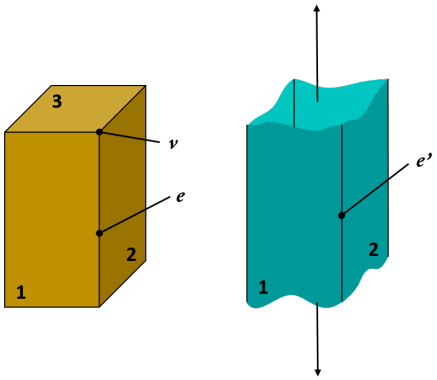



Figure 7: $\mathcal{CR}(v) = \{1,2,3\}$, $\mathcal{CR}(e) = \{1,2\}$, $\mathcal{CR}(e') = \{1,2\}$. The combinatorial representations of the 2 edges are the same, whether they are in the polytope (left) or the unbounded polyhedron (right).

**Theorem 3.2.** *Let $\Gamma_1, \Gamma_2$ be two polyhedra and $P_1$, $P_2$ their corresponding capped polytopes. Let $F_P$ be a facet of $P_1 \oplus P_2$ and $F_{P_1} + F_{P_2}$ its decomposition into faces of $P_1$ and $P_2$. $F_P$ has a matching facet $F_\Gamma \in \mathcal{L}^{n-1}_{\Gamma_1+\Gamma_2}$ such that $\mathcal{CR}(F_P) = \mathcal{CR}(F_\Gamma)$ if and only if the combinatorial representation of $F_{P_1}$ and $F_{P_2}$ does not contain any cap facet.*

**Proof.** We have $F_P \in \mathcal{L}^{n-1}_{P_1\oplus P_2}$ such that $F_P = F_{P_1} + F_{P_2}$ with no cap facets in $F_{P_1}$ and $F_{P_2}$ combinatorial representations. It means that $\exists F_{\Gamma_1} \in \mathcal{L}_{\Gamma_1}, \exists F_{\Gamma_2} \in \mathcal{L}_{\Gamma_2}$ such that

$$\begin{cases} \mathcal{CR}(F_{P_1}) = \mathcal{CR}(F_{\Gamma_1}) \text{ and } C_D(F_{\Gamma_1}) = C_D(F_{P_1}) \\ \mathcal{CR}(F_{P_2}) = \mathcal{CR}(F_{\Gamma_2}) \text{ and } C_D(F_{\Gamma_2}) = C_D(F_{P_2}) \end{cases}$$

with $F_{P_1} + F_{P_2} \in \mathcal{L}^{n-1}_{P_1\oplus P_2}$. So if $\gamma$ is the normal to the hyperplane supporting $F_{P_1} + F_{P_2}$, $\gamma = C_D(F_{P_1}) \cap C_D(F_{P_2}) = C_D(F_{\Gamma_1}) \cap C_D(F_{\Gamma_2})$. As a consequence, $F_{\Gamma_1} + F_{\Gamma_2} \in \mathcal{L}^{n-1}_{\Gamma_1\oplus\Gamma_2}$ and $F_{\Gamma_1} + F_{\Gamma_2}$ is the matching facet of $F_P = F_{P_1} + F_{P_2}$.

The reciprocal is straightforward: let us assume $F_P$ has a matching facet in $\Gamma_1 + \Gamma_2$, as a consequence $\exists F_\Gamma \in \mathcal{L}^{n-1}_{\Gamma_1+\Gamma_2}$ such that $F_P \parallel F_\Gamma$. Let us decompose the last facet, $\exists F_{\Gamma_1} \in \mathcal{L}_{\Gamma_1}, \exists F_{\Gamma_2} \in \mathcal{L}_{\Gamma_2}$ such that $F_\Gamma = F_{\Gamma_1} + F_{\Gamma_2}$. From definition 3.1, $\exists F_{P_1} \in \mathcal{L}_{P_1}, \exists F_{P_2} \in \mathcal{L}_{P_2}$ such that

$$\begin{cases} \mathcal{CR}(F_{P_1}) = \mathcal{CR}(F_{\Gamma_1}) \text{ and } C_D(F_{\Gamma_1}) = C_D(F_{P_1}) \\ \mathcal{CR}(F_{P_2}) = \mathcal{CR}(F_{\Gamma_2}) \text{ and } C_D(F_{\Gamma_2}) = C_D(F_{P_2}) \end{cases}$$

If $\gamma$ is the normal to the hyperplane supporting $F_\Gamma$ we knew that $\gamma = C_D(F_{\Gamma_1}) \cap C_D(F_{\Gamma_2})$, now we can write $\gamma = C_D(F_{P_1}) \cap C_D(F_{P_2})$. So $F_P = F_{P_1} + F_{P_2}$ and no half-space in the combinatorial representation of $F_{P_1}$ or $F_{P_2}$ is capped as $\mathcal{CR}(F_{P_1}) = \mathcal{CR}(F_{\Gamma_1})$ and $\mathcal{CR}(F_{P_2}) = \mathcal{CR}(F_{\Gamma_2})$.

