GeoInformatica manuscript No.
(will be inserted by the editor)

A Framework for Evaluating 3D Topological Relations based on a Vector Data Model

Alberto Belussi · Sara Migliorini · Mauro Negri

Received: date / Accepted: date

Abstract 3D topological relations are commonly used for testing or imposing the existence of desired properties between objects of a dataset, such as a city model. Currently available GIS systems usually provide a limited 3D support which usually includes a set of 3D spatial data types together with few operations and predicates, while limited or no support is generally provided for 3D topological relations. Therefore, an important problem to face is how such relations can be actually implemented by using the constructs already provided by the available systems. In this paper, we introduce a generic 3D vector model which includes an abstract and formal description of the 3D spatial data types and of the related basic operations and predicates that are commonly provided by GIS systems. Based on this model, we formally demonstrate how these limited sets of operations and predicates can be combined with 2D topological relations for implementing 3D topological relations.

Keywords 3D topological relations · topological relation implementation · spatial constraint validation · vector model

1 Introduction

In recent years many 3D modeling techniques have been applied for describing urban environments in different application contexts and with different goals. In particular, among other approaches, the OGC Standard CityGML [9] and its UML data model have been used for the representation of buildings located inside a common urban environment, requiring the explicit description

Alberto Belussi · Sara Migliorni
Department of Computer Science – University of Verona (Italy)
E-mail: sara.migliorini@univr.it

Mauro Negri
Department of Electronics, Information and Bioengineering – Politecnico of Milan (Italy)

of the spatial relations existing between each building and the environment and between buildings themselves.

This new scenario exploits 3D technologies in a new and challenging way, since it forces the designers of 3D solutions to deal with relationships among solids or solids and other geometries (like, surfaces or curves), while in the recent past 3D modeling techniques were applied only for the representation of a single construction particularly valuable in terms of the cultural heritage of a city or country, like an ancient church or palace. In the latter case the goal of the 3D technique was mainly to produce a 3D visualization of the construction that was as close as possible to its actual appearance in the reality. This approach places less emphasis on the overall correctness of the represented spatial objects, in particular as regards to the spatial relations between them and their parts or between them and the environment. However, as soon as we switch from a single construction context to a metropolitan area of hundreds or thousands of building, i.e. houses, palaces, churches, shopping centers, industries and so on, the type of required computation becomes completely different and more focused on queries or complex processing which exploit the topological structure of the set of buildings. In this case, the visualization is only one of the issues to face, and the correctness of the geometrical representation of the objects and their spatial relations become a key point for many applications.

In this new scenario, the modeling of 3D topological relations and the implementation of procedures for testing them is crucial for different reasons, for instance: (i) for expressing spatial integrity constraints and validating them, as in the approach proposed in [18] and extended in [3, 2] that allows the definition of spatial integrity constraints inside UML data models through a set of predefined OCL template; (ii) for extracting geometries satisfying given conditions, like in the execution of a spatial query in a spatial database system; (iii) in the processing of huge amount of 3D data in a cluster system for extracting spatial correlations between 3D geometries and mining new knowledge, as done in [15]. In particular, in this paper the work presented in [2] finds its completion, since the testing procedures presented here allows to automatically validate the topological integrity constraints specified by means of the proposed constraint templates. For example, in a CityGML like data model, we might need to specify the constraint exemplified in the following example.

Example 1 In a city model all buildings shall be disjoint or touch each other. Moreover, if a building consists of only one (homogeneous) part, it shall be represented by a unique solid element. Otherwise, if it is composed by several individual structures, it shall be modeled as a set of solid parts, such that all these parts touch each other to form a composite solid, see Fig. 1. With reference to CityGML, each building part must be related to exactly one building and it must touch it.

As regards to the reference set of topological relations to be considered, we adopt the extension to 3D geometries [5] of the well-known 9-intersection model [6]. Despite some attempts have been made for providing an implementation for them (as reported in Sect. 2), a feasible and complete solution

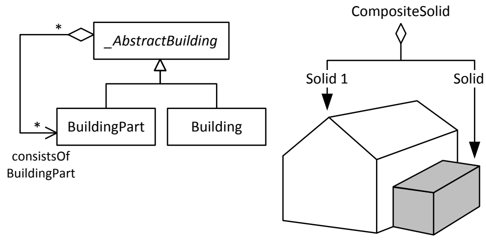


Fig. 1 Example of a building composed of two parts that touch each other.

has not been obtained yet on the current technology, namely a solution that covers all possible cases relatively to all the geometric types and possible configurations in the embedding space. Indeed, the development of 3D operators is actually a challenging issue and requires careful consideration about the dimensionality of objects (more specifically their geometric type) and their finite representation which is usually based on collection of geometric primitives, i.e. building blocks used for the definition of an object with complex shape.

Currently all mainstream GIS systems, such as PostGIS [17], provide a support for the Simple Features Access (SFA) model defined by the OpenGIS Consortium (OGC) [8]. Notice that, beside to the OGC SFA model published in 2011 which includes a definition for 3D types and operations, there exists also an ISO version of the same model, known as ISO Standard 19125 [11], which has been published in 2004 and includes only the specification for 2D types and operations. At now, a working group has been defined for including the support of 3D coordinates also in the ISO Standard 19125. More specifically, as regards to the 3D space, PostGIS [17] supports some 3D geometric types, e.g. 3D polygons and polyhedral surfaces, but a complete support for generic solids is still missing. Indeed, as defined in the OGC SFA model, a solid can be represented by a closed polyhedral surface describing its external boundary, while a multi-surface containing a set of closed polyhedral surfaces can be used to describe a solid with holes. However, only few operators are realized on these supported 3D data types, while the majority still works only on the 2D projection of them. Even the CGAL extension of PostGIS, called SFCGAL[1], which allows the representation of generic solids with holes, has no algorithm supporting them, as explicitly stated by the developers. In this situation, the specification of testing procedures for 3D topological relations may be challenging and highly dependent on the system considered for representing 3D geometries. For these reasons, the contribution presented in this paper is twofolds:

The definition of a *3D vector model* ($\mathcal{V}al3d$) – it is an abstract model that defines a set of basic types (describing primitives), which, starting from vertices, allows the representation in vector format of geometries belonging to the ISO Standards 19136 GML [12], together with the set of operations that

[1] `http://www.sfcgal.org/`

are necessary to navigate through the primitives of a complex object or to test basic predicates on primitives. We consider the ISO Standard 19136 since it is more known and widespread than the ISO 19107 Spatial Schema [10], however it is the ISO 19107 that defines the complete set of spatial types that are adopted by the family of standards denoted as ISO TC 211 Geographic information/Geomatics.

The definition of a set of *procedures for testing 3D topological relations* – given the vector model and the representation in that model of the GML types: gml::LineString, gml::LinearRing, gml::Polygon, gml::PolyhedralSurface and gml::Solid by means of sets of primitives, the implementation of the topological relations is presented and the proof of their completeness with respect to all possible cases is shown. These cases are explicitly enumerated in tables representing all possible interaction between primitives.

More specifically, this paper extends the work in [18] to the 3D space, in particular as regards to surfaces and volumes. The same idea applied to only few cases was first presented in [3] in order to investigate the applicability of the approach. In this paper we consider all possible combinations in terms of 3D types and demonstrate how a topological relation between them can be implemented using a small set of basic predicates and operations of the vector model. These operations are commonly available in current GIS systems, such as PostGIS, in addition to 2D topological relations, and this confirms that the procedures for testing the relations can be actually implemented with current technology. A prototypical implementation of such relations together with the required small set of additional procedures can be found in [1].

2 Related Work

Several works can be found in literature about the implementation of spatial 3D operators and topological relations on currently available GIS systems. In particular, many of them can be traced back to the work of Zlatanova et al, starting from its original categorization of possible 3D topological relations between multidimensional simple objects in 3D space [25]. This formalization has been performed by considering the 9-intersection model and the results presented in [5] and by identifying a set of negative conditions capturing situations that cannot be realized in reality. Topology is one of the mechanisms used to describe relationships between spatial objects, namely it is at the basis of many spatial operations. In [26] the author give an overview of the 3D topological models and frameworks presented in the literature, and discuss generic issues related to 3D modeling. Among all possible frameworks, here we consider the 9-intersection model originally proposed in [6]. In particular, the work in [25] together with [5] is at the basis of the 3D topological relation definitions in Tab. 1.

In order to discuss the implementation of 3D operations, such as 3D XOR, 3D union, 3D intersection, and 3D difference between solids, in [19] the authors develop a new 3D data type, the polyhedron, and provide an implementation of such operations on it using a geoDBMS. Subsequently, in [13] the authors consider the representation of the ISO Standard GM_Solid inside PostGIS and the implementation of the intersection operation between two solids. The work has been further extended in [24], where the authors develop the two 3D operators overlap and meet between solids, reusing existing 2D operators: starting from 2D topological relations between the 2D projections and then using some 3D operators. The approach is similar to the one proposed in this paper, but here we provide a more detailed and formalized approach, based on the formal definition of a generic vector model and the implementation of all topological relations, considering also other 3D spatial data types.