The theorem 3.2 can explain why the number of cap facets soars when we sum polytopes: when $F_P = F_{P_1} + F_{P_2}$, having only one cap facet in the combinatorial representation of $F_{P_1}$ or $F_{P_2}$ is enough to transfer this property to the sum $F_P$. Moreover, we can use the previous theorem to trace the attributes attached to the topology of the operands. Then, after a Minkowski sum of polytopes, the facets still representing real limits in the associated tolerancing problem can be distinguished. Similarly, the set of facets generated from cap half-spaces can be identified.

In response to this, we propose to 'clean' the polytope resulting from a sum of its cap facets. This can be done by removing from the whole list of half-spaces those labeled as caps and then bounding it again. After this procedure, a minimum set of half-spaces is reestablished.

### *3.2.2. The algorithm*

With the help of theorem 3.2, we can build $F_{P_1}$ and $F_{P_2}$ using the relation $F_P = F_{P_1} + F_{P_2}$. We first want to compute $\mathcal{V}(P_1)$ and $\mathcal{V}(P_2)$, the lists of vertices identifying both $F_{P_1}$ and $F_{P_2}$. For all of these vertices, we then have the list of their supporting half-spaces. From these we decide the combinatorial representation of $F_{P_1}$ and $F_{P_2}$. The final step checks whether we can find at least one cap facet in the combinatorial representation of $F_{P_1}$ or $F_{P_2}$. If this is the case, $F_P$ is marked as a cap facet, if not, $F_P$ can be matched directly with a facet of $\Gamma$. This process is detailed in Figure 8: from the facet $F$ we get $\{c_1, c_2, c_3, c_4\}$, then $\{a_1, a_2\}$ and $\{b_1, b_2\}$ to check if one half-space $\bar{H}_i^+$ $i = 1, .., 4$ is capped.

Let $P_i$ be polytopes $\forall i$, instead of computing directly $\sum_i P_i$ causing a strong increment of the number of cap half-spaces we propose to follow the scheme:

- Compute $P_1 + P_2$
- Remove the cap half-spaces from the last sum, intersect the remaining half-spaces to get $\Gamma_{1,2}$
- Bound $\Gamma_{1,2}$ with a new bounding box to get the polytope $P_{1,2}$
- Compute $P_{1,2} + P_3$
- Remove the cap half-spaces from the last sum, intersect the remaining half-spaces to get $\Gamma_{1,2,3}$
- Bound $\Gamma_{1,2,3}$ with a new bounding box to get the polytope $P_{1,2,3}$
- Compute $P_{1,2,3} + P_4$
- ...

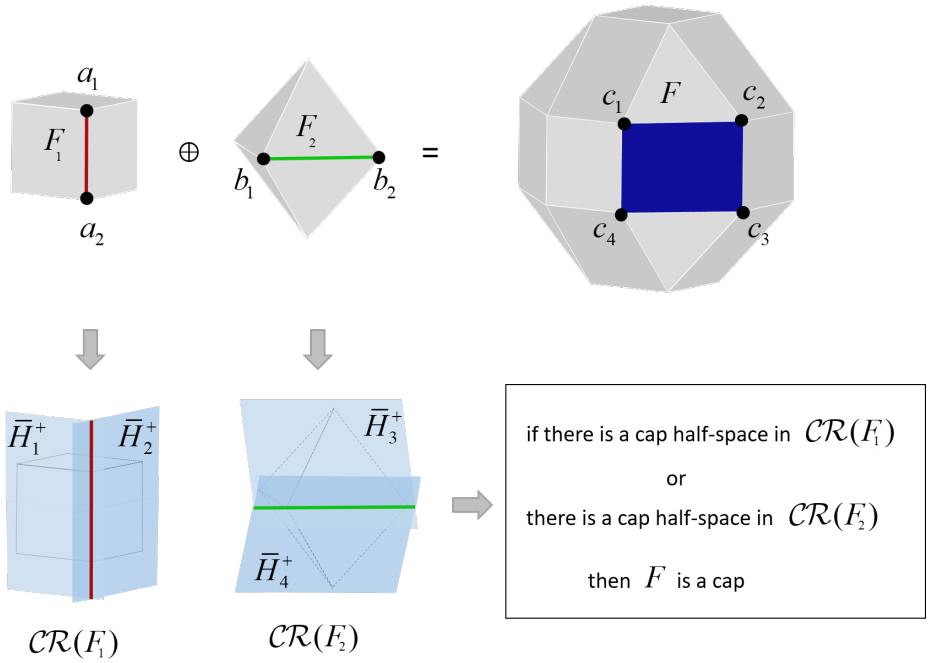



Figure 8: Tracing the facets of a sum

**Algorithm 1** Cap facets identification in a sum

**Require:** $\mathcal{V}(P_1), \mathcal{V}(P_2), \mathcal{V}(P), \mathcal{L}_{P_1}^{n-1}, \mathcal{L}_{P_2}^{n-1}, \mathcal{L}_{P}^{n-1}$
**Require:** The list of cap half-spaces in $P_1$ and $P_2$
**Ensure:** The list of cap half-spaces in $P$

**for** each $F_P \in \mathcal{L}_P^{n-1}$ **do**
  *// Compute $F_{P_1}$ and $F_{P_2}$ such that $F_P = F_{P_1} + F_{P_2}$*
  *// Get $\mathcal{V}(F_{P_1})$ the vertices of $F_{P_1}$*
  $\mathcal{V}(F_{P_1}) =$
  $\{a \in \mathcal{V}(P_1) / \exists b \in \mathcal{V}(P_2) \Rightarrow a + b \in \mathcal{V}(F_P)\}$
  *// Get $\mathcal{V}(F_{P_2})$ the vertices of $F_{P_2}$*
  $\mathcal{V}(F_{P_2}) =$
  $\{b \in \mathcal{V}(P_2) / \exists a \in \mathcal{V}(P_1) \Rightarrow b + a \in \mathcal{V}(F_P)\}$
  *// For each vertex collect its half-space numbers*
  **for** each $a \in \mathcal{V}(F_{P_1})$ **do**
    Get $\mathcal{H}_a = \{u, a \in \bar{H}_u\}$
  **end for**
  *// Get the combinatorial representation of $F_{P_1}$*
  Compute $\mathcal{CR}(F_{P_1}) = \{\cap \mathcal{H}_a, \forall a \in \mathcal{V}(F_{P_1})\}$
  **if** No cap half-space in $\mathcal{CR}(F_{P_1})$ **then**
    *// For each vertex collect its half-space numbers*
    **for** each $b \in \mathcal{V}(F_{P_2})$ **do**
      Get $\mathcal{H}_b = \{v, b \in \bar{H}_v\}$
    **end for**
    *// Get the combinatorial representation of $F_{P_2}$*
    Compute $\mathcal{CR}(F_{P_2}) = \{\cap \mathcal{H}_b, \forall b \in \mathcal{V}(F_{P_2})\}$
    **if** No cap half-space in $\mathcal{CR}(F_{P_2})$ **then**
      $F_P$ is not a cap half-space
    **else**
      $F_P$ is a cap half-space
    **end if**
  **else**
    $F_P$ is a cap half-space
  **end if**
**end for**

### *3.3. Tracing caps in intersections*

Even if intersections are not the most computationally expensive operations in geometric tolerancing with polytopes, they are usually present in the simulations. Hence, the recognition of cap facets during intersections allows us to maintain the traceability of the attributes throughout the entire simulation.

The intersection of two polytopes can be calculated by combining the list of half-spaces of both operands and removing the redundant ones. A half-space labeled as cap before an intersection just keeps its attribute if it is not removed because of redundancy. From the algorithmic point of view, this is not a complicated task. It simply involves implementing label management.

### *3.4. Tracing caps in inclusion checking*

The objective of a tolerance analysis is to determine whether a functional condition is satisfied. In the method based on polytopes, this can be done by checking the inclusion of the polytope representing the whole stack-up of deviations in the functional polytope [4].

When the cap half-spaces are not placed far enough from the origin (even if it is not advisable [3]), the inclusion of a calculated polytope in a functional one can be influenced in some instances. This situation could lead to the misleading conclusion that the design requirements are satisfied.

In order to avoid this, the inclusion of the calculated polytope must not depend on cap half-spaces. If this is the case, then the functional condition cannot be guaranteed. This can be done by tracing the labels of the half-spaces until the final operation, as described previously.

## 4. Industrial application

In order to show the advantages of cap traceability in geometric tolerancing, we solved an industrial application.

The same tolerance chain was simulated with and without the control of cap facets to show the gain in computational time and the reduction in complexity of the manipulated models.

The mechanism we studied is a sub-assembly of a cutting machine by Lectra © (`https://www.lectra.com/`). More specifically, it is the sharpening system of the machine. Control of the geometric variations for this system is important as the sharpening of the cutting blade has to be sufficiently precise.

For the simulation we considered only three parts: the supporting arm, represented by the number 1,0, is considered as the same part as the pin that blocks the rotation of the tension arm. The assembly of the tension arm with its axis is considered as a single part (numbered 2,0). The bearings are modelled as spherical joints, with clearance the tolerance provided by the manufacturer $c_b$. The last part is the pulley 3,0.

The functional condition (FC) requires control of the relative position of the pulley axis with respect to the tension arm axis (see FC in Figure 9). Using polytopes, we simulated the tolerance zones defined by the ISO specifications of the individual parts to calculate the maximal deviations of the assembly. For parts 1,0 and 2,0, a position specification was considered (Figures 10a and 10b respectively). A coaxiality specification was considered for part 3,0 (Figure 10c).

The topological model of the mechanism is shown in the contact graph in Figure 11. A node designated by two integers $\alpha, \beta$ represents the nominal model of a part when $\beta = 0$, and the substitute surface when $\beta \neq 0$. Each edge of the graph represents some deviations. These may be geometric deviations, in the case of inner edges, or deviations due to contacts, in the case of edges connecting two nodes from different parts. These deviations can be represented by geometric and contact polytopes respectively [3].

According to the assembly topology, the set of operations required to simulate the relative position of the surfaces involved by the functional condition can be determined:

$$\begin{aligned} P_{1,1/3,2} =& P_{1,1/1,0} \oplus ((P_{1,0/1,2} \oplus P_{1,2/2,2} \oplus P_{2,2/2,0}) \cap \\ & (P_{1,0/1,3} \oplus P_{1,3/2,3} \oplus P_{2,3/2,0})) \oplus P_{2,0/3,2} \end{aligned} \quad (3)$$

where $P_{f/g}$ is the polytope describing the relative position of surface $f$ with respect to surface $g$ (according to the numbering in Figure 11).