An extended study about the testing of topological relations involving generic solids with holes can be found in [14]. The author recognizes the gap between the formal and detailed definitions provided in the ISO Standard 19107 w.r.t. what is supported by currently available GIS systems. The treatment focuses mainly on the lack of support for 3D operations on such types. In particular, in order to study the relations existing between the various parts of a solid (the shells), the author makes use of the notion of Nef polyhedron [4, 7]. This work has been concretized into the realization of an open-source software, called *val3dity*, to validate 3D primitives accordingly to the ISO Standard 19107. The tool is based on the CGAL library and at now is not integrated with existing systems: it is currently provided as a stand-alone command line program.

In [23] the authors present a methodology to model and implement 3D geo-constraints based on four steps: natural language, geometric/topological abstractions, UML/OCL formulations and SQL implementations. As regards to the SQL implementation, the author considers as starting point the SDO_AnyInteract operation provided by Oracle Spatial and manually implement a new function 3D_SurfaceRelate for distinguishing what happens in the non-disjoint part. In [22] the author provides a set of domain-specific constraints for a Climate City Campus Database described using CityGML. Examples of constraints are the distance between buildings and trees, or between aquatic plant and water. Such constraints are provided in OCL and translated into ad-hoc spatial queries for Oracle. In [20] the authors propose an ad hoc implementation of the provided constraints through a Java Tool, while the constraints instantiated by means of the OCL templates that we propose are automatically translated into SQL spatial queries. The fundamental difference between the approach proposed in this paper and these last approaches is that instead of producing ad-hoc procedures from scratch whenever needed, we try to provide a generic solution by exploiting the operations already implemented in the currently available GIS systems.

3 Geometric Model

This section introduces the set of spatial data types and the formal specification of the topological relations of interest. In particular, for this set of relations a possible implementation will be provided in the following section using as building blocks a limited set of commonly available 3D operations together with 2D topological relations.

This paper refers to the 3D geometric types of the ISO Standard 19136 GML [12], which in turn are compliant with the ISO Standard 19107 Spatial Schema [10], while the set of possible topological relations existing between them are defined in terms of the well-known 9-intersection model [6]. More specifically, it considers a subset of the GML data types which are the most commonly used during the representation of a 3D urban scene. The chosen classes make the approach sufficiently generic without unnecessarily increasing the complexity of the treatment. In particular, the paper considers: (i) line-strings as implementation of curves (ii) polyhedral surfaces and polygons as implementations of surfaces, and (iii) solids defined by means of closed polyhedral surfaces.

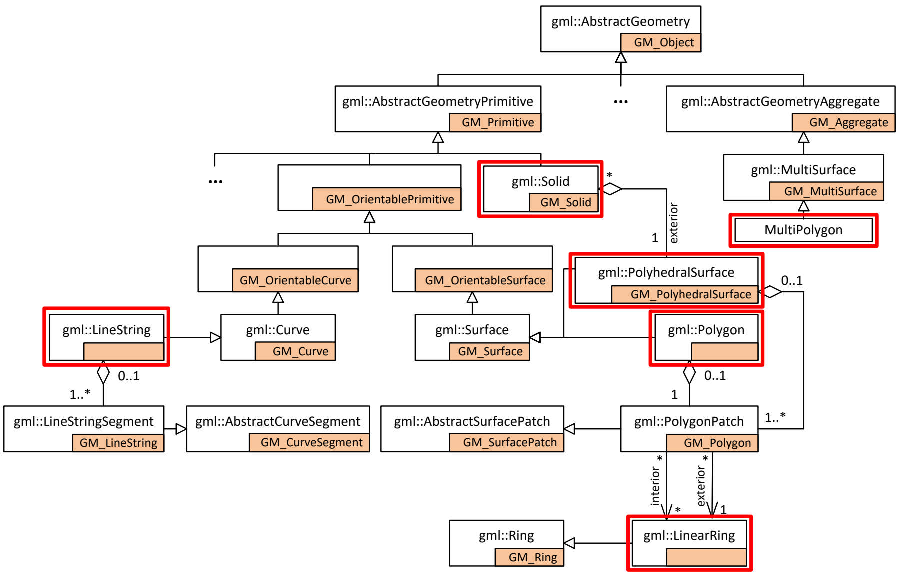


Fig. 2 Hierarchy of spatial data types, the red boxes highlight the concrete classes considered in the paper.The name with the prefix *gml* is the class contained in the ISO Standard 19136 GML, while the name in the small coloured box is the corresponding class in the ISO Standard 19107.

Fig. 2 summarizes the hierarchy of considered data types, reporting both the name used in the GML specification (with prefix *gml*::) and the name of the corresponding class in the ISO Standard 19107 (small coloured inner box). Inside the hierarchy, the set of considered concrete classes are surrounded by an additional red border. Details about the characteristics of such spatial data types can be found in [12,2], here we recall only some additional restriction made to simply the following definition of 3D topological relations. More specifically, without loss of generality the following simplications are introduced w.r.t. the standard.

Assumption 1 (LineString) *Adjacent collinear segments are not admitted in a LineString, since they can be replaced by the segment obtained by merging them.*

Assumption 2 (PolyhedralSurfaces) *Adjacent coplanar patches are not admitted in PolyhedralSurfaces, since they can be replaced by the patch obtained by merging them.*

Assumption 3 (Solid) *A solid is characterized by only one external boundary surface which is a PolyhedralSurface, and zero internal boundaries. It follows that such kind of solid object has no holes (i.e., enclaves). This simplification is required by the model in [5] for the specification of topological relations in 3D. As discussed in [2], it keeps simple the definition of topological relations involving solids without reducing the generality of the model. A deep discussion about that is given below.*

Assumption 3 does not limit the generality of the model, since a solid with holes can be: (1) replaced by a set of adjacent solids obtained by splitting it into two or more parts without holes, as suggested in [21], or (2) described by a set of solids, where the first one is its external envelope and the other ones are the holes. This last representation is possible because the ISO Standard 19107 prescribes that the shells representing the internal boundary of a solid has to satisfy the Jordan Separation Theorem, namely each shell divides the space into exactly two regions, one bounded and one unbounded. As regards to the testing of topological relations on a solid with holes, in both cases it can be translated into a corresponding set of tests on its constituent parts/components. For instance, let us consider case (2) where a solid with holes sh is represented by a solid s describing its external part and a set of other solids $h_1, \dots h_n$ describing the holes. In order to check if a generic geometry g is contained in s, we can check if (i) g is contained in s and (ii) for all holes h_i, g is disjoint from h_i (or g touches h_i, provided that g is not completely contained in the boundary of h_i). Given such possibilities, we can omit the case of solid with holes in order to keep the treatment simpler.

Each considered class provides a specific formal definition of the concept of *boundary*, *interior* and *exterior* in terms of the point sets topology. These concepts define a partition of the space in which an object is embedded: intuitively, the boundary separates the interior of an object from the outer space, which represents its exterior. Such subdivision produces three point sets that are used to formally specify a reference set of topological relations.

The 9-intersection model (9IM) [6] is the most common model for defining *Binary Topological Relations*. It specifies the topological relation R existing between two objects A and B considering the intersection between their interior

(A°, B°), boundary (∂A, ∂B) and exterior (A^-, B^-).

$$R(A,B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

Table 1 3D topological relations between solids (V), surfaces (S), curves (C). Possible topological relations are disjoint (DJ), touch (TC), in (IN), contains (CN), equal (EQ), overlap (OV). The matrix patterns are specified as $1st\ row - 2nd\ row - 3rd\ row$. Used symbols are: T = not empty, F = empty, $*$ = any result, $\mathbf{T}$ = always not empty for the considered combination of geometric types, $T_\odot$ = not empty when the geometries, for which the boundary is considered, are not cycles (e.g., rings are cycles), empty otherwise, T_∂ = not empty, but in the case in which the boundary of the first geometry (e.g., a solid) is equal to the second one (e.g. a surface). Finally, A^T denotes the transpose of a matrix A.