In Eq. (3) some polytopes are defined over the same surface and consequently they are homothetic. This is the case for operands $P_{2,0/2,1}$, $P_{2,1/3,1}$ and $P_{3,1/3,0}$. The sum of homothetic polyhedra can be performed directly by homothetic transformations and no numerical computation is needed. Datum surfaces are considered to be equal to their respective nominal surfaces, then their derived polytopes are neutral elements in a Minkowski sum ($P_{1,0/1,3}$, $P_{1,3/2,3}$, $P_{2,3/2,0}$ and $P_{3,0/3,2}$). Finally, Eq. (3) can be simplified as:

$$P_{1,1/3,2} = P_{1,1/1,0} \oplus (P_{1,0/2,0-a} \cap P_{1,0/2,0-b}) \oplus P_{2,0/3,2} \quad (4)$$

with:

$$P_{1,0/2,0-a} = P_{1,0/1,2} \oplus P_{1,2/2,2} \oplus P_{2,2/2,0}$$

$$P_{1,0/2,0-b} = P_{1,0/1,3} \oplus P_{1,3/2,3} \oplus P_{2,3/2,0}$$

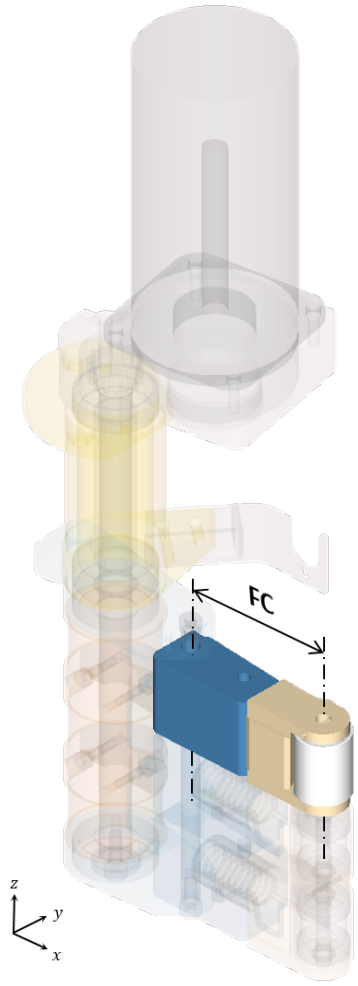



Figure 9: Case study: sharpening system

Operand polytopes were created according to the tolerances and lengths values presented in table 2. As shown in table 1, a couple of geometric half-spaces is created per each discretization point (DP). The final objective of the simulation is to check if $P_{1,1/3,2} \subseteq ? P_{FC}$, where $P_{FC}$ is the polytope representing the functional condition.

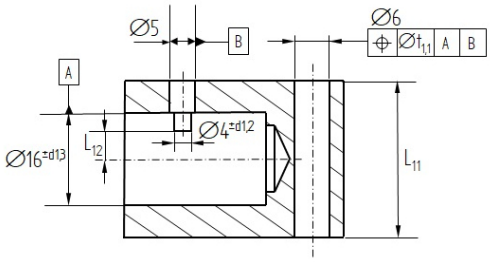
### *4.1. Simulation with 6D polytopes without caps control*

To perform the operations declared in Eq. (4), the operands must be bounded and belong to the same space, i.e. $\mathbb{R}^6$. To do this, some cap half-spaces were introduced into the operands. Polytopes $P_{1,1/1,0}$ and $P_{2,0/3,2}$ (from CAD cylindrical surfaces oriented along the $z$-axis) required two pairs of cap facets to virtually limit $r_z$ and $t_z$. Two pairs of cap half-spaces were introduced to $P_{1,0/2,0-b}$ (derived from a cylindrical surface oriented along the $x$-axis). As the contact between surfaces 1,2 and 2,2 was modelled as a ball-and-plane pair, five pairs of caps were required to bound the set. As shown in Table 1, the number of cap half-spaces was set at the strict minimum: two caps for each degree of freedom (DOF).

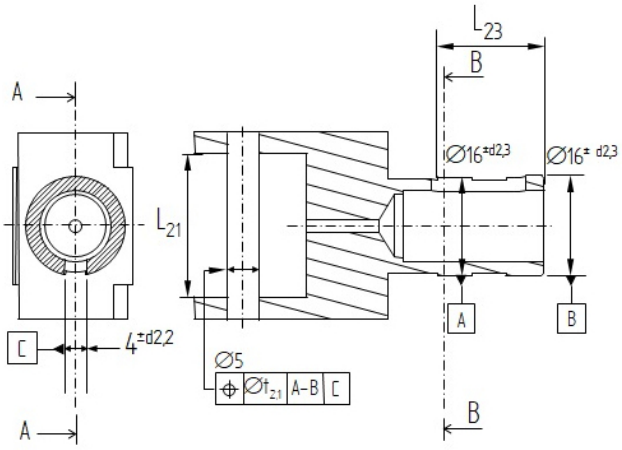
Figure 12 shows a 3D representation of the operands of the simulation. It is worth mentioning that some of the figures showing 3D polytopes in this paper are just partial representations since they were originally 6D objects. A projection into a three-dimensional subspace was therefore

Table 1: Details of the operands in $\mathbb{R}^6$ (HS: half-space)

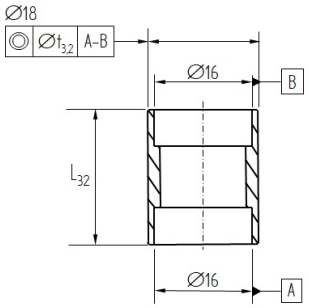
| Operand polytope | Feature type | Second member $b_k$ | DOF | Num Cap HS | DP | Num non-cap HS | Num vertices |
|---|---|---|---|---|---|---|---|
| $P_{1,1/1,0}$ | Cylindrical | $t_{1,1}/2$ | 2 | 4 | 8 | 16 | 256 |
| $P_{1,0/2,0-a}$ | Cylindrical | $(d_{1,3}+d_{2,3})/2$ | 2 | 4 | 8 | 16 | 256 |
| $P_{1,0/2,0-b}$ | Ball-and-plane | $(d_{1,2}+d_{2,2})/2$ | 5 | 10 | 1 | 2 | 64 |
| $P_{2,0/3,2}$ | Cylindrical | $(t_{2,1}+c_b+t_{3,2})/2$ | 2 | 4 | 8 | 16 | 256 |




(a) ISO specifications for part 1,0

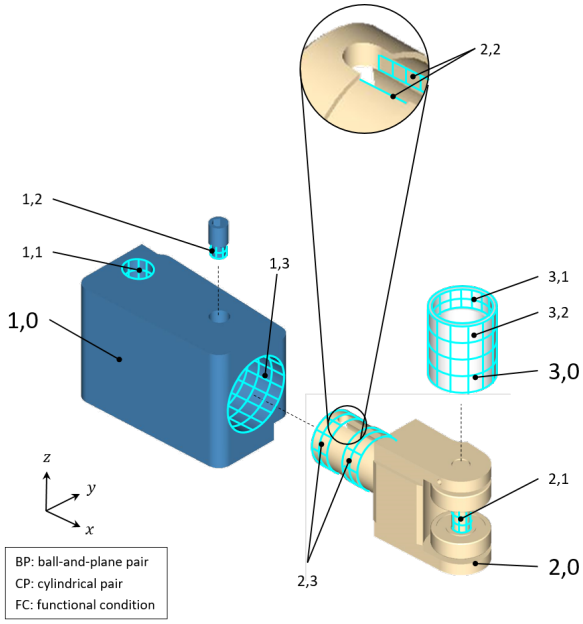



(b) ISO specifications for part 2,0




(c) ISO specifications for part 3,0

Figure 10: Features and contacts considered in the simulation



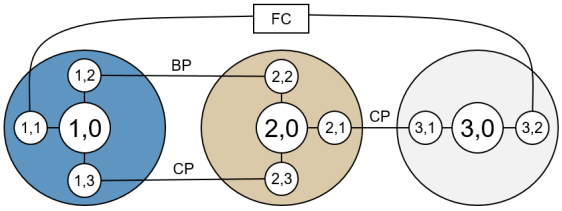




Figure 11: Surfaces numbering and contact graph

carried out. For this case, the operands were calculated at the centroid of surface 3,2 and were then projected to the subspace spanned by $[r_x, r_y, r_z]$ for visualization purposes.

The actual approach for performing the operations suggests directly intersecting and summing the initial set of operands. Table 3 summarizes the results of these operations. The computations were performed with the open source library, politopix: `http://i2m.u-bordeaux.fr/politopix.html`. Due to the propagation of the DOFs we can observe a large increase in the number of cap half-spaces after each sum. Having a pair of caps for each DOF is no longer guaranteed. This problem worsens when the

Table 2: Tolerances, clearances and lengths used for the simulation

| Tolerance | Magnitude [mm] |
|---|---|
| $t_{1,1}$ | 0.2 |
| $d_{1,2}$ | 0.2 |
| $d_{1,3}$ | 0.2 |
| $d_{2,3}$ | 0.2 |
| $d_{2,2}$ | 0.2 |
| $t_{2,1}$ | 0.2 |
| $c_b$ | 0.2 |
| $t_{3,2}$ | 0.2 |

| Length | Magnitude [mm] |
|---|---|
| $L_{11}$ | 27 |
| $L_{12}$ | 5.5 |
| $L_{21}$ | 21.6 |
| $L_{32}$ | 21.6 |
| $L_{23}$ | 16.5 |