Rel.	**Definition**	**Geom.**	**Matrix Pattern**
DJ	$A \cap B = \emptyset$	V/V	$FFT - FFT - TT\mathbf{T}$
		V/S	$FF\mathbf{T} - FFT - TT_\odot\mathbf{T}$
		V/C	$FF\mathbf{T} - FF\mathbf{T} - TT_\odot\mathbf{T}$
		S/V, C/V	DJ(V/S)^T, DJ(V/C)^T
		S/S, C/C	$FFT - FFT_\odot - TT_\odot\mathbf{T}$
		S/C	$FF\mathbf{T} - FFT_\odot - TT_\odot\mathbf{T}$
		C/S	DJ(S/C)^T
TC	$(A^\circ \cap B^\circ = \emptyset) \wedge$ $(A \cap B \neq \emptyset)$	V/V	$FFT - FTT - TT\mathbf{T}$
		V/S	$FF\mathbf{T} - T*T_\partial - **\mathbf{T} \cup$ $FF\mathbf{T} - FTT - T*\mathbf{T}$
		V/C	$FF\mathbf{T} - T*\mathbf{T} - **\mathbf{T} \cup$ $FF\mathbf{T} - FT\mathbf{T} - T*\mathbf{T}$
		S/V, C/V	TC(V/S)^T, TC(V/C)^T
		S/S, C/C	$FTT - *** - T*\mathbf{T} \cup$ $FFT - T** - T*\mathbf{T} \cup$ $FFT - FT* - T*\mathbf{T}$
		S/C	$FT\mathbf{T} - *** - T*\mathbf{T} \cup$ $FF\mathbf{T} - T** - T*\mathbf{T} \cup$ $FF\mathbf{T} - FT* - T*\mathbf{T}$
		C/S	TC(S/C)^T
IN	$(A \cap B = A) \wedge$ $(A^\circ \cap B^\circ \neq \emptyset)$	V/V	$TFF - T*F - TT\mathbf{T}$
		S/S, C/C	$TFF - **F - TT_\odot\mathbf{T}$
		S/V	$T*F - **F - \mathbf{T}T\mathbf{T}$
		C/V	$T*F - **F - \mathbf{TTT}$
		C/S	$T*F - **F - \mathbf{T}T_\odot\mathbf{T}$
CN	$(A \cap B = B) \wedge$ $(A^\circ \cap B^\circ \neq \emptyset)$	V/V	IN(V/V)^T
		S/S, C/C	IN(S/S)^T, IN(C/C)^T
		V/S	IN(S/V)^T
		V/C	IN(C/V)^T
		S/C	IN(C/S)^T
EQ	$A = B$	V/V, S/S, C/C	$TFF - FTF - FF\mathbf{T}$
OV	$(A^\circ \cap B^\circ \neq \emptyset) \wedge$ $(A \cap B \neq A) \wedge$ $(A \cap B \neq B)$	V/V	$TTT - TTT - TT\mathbf{T}$
		V/S	$T*\mathbf{T} - T*T_\partial - T*\mathbf{T}$
		V/C	$T*\mathbf{T} - T*\mathbf{T} - T*\mathbf{T}$
		S/S, C/C	$T*T - *** - T*\mathbf{T}$
		S/C	$T*\mathbf{T} - **T_\partial - T*\mathbf{T}$
		S/V, C/V, C/S	OV(V/S)^T, OV(V/C)^T, OV(S/C)^T

The topological relations described above apply to primitive types and can be extended to aggregate geometries by imposing some constraints on their components, as formalized in [8] and done in available systems such as PostGIS. For instance, the polygons composing a *MultiPolygon* (as defined in [8]) cannot overlap. Such constraints do not reduce the expressive power of the type, namely the kind of representable objects, since each generic aggregate can be translated into one that satisfy the given constraints.

Tab. 1 reports the formal definition of the topological relations considered in the sequel by means of template specifications. Such templates are obtained by grouping several configurations of the matrix of the 9IM and they are usually implemented in current GIS systems at least for 2D geometries. For each relation, the table shows a name, together with the specification of the pair of geometric types to which it applies, and the corresponding configurations of the 9-intersection matrix representing scenes where the relation exists, table caption contains more details about the formalism used for representing matrix configurations.

4 Vector Model

Currently available GIS systems usually provide a limited support for 3D spatial data which includes the definition of the geometric types introduced in the previous section, together with a limited set of basic operations. Conversely, the implementation of 3D topological relations is typically not directly provided. This is for instance the situation in PostGIS 3.0 [17] that is able to represent the main 3D types of interest, such as: *LineStrings*, *PolyhedralSurfaces* and indirectly *Solids* by means of closed *PolyhedralSurfaces*, on which a limited set of 3D spatial operations are defined. In order to keep the proposed validation framework as independent as possible from a particular implementation, in this section we introduce a generic vector model characterized by a discrete representation of the types in Sect. 3, together with a set of basic operations, that are necessary for the evaluation of 3D topological relations.

Definition 1 The validation framework $\mathcal{V}al3d$ is based on a set of **basic vector types** defined in terms of *vertices*, represented as triples of finite numbers in a reference system. The basic vector types are: vertex, segment, ring and patch. Tab. 2 shows their formal definition and vector representation. □

Definition 2 Given a geometry g, its **3D vector representation** is defined as in Tab. 3 in terms of the introduced basic vector types, where v denotes a generic vertex, s a generic segment, p a generic patch and r a generic ring. Notice that different equivalent representations (denoted as $VR_x(g)$) can be defined according to the primitives ($x \in \{\mathsf{vertex}, \mathsf{surface}, \mathsf{patch}, \mathsf{ring}\}$) used as building blocks. □

Given the basic vector types in Def. 1 and the vector representation of the geometric types in Def. 2, the $\mathcal{V}al3d$ framework provides a set of basic operations and predicates. In particular, for each of them it specifies: one or more

Table 2 Basic vector types of the $\mathcal{V}al3d$ framework: v denotes a generic vertex, s a generic segment, p a generic patch and r a generic ring. $\mathbf{VR}_v()$ is the vector representation in terms of vertices, while $\mathbf{VR}_s()$ is the vector representation in terms of segments.

Primitive	$\mathbf{VR}_v()$	$\mathbf{VR}_s()$
vertex v_0	v_0	–
It is a tuple of finite numbers representing a 3D coordinate: $v = (x, y, z)$.		
segment s_0	(v_1, v_2)	s_0
It is a pair of vertices and it represents the segment obtained by considering the linear interpolation between them.		
ring r_0	$(v_1, \ldots, v_n)$	$(s_1, \ldots, s_{n-1})$ where $s_i = (v_i, v_{i+1})$
It is a list of vertices, its linear interpolation represents a ring $(v_1 = v_n)$.		
patch p	$((v_{1,1}, \ldots, v_{1,n_1}), \ldots,$ $(v_{k,1}, \ldots, v_{k,n_k}))$ $k > 1 \wedge n_i > 3$	$((s_{1,1}, \ldots, s_{1,n_1-1}), \ldots,$ $(s_{k,1}, \ldots, s_{k,n_k-1}))$
It is a finite portion of a plane whose external boundary is defined by a ring $(v_{1,1}, \ldots, v_{1,n_1})$ and its internal boundaries, if exist, are defined by a list of rings $((v_{2,1}, \ldots, v_{2,n_2}), \ldots, (v_{k,1}, \ldots, v_{k,n_k}))$. Internal boundaries, when exists, define the holes of the patch. Notice that the vertices of the patch are coplanar.		

Table 3 3D vector representation of a geometry g. $\mathbf{VR}_v()$ is the vector representation in terms of vertices, $\mathbf{VR}_s()$ is the vector representation in terms of segments and $\mathbf{VR}_p()$ is the vector representation in terms of patches.

Geom. Type	$\mathbf{VR}_v()$	$\mathbf{VR}_s()$	$\mathbf{VR}_p()$
line-string	$(v_1, \ldots, v_n)$	$(s_1, \ldots, s_{n-1})$ with $s_i = (v_i, v_{i+1})$	–
linear-ring	Its representation is equal to the one proposed for line-string with the additional constraints that it is closed and simple.		
polygon	$((v_{1,1}, \ldots, v_{1,n_1}), \ldots,$ $(v_{k,1}, \ldots, v_{k,n_k}))$ with $k > 1 \wedge n_i > 3$	$((s_{1,1}, \ldots, s_{1,m_1}), \ldots,$ $(s_{k,1}, \ldots, s_{k,m_k}))$ with $m_i = n_i - 1 \wedge$ $s_{i,j} = (v_{i,j}, v_{i,j+1})$ $(1 \leq i \leq k)$	p
polyhedral-surface	$(((v^1_{1,1}, \ldots, v^1_{1,n^1_1}), \ldots,$ $(v^1_{r_1,1}, \ldots, v^1_{r_1,n^1_{r_1}})), \ldots,$ $((v^k_{1,1}, \ldots, v^k_{1,n^k_1}), \ldots,$ $(v^k_{r_k,1}, \ldots, v^k_{r_k,n^k_{r_k}})))$ with $k > 1 \wedge$ $r_i > 1 \ \wedge \ n^i_j > 2$ $(1 \leq i \leq k, 1 \leq j \leq r_i)$	$(((s^1_{1,1}, \ldots, s^1_{1,m_{1,1}}), \ldots,$ $(s^1_{r_1,1}, \ldots, s^1_{r_1,m^1_{r_1}})), \ldots,$ $((s^k_{1,1}, \ldots, s^k_{1,m_{k,1}}), \ldots,$ $(s^k_{r_k,1}, \ldots, s^k_{r_k,n^k_{r_k}})))$ with $m^i_j = n^i_j - 1$ and $s^i_{j,l} = (v^i_{j,l}, v^i_{j,l+1})$, $(1 \leq i \leq k, 1 \leq j \leq r_i)$	$\{p_1, \ldots, p_k\}$
multi-polygon	Its representation is equal to that proposed for polyhedral-surface, without any constraints among the patches.		
solid	Its representation is equal to that proposed for polyhedral-surface, with the constraint that the overall surface is *simple*, i.e. it has no self-intersections, and is a *cycle*, i.e. it has empty boundary, thus being topologically closed.		

domains for the parameters (the set of objects where the operation/predicate applies) and the target domain (the set of produced objects). The possible domains are: vertex, segment, ring, patch (referenced together as primitive), line-string, linear-ring (referenced together as curve), polygon, multi-polygon, polyhedral-surface (referenced together as surface) and solid; the domain geometry is the union of all previous domains ($\mathsf{geometry} = \mathsf{primitive} \cup \mathsf{curve} \cup \mathsf{surface} \cup \mathsf{solid}$).

Definition 3 The set of **basic vector operations** provided by the $\mathcal{V}al3d$ framework are the following ones. The symbol $\wp(S)$ is used to denote the power set of a set S.

- $\mathsf{vert} : \mathsf{geometry} \to \wp(\mathsf{vertex})$
  $g.\mathsf{vert}() = \{v_i \mid v_i \in \mathbf{VR}_v(g)\}$ and $v.\mathsf{vert}() = \{v\}$
  It returns the set of vertices defining the geometry g according to its vector representation.
- $\mathsf{seg} : \mathsf{geometry} \to \wp(\mathsf{segment})$
  $g.\mathsf{seg}() = \{s_i \mid s_i \in \mathbf{VR}_s(g)\}$, $v.\mathsf{seg}() = \emptyset$ and $s.\mathsf{seg}() = \{s\}$
  It returns the set of segments defining the geometry g according to its vector representation.
- $\mathsf{pat} : \mathsf{geometry} \to \wp(\mathsf{patch})$
  $g.\mathsf{pat}() = \{p_i \mid p_i \in \mathbf{VR}_p(g)\}$
  $v.\mathsf{pat}() = s.\mathsf{pat}() = r.\mathsf{pat}() = \emptyset$ and $p.\mathsf{pat}() = \{p\}$.
  It returns the set of patches defining the geometry g, according to its vector representation.
- $\mathsf{bnd} : \mathsf{curve} \to \wp(\mathsf{vertex})$
  if $\mathbf{VR}_v(cv) = (v_1, \ldots, v_n) \wedge v_1 = v_n$ then $cv.\mathsf{bnd}() = \emptyset$
  if $\mathbf{VR}_v(cv) = (v_1, \ldots, v_n) \wedge v_1 \neq v_n$ then $cv.\mathsf{bnd}() = \{v_1, v_n\}$
  It returns the set of vertices defining the boundary of the curve cv.
- $\mathsf{bnd} : \mathsf{surface} \to \wp(\mathsf{linear\text{-}ring})$
  $sf.\mathsf{bnd}() = \mathsf{buildRings}(\{s_i \mid s_i \in \mathbf{VR}_s(sf) \wedge \exists! p \in \mathbf{VR}_p(sf)(s_i \in p.\mathsf{seg}())\})$
  It returns the set of linear rings defining the boundary of the surface sf. The function $\mathsf{buildRings}()$, starting from a set of segments, groups the segments to produce lists of segments composing simple rings.
- $\mathsf{bnd} : \mathsf{solid} \to \mathsf{polyhedral\text{-}surface}$
  $sd.\mathsf{bnd}() = \{p_i \mid p_i \in \mathbf{VR}_p(sd)\}$
  It returns the polyhedral surface defining the boundary of the solid sd.
- $\mathsf{intSeg} : \mathsf{surface} \to \wp(\mathsf{segment})$
  $sf.\mathsf{intSeg}() = sf.\mathsf{seg}() \setminus \{s_i \mid s_i \in \mathbf{VR}_s(sf) \wedge \exists! p \in \mathbf{VR}_p(sf)(s_i \in p.\mathsf{bnd}())\}$
  It returns the set of segments defining the patches of the surface, but that do not belong to its boundary.
- $\mathsf{intVert} : \mathsf{surface} \to \wp(\mathsf{segment})$
  $sf.\mathsf{intVert}() = sf.\mathsf{vert}() \setminus \{v_i \mid v_i \in \mathbf{VR}_v(sf) \wedge \nexists r \in sf.\mathsf{bnd}()(v_i \in r.\mathsf{vert}())\}$
  It returns the set of vertices belonging to the patches of the surface, but that do not belong to its boundary.
- $\mathsf{ray}_3 : \mathsf{vertex} \cup \mathsf{segment} \times \mathsf{solid} \to \textsc{integer}$
  $g.\mathsf{ray}_3(sd) = |\{p | p \in sd.\mathsf{pat}() \wedge p \cup \mathsf{semi\text{-}straight\text{-}line}(g) \neq \emptyset\}|$
  It returns the number of patches of $sd.\mathsf{pat}()$ that are intersected by the semi-straight line starting from $g.\mathsf{start}()$ and passing through $g.\mathsf{end}()$ (function $\mathsf{semi\text{-}straight\text{-}line}(g)$), when g is a vertex v the semi-straight line starting in $v = (x_v, y_v, z_v)$ with equation $y = y_v$, $z = z_v$ and $x > x_v$ is considered. When $g.\mathsf{start}()$ lies on the solid boundary, i.e it belongs to the point set of

one of the patches of $sd.\mathsf{pat}()$, the result of ray_3 is fixed to zero, i.e. in this case $g.\mathsf{start}()$ is considered outside the solid.

- $\mathsf{mid} : \mathsf{vertex} \times \mathsf{vertex} \rightarrow \mathsf{vertex}$
  $v.\mathsf{mid}(v_0)$ returns the vertex v_m representing the midpoint of the segment having endpoints in v and v_0
- $\cap_3 : \mathsf{segment} \times \mathsf{segment} \rightarrow \mathsf{vertex}$
  $s.\cap_3(s_0)$ returns the vertex v_{int} representing the 3D intersection between s and s_0; when the segments do not intersect, it returns the empty geometry.

□

Definition 4 The set of **basic vector predicates** provided by the $\mathcal{V}al3d$ are the following ones. The symbol $\mathcal{PS}(g)$ is used to denote the point-set representation of generic geometry g.

- $\mathsf{eq}_3 : \mathsf{geometry} \times \mathsf{geometry} \rightarrow \text{BOOLEAN}$
  $g.\mathsf{eq}_3(g_0) \equiv \mathsf{type}(g) = \mathsf{type}(g_0) \wedge VR_v(g) = VR_v(g_0)$
  It tests the equality between two geometries, two geometries are equal only if they have the same type and an identical vector representation.
- $\mathsf{cnt}_3 : \mathsf{segment} \times \mathsf{vertex} \rightarrow \text{BOOLEAN}$
  $s.\mathsf{cnt}_3(v) \equiv \text{true if } v \in \mathcal{PS}(s) \wedge v \notin s.\mathsf{vert}()$
  It tests the containment between a vertex and the interior of a segment.
- $\mathsf{cnt}_3 : \mathsf{patch} \times \mathsf{vertex} \rightarrow \text{BOOLEAN}$
  $p.\mathsf{cnt}_3(v) \equiv \text{true if } v \in \mathcal{PS}(p) \wedge v \notin p.\mathsf{vert}() \wedge \neg\exists s \in p.\mathsf{seg}()(s.\mathsf{cnt}_3(v))$
  It tests the containment between a vertex and the interior of a patch.
- $\mathsf{int}_3 : \mathsf{patch} \times \mathsf{segment} \rightarrow \text{BOOLEAN}$
  $p.\mathsf{int}_3(s) \equiv \text{true if } \exists! v \in \mathcal{V}\ (\mathcal{PS}(v) \in \mathcal{PS}(p) \wedge$
  $\quad (v \notin p.\mathsf{vert}() \wedge \neg\exists s_i \in p.\mathsf{seg}()(\mathcal{PS}(v) \in \mathcal{PS}(s_i)) \wedge$
  $\quad \mathcal{PS}(v) \in \mathcal{PS}(s) \wedge v \notin s.\mathsf{vert}())$
  It tests the intersection between the interior of a patch and the interior of a segment: if the intersection is a single point, then it returns true, otherwise it returns false.
- $\mathsf{int}_3 : \mathsf{patch} \times \mathsf{patch} \rightarrow \text{BOOLEAN}$
  $p_1.\mathsf{int}_3(p_2) \equiv \text{true if } \exists! s \in \mathcal{S}($
  $\quad \mathcal{PS}(s) \subset \mathcal{PS}(p_1) \wedge$
  $\quad \neg\exists s_1 \in p_1.\mathsf{seg}()(\mathcal{PS}(s) \subseteq \mathcal{PS}(s_1) \vee \mathcal{PS}(s_1) \subseteq \mathcal{PS}(s)) \wedge$
  $\quad \mathcal{PS}(s) \subset \mathcal{PS}(p_2) \wedge$
  $\quad \neg\exists s_2 \in p_2.\mathsf{seg}()\mathcal{PS}(s) \subseteq \mathcal{PS}(s_2) \vee \mathcal{PS}(s_2) \subseteq \mathcal{PS}(s))$
  It tests the intersection between the interior of two patches: if the intersection is a segment, then it returns true, otherwise it returns false.
- $\mathsf{cop} : \mathsf{patch} \cup \mathsf{segment} \times \mathsf{segment} \rightarrow \text{BOOLEAN}$
  $s_0.\mathsf{cop}(s) \equiv \text{true if } \exists! pl \in \mathcal{P}(\mathcal{PS}(s_0) \in \mathcal{PS}(pl) \wedge \mathcal{PS}(s) \in \mathcal{PS}(pl))$
  $p.\mathsf{cop}(s) \equiv \text{true if } \exists! pl \in \mathcal{P}(\mathcal{PS}(p) \in \mathcal{PS}(pl) \wedge \mathcal{PS}(s) \in \mathcal{PS}(pl))$
  They test the coplanarity between the patch p (or the segment s_0) and the segment s.
- $\mathsf{cop} : \mathsf{patch} \times \mathsf{patch} \rightarrow \text{BOOLEAN}$
  $p_0.\mathsf{cop}(p) \equiv \text{true if } \exists! pl \in \mathcal{P}(\mathcal{PS}(p_0) \in \mathcal{PS}(pl) \wedge \mathcal{PS}(p) \in \mathcal{PS}(pl))$
  It tests the coplanarity between the patches p and p_0.