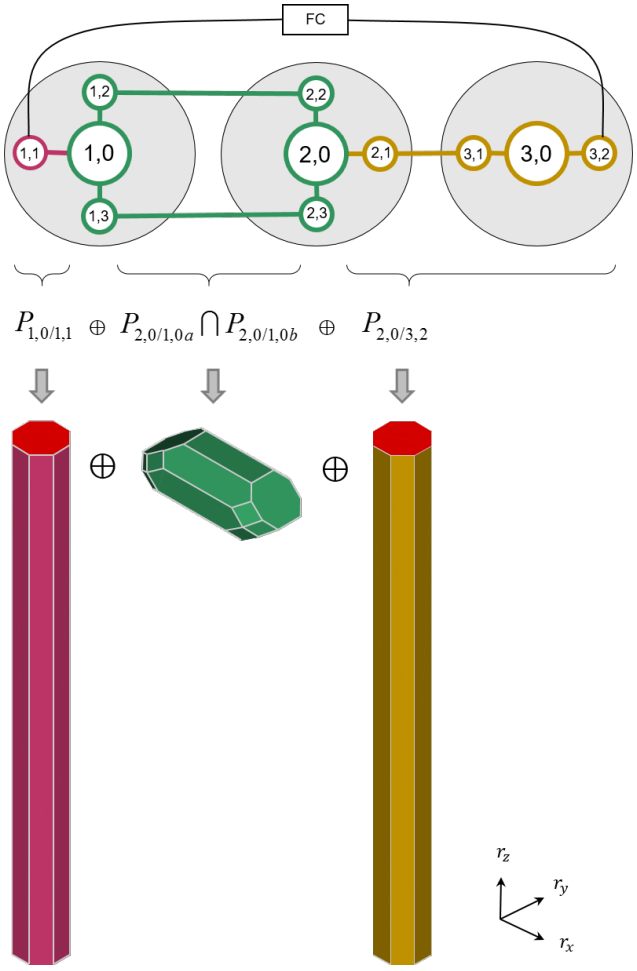



Figure 12: 3D representation of the 6D operands. Cap faces are shown in red.

calculated polytope, highly 'contaminated' with caps, is used again as an operand for a subsequent sum. This is actually the scenario of the final operation of the simulation. During this operation, 99.93% of the calculated facets come from cap half-spaces, and they therefore have no meaning in the related tolerancing problem.

### 4.2. Simulation with 6D polytopes with caps control

With the strategy proposed in this article, our aim is to keep the number of cap half-spaces of the operands to a minimum. After each sum we take the calculated polytope and filter meaningful half-spaces to restore a minimum set of caps. This implies a reduction in the complexity of the operands which is translated afterwards into a reduction in computational time when subsequent operations are calculated. This situation is illustrated in Table 4, summarizing the results of the simulation, following the proposed strategy. Although the first sum took approximately twice the time to compute, the complexity of the polytope we obtained was notably reduced. The increase in calculation time is due to the filtering process launched after the regular summation. The advantages of this can be clearly seen in the next operation: the final sum took just 3% of the computational time required for the first simulation. We conclude from this that the rest of the computational time in the first simulation was spent in calculating cap faces. This phenomenon is shown in Figure 13, where the final polytope calculated without controlling cap facets presents a complex set of faces to bound the $r_z$-axis, while only two appear in the calculated polytope when the trace and the control of caps was performed. It is worth mentioning that no information was lost by applying the new method. We are simply avoiding the calculation of meaningless information.

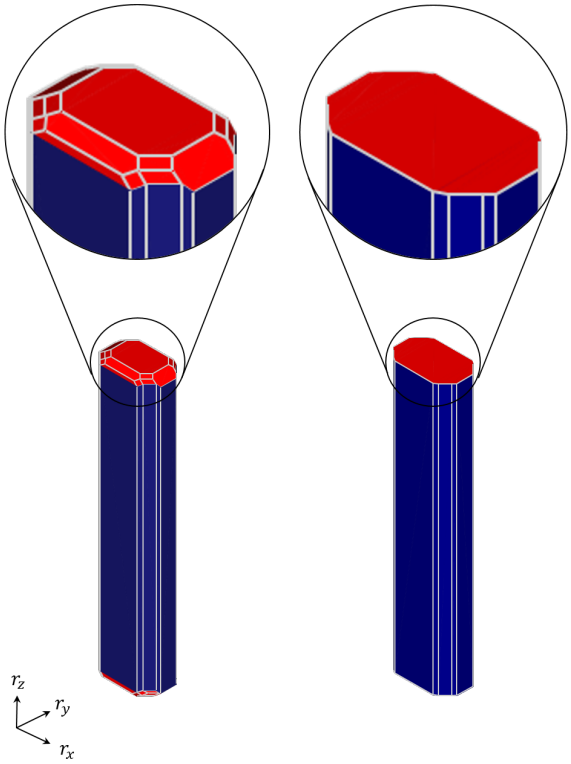



Figure 13: Calculated polytopes: without caps control at the left and with caps control at the right. Cap facets shown in red

### 4.3. Validation of results

The simulation described above was validated using the commercial tool MECAmaster [27]. The same CAD model was imported into the GUI and contacts between parts were created following the same assumptions. The same ISO specifications presented in Figure 10 were considered. Figure 14 shows the kinematic constraints that were created (magenta marks) and the functional condition (blue arrows).