– $\langle \mathsf{rel} \rangle_2$: $\mathsf{geometry} \times \mathsf{geometry} \to$ BOOLEAN
  $g.\langle \mathsf{rel} \rangle_2(g_0)$ tests the topological relation $\langle \mathsf{rel} \rangle \in \{\mathsf{DJ}, \mathsf{TC}, \mathsf{IN}, \mathsf{CN}, \mathsf{OV}, \mathsf{CR}\}$ between the projection of the geometries g and g_0 on the 2D plane where both geometries lie. The semantics of these relations is well known and their formal definition is reported in the second column of Table 1, notice that only the crosses (CR) relation is not reported, but it is a specialization of the overlaps (OV) relation that requires a 1-dimensional intersection among geometry interiors. □

The 8 basic vector operations in Def. 3 and the 7 basic vector predicates of Def. 4, together with the 2D topological relation tests already available in current GISs, are all the tools necessary for implementing the considered 3D topological relations. Before showing how such implementations can be obtained starting from this limited set of constructs, the following section discusses all possible scenarios that can occur between the components of the 3D vector representation of two given geometric types.

5 Relations between the components of two vector representations

In this section we discuss the possible scenarios between two geometries that have to be considered during the evaluation of a 3D topological relation. In particular, we will examine the possible relations that can exist between the components of the vector representation of two geometries.

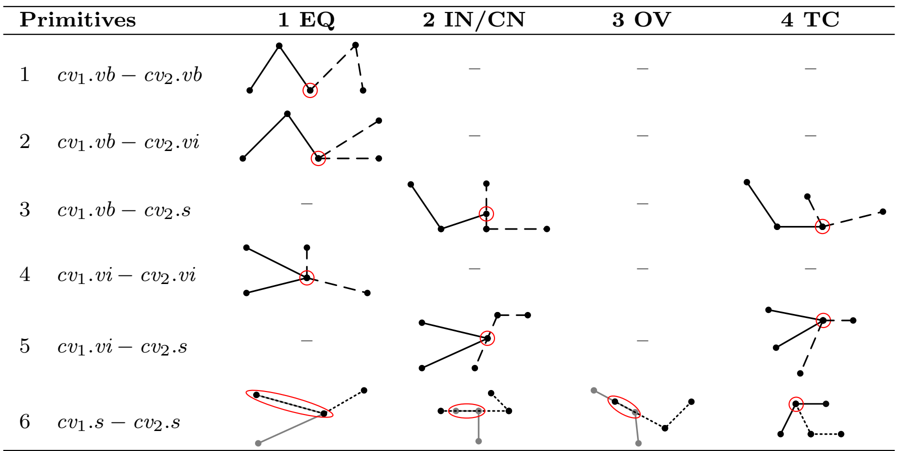


Fig. 3 All possible scenarios to be considered to evaluate the existence of a topological relation between two curves. The following symbols are used: cv curve, $cv.vi$ ($cv.vb$) an internal (or boundary) vertex of a curve cv and $cv.s$ a segment of a curve cv.

Proposition 1 (VR(cv) – VR(cv) relations) *Let $cv_1, cv_2 \in \mathsf{curve}$ two geometries whose vector representation in terms of vertices is denoted as $\boldsymbol{VR}_v(*)$*

and in terms of segments as $\boldsymbol{VR}_s(*)$*, respectively. The possible scenarios to be considered in the evaluation of a topological relation between* cv_1 *and* cv_2 *are the ones reported in Fig. 3 where* $cv.vi \in \boldsymbol{VR}_v(cv)$ *denotes an internal vertex of* cv*,* $cv.vb \in \boldsymbol{VR}_v(cv)$ *is a boundary vertex of* cv*, and* $cv.s \in \boldsymbol{VR}_s(cv)$ *is a segment of* cv*.* □

Proof Given a curve cv, its vector representation in terms of vertices $\mathbf{VR}_v(cv)$ is given by a sequence of vertices $(v_1, \ldots, v_n)$. For the purpose of implementing topological relations, we can distinguish between the vertices that belongs to the curve boundary, denoted as vb_k, and the internal boundaries, denoted as vi_k. Similarly, its vector representation in terms of segments $\mathbf{VR}_s(cv)$ is given by a list of segments $(s_1, \ldots, s_n)$.

In order to identify all possible scenarios, it is necessary to consider all possible relations that can exist between the components of the vector representations of cv_1 and cv_2:

– $\forall(v_1, v_2) \in \mathbf{VR}_v(cv_1) \times \mathbf{VR}_v(cv_2) \implies R_{topo}(v_1, v_2) \in \{\mathsf{DJ}, \mathsf{EQ}\}$.
– $\forall(v, s) \in \mathbf{VR}_v(cv_1) \times \mathbf{VR}_s(cv_2) \implies R_{topo}(v, s) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
– $\forall(s, v) \in \mathbf{VR}_s(cv_1) \times \mathbf{VR}_v(cv_2) \implies R_{topo}(s, v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
– $\forall(s_1, s_2) \in \mathbf{VR}_s(cv_1) \times \mathbf{VR}_s(cv_2) \implies R_{topo}(s_1, s_2) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}/\mathsf{CN}, \mathsf{TC}\}$

These scenarios are exactly the ones reported in Tab. 3, except for DJ which is not included for not cluttering the presentation.

Proposition 2 ($\mathbf{VR}(cv) - \mathbf{VR}(sf)$ **relations)** *Let* $cv \in$ curve *and* $sf \in$ surface *two geometries whose vector representation in terms of vertices is denoted as* $\boldsymbol{VR}_v(*)$*, in terms of segments as* $\boldsymbol{VR}_s(*)$ *and in terms of patches as* $\boldsymbol{VR}_p(*)$*, respectively. The possible scenarios to be considered in the evaluation of a topological relation between* cv *and* sf *are the ones reported in Fig. 4 where* $cv.vi \in \boldsymbol{VR}_v(cv)$ *(or* $sf.vi \in \boldsymbol{VR}_v(sf)$*) denotes an internal vertex of* cv *(or* sf*),* $cv.vb \in \boldsymbol{VR}_v(cv)$ *(or* $sf.vb \in \boldsymbol{VR}_v(sf)$*) is a boundary vertex of* cv *(or* sf*),* $cv.s \in \boldsymbol{VR}_s(cv)$ *is a segment of* cv*,* $sf.si \in \boldsymbol{VR}_s(sf)$ *is an internal segment of* sf*,* $sf.sb \in \boldsymbol{VR}_s(sf)$ *is a boundary segment of* sf*, and* $sf.p \in \boldsymbol{VR}_p(sf)$ *is a patch of* sf*.* □

Proof Given a curve cv, its vector representation in terms of vertices $\mathbf{VR}_v(cv)$ is given by a sequence of vertices $(v_1, \ldots, v_n)$. For the purpose of implementing topological relations, we can distinguish between the vertices that belongs to the curve boundary, denoted as vb_k, and the internal boundaries, denoted as vi_k. Similarly, its vector representation in terms of segments $\mathbf{VR}_s(cv)$ is given by a list of segments $(s_1, \ldots, s_n)$.

Given a surface sf, its vector representation in terms of vertices $\mathbf{VR}_v(sf)$ is given by a list of vector tuples, while its vector representation in terms of segments $\mathbf{VR}_s(sf)$ is a list of segment tuples, and its vector representation in terms of patch $\mathbf{VR}_p(sf)$ is a list of patches, as reported in Tab. 3. As regards to vertices and segments, we can distinguish between internal vertices (or segments), denoted as $vi_{k,j}$ (or $si_{k,j}$), and boundary vertices (or segments), denoted as $vb_{k,j}$ (or $sb_{k,j}$).

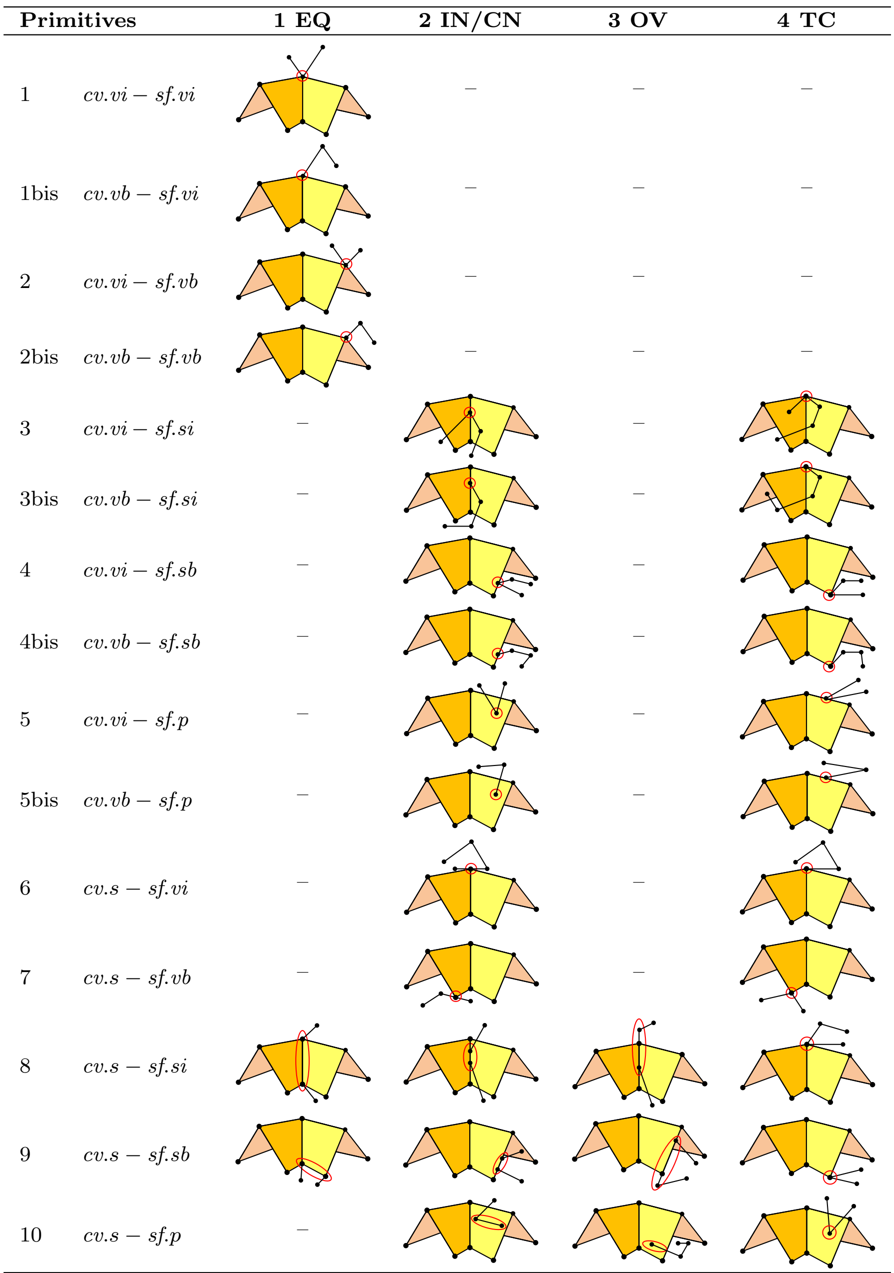


Fig. 4 All possible scenarios to be considered to evaluate the existence of a topological relation between curves and surfaces. The following symbols are used: *cv* is a curve and *sf* is a surface, *vi* (*vb*) is an internal (or boundary) vertex of a surface or curve, *s* is a segment, *si* (*sb*) is an internal (or boundary) segment of surfaces, *p* is a patch of a surface.

The possible relations that can exist between the components of the vector representations of cv and sf are:

- $\forall(v_1, v_2) \in \mathbf{VR}_v(cv) \times \mathbf{VR}_v(sf) \implies R_{topo}(v_1, v_2) \in \{\mathsf{DJ}, \mathsf{EQ}\}$.
- $\forall(v, s) \in \mathbf{VR}_v(cv) \times \mathbf{VR}_s(sf) \implies R_{topo}(v, s) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(v, p) \in \mathbf{VR}_v(cv) \times \mathbf{VR}_p(sf) \implies R_{topo}(v, p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(s, v) \in \mathbf{VR}_s(cv) \times \mathbf{VR}_v(sf) \implies R_{topo}(v, p) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(s_1, s_2) \in \mathbf{VR}_s(cv_1) \times \mathbf{VR}_s(cv_2) \implies R_{topo}(s_1, s_2) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(s_1, p) \in \mathbf{VR}_s(cv_1) \times \mathbf{VR}_p(cv_2) \implies R_{topo}(s_1, p) \in \{\mathsf{DJ}, \mathsf{IN}/\mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$

These scenarios are exactly the ones reported in Tab. 4, except for DJ which is not included for not cluttering the presentation.

Proposition 3 ($\mathbf{VR}(cv) - \mathbf{VR}(sd)$ relations) *Let $cv \in$ curve and $sd \in$ solid two geometries whose vector representation in terms of vertices is denoted as $\boldsymbol{VR}_v(*)$, in terms of segments as $\boldsymbol{VR}_s(*)$ and in terms of patches a $\boldsymbol{VR}_p(*)$, respectively. The possible scenarios to be considered in the evaluation of a topological relation between cv and sd are the ones reported in Fig. 5 where $cv.vi \in \boldsymbol{VR}_v(cv)$ denotes an internal vertex of cv, $cv.vb \in \boldsymbol{VR}_v(cv)$ is a boundary vertex of cv, $sd.v \in \boldsymbol{VR}_v(sd)$ is a vertex of sd, $cv.s \in \boldsymbol{VR}_s(cv)$ (or $sd.s \in \boldsymbol{VR}_s(sd)$) is a segment of sd, $sd.p \in \boldsymbol{VR}_p(sd)$ is a patch of sd, and vl is used to refer to the solid interior.* □

Proof Given a curve cv, its vector representation in terms of vertices $\mathbf{VR}_v(cv)$ is given by a sequence of vertices $(v_1, \ldots, v_n)$. For the purpose of implementing topological relations, we can distinguish between the vertices that belongs to the curve boundary, denoted as vb_k, and the internal boundaries, denoted as vi_k. Similarly, its vector representation in terms of segments $\mathbf{VR}_s(cv)$ is given by a list of segments $(s_1, \ldots, s_n)$.

Given a solid sd, its vector representation in terms of vertices $\mathbf{VR}_v(sd)$ is given by a list of vector tuples, while its vector representation in terms of segments $\mathbf{VR}_s(sd)$ is a list of segment tuples, and its vector representation in terms of patch $\mathbf{VR}_p(sd)$ is a list of patches, as reported in Tab. 3. In order to identify all possible scenarios, it is necessary to consider also the interior of a solid, denoted as vl.

The possible relations that can exist between the components of the vector representations of cv and sd are:

- $\forall(v_1, v_2) \in \mathbf{VR}_v(cv) \times \mathbf{VR}_v(sd) \implies R_{topo}(v_1, v_2) \in \{\mathsf{DJ}, \mathsf{EQ}\}$.
- $\forall(v, s) \in \mathbf{VR}_v(cv) \times \mathbf{VR}_s(sd) \implies R_{topo}(v, s) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(v, p) \in \mathbf{VR}_v(cv) \times \mathbf{VR}_p(sd) \implies R_{topo}(v, p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(s, v) \in \mathbf{VR}_s(cv) \times \mathbf{VR}_v(sd) \implies R_{topo}(s, v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(s_1, s_2) \in \mathbf{VR}_s(cv) \times \mathbf{VR}_s(sd) \implies R_{topo}(s_1, s_2) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(s, p) \in \mathbf{VR}_s(cv) \times \mathbf{VR}_p(sd) \implies R_{topo}(s_1, p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(v, vl) \in \mathbf{VR}_v(cv) \times sd.vl \implies R_{topo}(v, vl) \in \{\mathsf{DJ}, \mathsf{IN}\}$
- $\forall(s, vl) \in \mathbf{VR}_s(cv) \times sd.vl \implies R_{topo}(s_1, s_2) \in \{\mathsf{DJ}, \mathsf{IN}\}$

These scenarios are exactly the ones reported in Tab. 5, except for DJ which is not included for not cluttering the presentation.

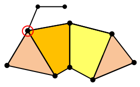
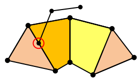
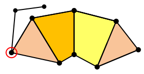
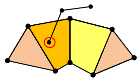
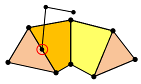
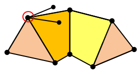
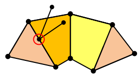
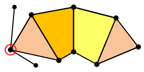
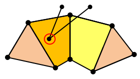
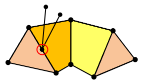
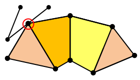
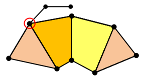
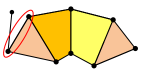
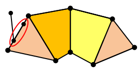
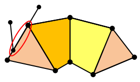
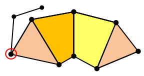
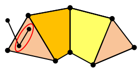
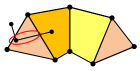
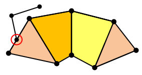
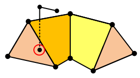
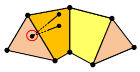
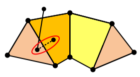
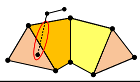
Primitives		1 EQ	2 IN/CN	3 OV	4 TC
1	$cv.vb - sd.v$		–	–	–
2	$cv.vb - sd.s$	–		–	
3	$cv.vb - sd.p$	–		–	
4	$cv.vi - sd.v$		–	–	–
5	$cv.vi - sd.s$	–		–	
6	$cv.vi - sd.p$	–		–	
7	$cv.s - sd.v$	–		–	
8	$cv.s - sd.s$				
9	$cv.s - sd.p$	–			
10	$cv.vb - sd.vl$	–		–	–
11	$cv.vi - sd.vl$	–		–	–
12	$cv.s - sd.vl$	–			–

Fig. 5 All possible scenarios to be considered to evaluate the existence of a topological relation between a curve and a solid. The following symbols are used: *sd* solid, *cv* curve, *cv.s*, *cv.vi* and *cv.vb* a segment, an internal vertex or a boundary vertex of a curve *cv*, *sd.p*, *sd.s* and *sd.v* a patch, a segment, a vertex of a solid *sd*. Finally, *sd.vl* denotes the interior of a solid *sd* and dashed lines are used to represent segments inside a solid.