The results of the MECAmaster simulation are represented as two pairs of points superimposed on the polytope in Figure 15. One simulation was carried out to obtain the deviations in $r_y$ and another was required to obtain the deviations in $r_x$. In fact, each simulation provides just a couple of values (a maximal and a minimal deviation) related to just one extreme configuration of the mechanism. Therefore, several simulations are usually required to study the behavior of the assembly under different extreme positions. In contrast, just one simulation with a set of constraints considers all possible configurations simultaneously.

As shown in Figure 15, the difference in the values of $r_y$ when $r_x = 0$ between the two methods is $10^{-6}$ rad, which is insignificant in the small displacements field. We attribute this difference to numerical approximations.

Table 3: Simulation of tolerances for the sharpening system without caps control (HS: half-space)

| Operand polytope | DOF | Num Cap HS | Num non-cap HS | Num vertices | Time [s] |
|---|---|---|---|---|---|
| $P_{1,0/2,0-a} \cap P_{1,0/2,0-b}$ | 1 | 2 | 18 | 256 | 0.01 |
| $P_{1,1/1,0} \oplus \left(P_{1,0/2,0-a} \cap P_{1,0/2,0-b}\right)$ | 3 | 2928 | 116 | 10728 | 34 |
| $P_{1,1/1,0} \oplus \left(P_{1,0/2,0-a} \cap P_{1,0/2,0-b}\right) \oplus P_{2,0/3,2}$ | 4 | 23932 | 16 | 60944 | 1641 |

Table 4: Simulation of tolerances for the sharpening system tracing and controlling caps (HS: half-space)

| Operand polytope | DOF | Num Cap HS | Num non-cap HS | Num vertices | Time [s] |
|---|---|---|---|---|---|
| $P_{1,0/2,0-a} \cap P_{1,0/2,0-b}$ | 1 | 2 | 18 | 256 | 0.02 |
| $P_{1,1/1,0} \oplus \left(P_{1,0/2,0-a} \cap P_{1,0/2,0-b}\right)$ | 3 | 6 | 116 | 1152 | 64 |
| $P_{1,1/1,0} \oplus \left(P_{1,0/2,0-a} \cap P_{1,0/2,0-b}\right) \oplus P_{2,0/3,2}$ | 4 | 8 | 16 | 256 | 57 |

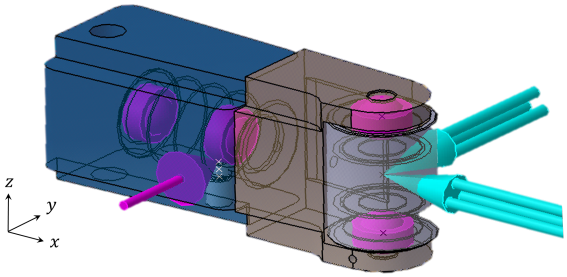



Figure 14: Kinematic model of the tolerance simulation with MECAmaster

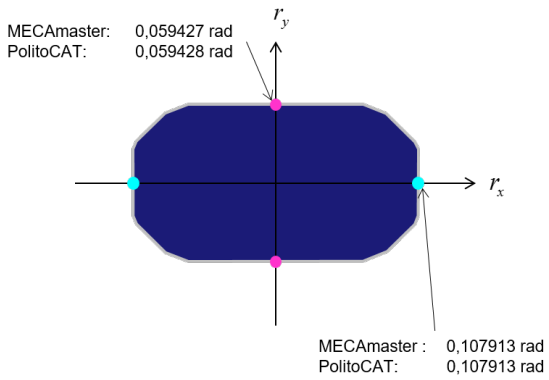



Figure 15: Comparison of the results from PolitoCAT and MECAmaster

## 5. Discussion and future work

We found that tracing the cap half-spaces along the different operations in geometric tolerancing with polytopes is able to reduce the complexity of the calculated sets during a simulation. This reduction leads to a decrease in computational time. This was demonstrated by means of an industrial application.

Cap half-spaces are initially required to bound the polyhedra to generate bounded objects, i.e. polytopes. The fact of having unbounded sets is due to the degrees of freedom of joints or the degrees of invariance of toleranced features. In order to be able to distinguish the cap half-spaces from those representing real limits in the tolerancing problem, traceability is the key concept. It has to be done throughout all the operations: sums, intersections, inclusion checking, and is based on the combinatorial properties of the polytopes demonstrated in theorem 3.2. The main contribution of the present work is the limitation of the impact of the cap facets during the computations but the principle of traceability, which identifies the contributors among a list of candidates, is general enough to be applied to other kinds of scientific problems whether they are related or not to tolerancing.

The strategy of labeling and tracing half-spaces is an important step towards tolerance synthesis. Future work is therefore required to develop a strategy for linking the topological elements of a calculated polytope with our input data: the initial set of ISO geometric specifications. With this information, the most influential features in the tolerance chain can be determined. This enables tolerances to be allocated in an optimal way, in order to reduce production costs.

The ideal scenario is when no cap half-spaces are used at all in the computations. This implies that the operations have to be performed directly with unbounded sets, i.e. polyhedra. However, the Minkowski sum of unbounded objects is still a challenging matter.

## References

[1] Pierre Bourdet, Luc Mathieu, Claire Lartigue, and Alexandre Ballu. The concept of the small displacement torsor in metrology. *Series on advances in Mathematics for Applied Sciences*, 40:110–122, 1996.