Proposition 4 (VR(sf) − VR(sf) relations) *Let $sf_1, sf_2 \in$ surface two geometries whose vector representation in terms of vertices is denoted as $\boldsymbol{VR}_v(*)$, in terms of segments as $\boldsymbol{VR}_s(*)$ and in terms of patches a $\boldsymbol{VR}_p(*)$, respectively. The possible scenarios to be considered in the evaluation of a topological relation between sf_1 and sf_2 are the ones reported in Fig. 6-7 where*

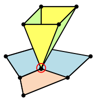
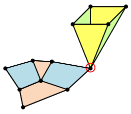
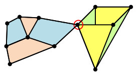
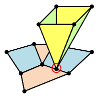
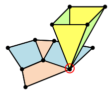
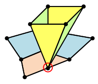
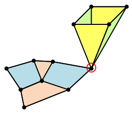
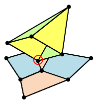
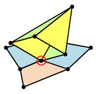
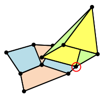
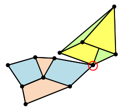
Primitives		**1 EQ**	**2 IN/CN**	**3 OV**	**4 TC**
1	$sf_1.vi - sf_2.vi$		–	–	–
2	$sf_1.vi - sf_2.vb$		–	–	–
3	$sf_1.vb - sf_2.vi$	as 2	–	–	–
4	$sf_1.vb - sf_2.vb$		–	–	–
5	$sf_1.vi - sf_2.si$	–		–	
6	$sf_1.vi - sf_2.sb$	–		–	
7	$sf_1.vb - sf_2.si$	–		–	
8	$sf_1.vb - sf_2.sb$	–		–	
9	$sf_1.si - sf_2.vi$	–	as 5	–	as 5
10	$sf_1.si - sf_2.vb$	–	as 7	–	as 7
11	$sf_1.sb - sf_2.vi$	–	as 6	–	as 6
12	$sf_1.sb$ - $sf_2.vb$	–	as 8	–	as 8

Fig. 6 All possible scenarios to be considered to evaluate the existence of a topological relation between two surfaces. The following symbols are used: *sf* surface, $sf.vi$ and $sf.vb$ an internal vertex or a boundary vertex of a surface *sf*, $sf.si$ and $sf.sb$ an internal vertex or a boundary segment of a surface *sf*, and $sf.p$, a patch of a surface *sf* (continue in Fig. 7).

$sf.vi \in \boldsymbol{VR}_v(sf)$ denotes an internal vertex of sf, $sf.vb \in \boldsymbol{VR}_v(sf)$ is a boundary vertex of sf, $sf.si \in \boldsymbol{VR}_s(sf)$ is an internal segment of sf, $sf.sb \in \boldsymbol{VR}_s(sf)$ is a boundary segment of sf, and $sf.p \in \boldsymbol{VR}_p(sf)$ is a patch of sf. □

Proof Given a surface *sf*, its vector representation in terms of vertices $\mathbf{VR}_v(sf)$ is given by a list of vector tuples, while its vector representation in terms of segments $\mathbf{VR}_s(sf)$ is a list of segment tuples, and its vector representation in terms of patch $\mathbf{VR}_p(sf)$ is a list of patches, as reported in Tab. 3. For the evaluation of topological relations, vertices and segments of a surface are

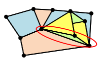
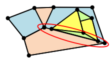
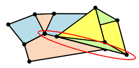
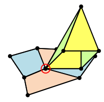
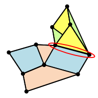
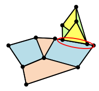
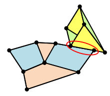
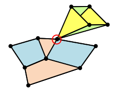
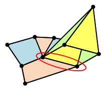
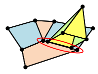
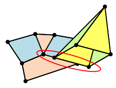
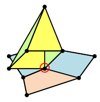
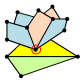
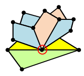
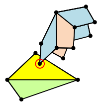
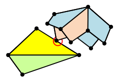
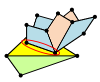
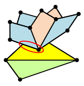
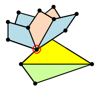
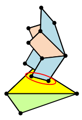
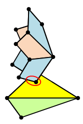
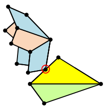
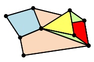
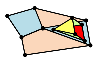
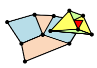
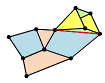
	Primitives	**1 EQ**	**2 IN/CN**	**3 OV**	**4 TC**
13	$sf_1.si - sf_2.si$				
14	$sf_1.si - sf_2.sb$				
15	$sf_1.sb - sf_2.si$	as 14	as 14	as 14	as 14
16	$sf_1.sb - sf_2.sb$				
17	$sf_1.p$ - $sf_2.vi$	–		–	
18	$sf_1.p - sf_2.vb$	–		–	
19	$sf_1.p - sf_2.si$	–			
20	$sf_1.p - sf_2.sb$	–			
21	$sf_1.p - sf_2.p$				
22	$sf_1.vi - sf_2.p$	–	as 17	–	as 17
23	$sf_1.vb - sf_2.p$	–	as 18	–	as 18
24	$sf_1.si$ - $sf_2.p$	–	as 19	as 19	as 19
25	$sf_1.sb$ - $sf_2.p$	–	as 20	as 20	as 20

Fig. 7 All possible scenarios to be considered to evaluate the existence of a topological relation between two surfaces (continue from Fig. 6).

distinguished between internal and boundary and denoted as $sf.vi$ (or $sf.vb$) and $sf.si$ (or $sf.sb$), respectively.

The possible relations that can exist between the components of the vector representations of sf_1 and sf_2 are:

- $\forall(v_1, v_2) \in \mathbf{VR}_v(sf_1) \times \mathbf{VR}_v(sf_2) \implies R_{topo}(v_1, v_2) \in \{\mathsf{DJ}, \mathsf{EQ}\}$.
- $\forall(v, s) \in \mathbf{VR}_v(sf_1) \times \mathbf{VR}_s(sf_2) \implies R_{topo}(v, s) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$

- $\forall(v,p) \in \mathbf{VR}_v(sf_1) \times \mathbf{VR}_p(sf_2) \implies R_{topo}(v,p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(s,v) \in \mathbf{VR}_s(sf_1) \times \mathbf{VR}_v(sf_2) \implies R_{topo}(s,v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(s_1,s_2) \in \mathbf{VR}_s(sf_1) \times \mathbf{VR}_s(sf_2) \implies R_{topo}(s_1,s_2) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(s,p) \in \mathbf{VR}_s(sf_1) \times \mathbf{VR}_p(sf_2) \implies R_{topo}(s,p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(p,v) \in \mathbf{VR}_p(sf_1) \times \mathbf{VR}_v(sf_2) \implies R_{topo}(p,v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(p,s) \in \mathbf{VR}_p(sf_1) \times \mathbf{VR}_s(sf_2) \implies R_{topo}(p,s) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(p_1,p_2) \in \mathbf{VR}_p(sf_1) \times \mathbf{VR}_p(sf_2) \implies R_{topo}(s,p) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$

These scenarios are exactly the ones reported in Tab. 6-7, except for DJ which is not included for not cluttering the presentation.

Proposition 5 ($\mathbf{VR}(sf)-\mathbf{VR}(sd)$ relations) *Let $sf \in$ surface and $sd \in$ solid two geometries whose vector representation in terms of vertices is denoted as $\boldsymbol{VR}_v(*)$, in terms of segments as $\boldsymbol{VR}_s(*)$ and in terms of patches a $\boldsymbol{VR}_p(*)$, respectively. The possible scenarios to be considered in the evaluation of a topological relation between sf and sd are the ones reported in Fig. 8-9 where $sf.vi \in \boldsymbol{VR}_v(sf)$ denotes an internal vertex of sf, $sf.vb \in \boldsymbol{VR}_v(sf)$ is a boundary vertex of sf, $sd.v \in \boldsymbol{VR}_v(sd)$ is a vertex of sd, $sf.si \in \boldsymbol{VR}_s(sf)$ is an internal segment of sf, $sf.sb \in \boldsymbol{VR}_s(sf)$ is a boundary segment of sf, $sd.s \in \boldsymbol{VR}_s(sd)$ is a segment of sd, $sf.p \in \boldsymbol{VR}_p(sf)$ (or $sd.p \in \boldsymbol{VR}_p(sd)$) is a patch of sf (or sd), and $sd.vl$ denotes the solid interior.* □

Proof Given a surface sf (or a solid sd), its vector representation in terms of vertices $\mathbf{VR}_v(sf)$ (or $\mathbf{VR}_v(sd)$) is given by a list of vector tuples, while its vector representation in terms of segments $\mathbf{VR}_s(sf)$ or $(\mathbf{VR}_s(sd))$ is a list of segment tuples, and its vector representation in terms of patches $\mathbf{VR}_p(sf)$ (or $\mathbf{VR}_p(sd)$) is a list of patches, as reported in Tab. 3. For the evaluation of topological relations, vertices and segments of a surface are distinguished between internal and boundary, and denoted as $sf.vi$ (or $sf.vb$) and $sf.si$ (or $sf.sb$), respectively; while the interior of a solid is denoted as vl.