[2] D. Teissandier, V. Delos, and Y. Couétard. Operations on polytopes: application to tolerance analysis. In *Global Consistency of Tolerances*, pages 425–433, Enschede (Netherlands), 1999. Kluwer academic.

[3] Lazhar Homri, Denis Teissandier, and Alex Ballu. Tolerance analysis by polytopes: Taking into account degrees of freedom with cap half-spaces. *Computer-Aided Design*, 62:112 – 130, 2015.

[4] Santiago Arroyave-Tobón, Denis Teissandier, and Vincent Delos. Tolerance analysis with polytopes in HV-description. *ASME. J. Comput. Inf. Sci. Eng.*, 17(4):041011–041011–9, 2017.

[5] M. Giordano, D. Duret, S. Tichadou, and R. Arrieux. Clearance space in volumic dimensioning. *CIRP Annals - Manufacturing Technology*, 41(1):565 – 568, 1992.

[6] J.K Davidson, A Mujezinovic, and JJ Shah. A new mathematical model for geometric tolerances as applied to round faces. *Journal of mechanical design*, 124(4):609–622, 2002.

[7] G Ameta, S Serge, and Max Giordano. Comparison of spatial math models for tolerance analysis: tolerance-maps, deviation domain, and ttrs. *Journal of Computing and Information Science in Engineering*, 11(2):021004, 2011.

[8] Mathieu Mansuy, Max Giordano, and Joseph K Davidson. Comparison of two similar mathematical models for tolerance analysis: T-map and deviation domain. *Journal of Mechanical Design*, 135(10):101008, 2013.

[9] Jami J. Shah, Gaurav Ameta, Zhengshu Shen, and Joseph Davidson. Navigating the tolerance analysis maze. *Computer-Aided Design and Applications*, 4(5):705–718, 2007.

[10] Günter M. Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 1995.

[11] Alan Fleming. Geometric relationships between toleranced features. *Artificial Intelligence*, 37(13):403 – 412, 1988.

[12] Doriane Gouyou, Yann Ledoux, Denis Teissandier, and Vincent Delos. Tolerance analysis of overconstrained and flexible assemblies by polytopes and finite element computations: application to a flange. *Research in Engineering Design*, 29(1):55–66, Jan 2018.

[13] Martin Peternell and Tibor Steiner. Minkowski sum boundary surfaces of 3d-objects. *Graphical Models*, 69(34):180 – 190, 2007.

[14] Gokul Varadhan and Dinesh Manocha. Accurate minkowski sum approximation of polyhedral models. *Graphical Models*, 68(4):343 – 355, 2006.

[15] Yanyan Wu, Jami J. Shah, and Joseph K. Davidson. Improvements to algorithms for computing the minkowski sum of 3-polytopes. *Computer-Aided Design*, 35(13):1181 – 1192, 2003.

[16] Efi Fogel and Dan Halperin. Exact and efficient construction of minkowski sums of convex polyhedra with applications. *Computer-Aided Design*, 39(11):929 – 940, 2007.

[17] Christophe Weibel. *Minkowski sums of polytopes: combinatorics and computation*. PhD thesis, École Polytechnique Fédérale De Lausanne, 2007.

[18] Komei Fukuda. From the zonotope construction to the minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261 – 1272, 2004.

[19] Vincent Delos and Denis Teissandier. Minkowski sum of polytopes defined by their vertices. *Journal of Applied Mathematics and Physics (JAMP)*, 3(1):62–67, January 2015.

[20] Denis Teissandier and Vincent Delos. Algorithm to calculate the Minkowski sums of 3-polytopes based on normal fans. *Computer-Aided Design*, 43:1567–1576, 2011.

[21] Vincent Delos and Denis Teissandier. Minkowski sum of HV-polytopes in Rn. In *4th Annual International Conference on Computational Mathematics, Computational Geometry and Statistics*, Singapore, Singapore, January 2015.

[22] A. Clément, A. Rivière, and P. Serré. *A Declarative Information Model for Functional Requirements*, pages 3–16. Springer Netherlands, Dordrecht, 1996.

[23] Max Giordano and Daniel Duret. Clearance space and deviation space. In *3rd. CIRP Seminars on Computer Aided Tolerancing, ENS Cachan, France, Apr*, pages 27–28, 1993.

[24] David Avis, David Bremner, and Raimund Seidel. How good are convex hull algorithms? *Computational Geometry*, 7(5):265 – 301, 1997. 11th ACM Symposium on Computational Geometry.

[25] Vincent Delos, Denis Teissandier, and Santiago Arroyave-Tobón. How to trace the significant information in tolerance analysis with polytopes. In *Lecture Notes in Mechanical Engineering*, volume 2, pages 1003–1012, Catania, Italy, September 2016. Springer International Publishing.

[26] Komei Fukuda and Vera Rosta. Combinatorial face enumeration in convex polytopes. *Computational Geometry*, 4(4):191 – 198, 1994.

[27] Paul Clozel and Pierre-Alain Rance. Mecamaster: a tool for assembly simulation from early design, industrial approach. *Geometric Tolerancing of Products*, pages 241–273, 2010.