The possible relations that can exist between the components of the vector representations of sf and sd are:

- $\forall(v_1,v_2) \in \mathbf{VR}_v(sf) \times \mathbf{VR}_v(sd) \implies R_{topo}(v_1,v_2) \in \{\mathsf{DJ}, \mathsf{EQ}\}$.
- $\forall(v,s) \in \mathbf{VR}_v(sf) \times \mathbf{VR}_s(sd) \implies R_{topo}(v,s) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(v,p) \in \mathbf{VR}_v(sf) \times \mathbf{VR}_p(sd) \implies R_{topo}(v,p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(s,v) \in \mathbf{VR}_s(sf) \times \mathbf{VR}_v(sd) \implies R_{topo}(s,v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(s_1,s_2) \in \mathbf{VR}_s(sf) \times \mathbf{VR}_s(sd) \implies R_{topo}(s_1,s_2) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(s,p) \in \mathbf{VR}_s(sf) \times \mathbf{VR}_p(sd) \implies R_{topo}(s,p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(p,v) \in \mathbf{VR}_p(sf) \times \mathbf{VR}_v(sd) \implies R_{topo}(p,v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(p,s) \in \mathbf{VR}_p(sf) \times \mathbf{VR}_s(sd) \implies R_{topo}(p,s) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(p_1,p_2) \in \mathbf{VR}_p(sf) \times \mathbf{VR}_p(sd) \implies R_{topo}(s,p) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(v,vl) \in \mathbf{VR}_v(sf) \times sd.vl \implies R_{topo}(v,vl) \in \{\mathsf{DJ}, \mathsf{IN}\}$
- $\forall(s,vl) \in \mathbf{VR}_s(sf) \times sd.vl \implies R_{topo}(s,vl) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}\}$
- $\forall(p,vl) \in \mathbf{VR}_s(sf) \times sd.vl \implies R_{topo}(s,vl) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}\}$

These scenarios are exactly the ones reported in Tab. 8-9, except for DJ which is not included for not cluttering the presentation.

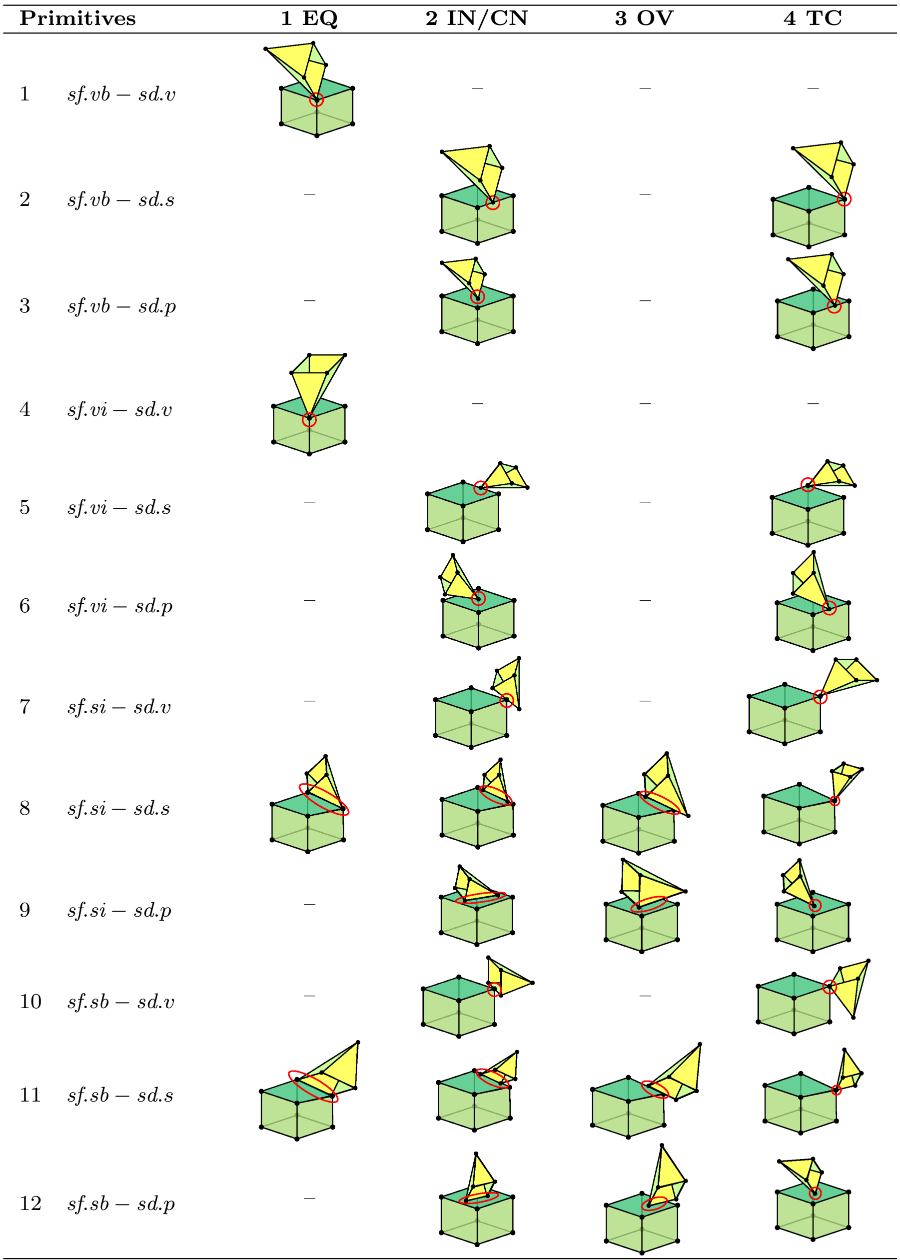


Fig. 8 All possible scenarios to be considered to evaluate the existence of a topological relation between a surface and a solid. The following symbols are used: *sd* solid, *sf* surface, *sf.p*, *sf.si*, *sf.sb*, *sf.vi* and *cv.vb* a patch, internal segment, boundary segment, internal vertex or boundary vertex of a surface *sf* and *sd.p*, *sd.s* and *sd.v* a patch, a segment, a vertex of a solid *sd*. Finally, *sd.vl* indicates the interior of a solid *sd* (continue. in Fig. 9)

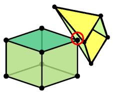
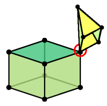
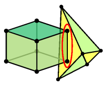
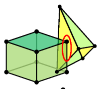
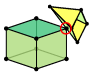
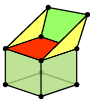
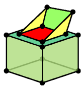
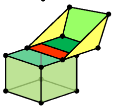
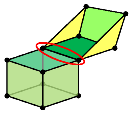
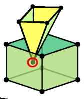
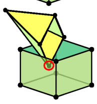
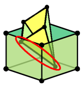
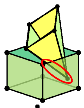
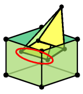
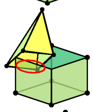
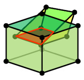
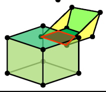
Primitives	1 EQ	2 IN/CN	3 OV	4 TC
13 $sf.p - sd.v$	–		–	
14 $sf.p - sd.s$	–			
15 $sf.p - sd.p$				
16 $sf.vi - sd.vl$	–		–	–
17 $sf.vb - sd.vl$	–		–	–
18 $sf.si - sd.vl$	–			–
19 $sf.sb - sd.vl$	–			–
20 $sf.p - sd.vl$	–			–

Fig. 9 All possible scenarios to be considered to evaluate the existence of a topological relation between a surface and a solid (continue from Fig. 8).

Proposition 6 ($\mathbf{VR}(sd) - \mathbf{VR}(sd)$ relations) *Let $sd_1, sd_2 \in \mathsf{solid}$ two geometries whose vector representation in terms of vertices is denoted as $\boldsymbol{VR}_v(*)$, in terms of segments as $\boldsymbol{VR}_s(*)$ and in terms of patches a $\boldsymbol{VR}_p(*)$, respectively. The possible scenarios to be considered in the evaluation of a topological relation between sd_1 and sd_2 are the ones reported in Fig. 10 where $sd.v \in \boldsymbol{VR}_v(sd)$ is a vertex of sd, $sd.s \in \boldsymbol{VR}_s(sd)$ is a segment of sd, $sd.p \in \boldsymbol{VR}_p(sd)$ is a patch of sd, and $sd.vl$ denotes the solid interior.* □

Proof Given a solid sd, its vector representation in terms of vertices $\mathbf{VR}_v(sd)$ is given by a list of vector tuples, while its vector representation in terms of segments $\mathbf{VR}_s(sd)$ is a list of segment tuples, and its vector representation in terms of patch or $\mathbf{VR}_p(sd)$ is a list of patches, as reported in Tab. 3. For the evaluation of topological relations it is necessary to consider also the interior of a solid, denoted as vl.

The possible relations that can exist between the components of the vector representations of sd_1 and sd_2 are:

– $\forall (v_1, v_2) \in \mathbf{VR}_v(sd_1) \times \mathbf{VR}_v(sd_2) \implies R_{topo}(v_1, v_2) \in \{\mathsf{DJ}, \mathsf{EQ}\}$

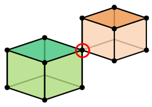
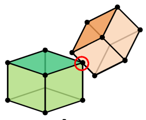
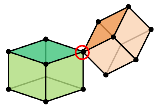
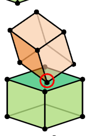
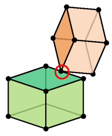
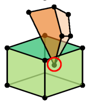
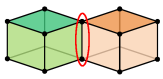
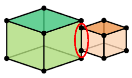
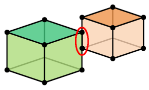
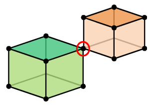
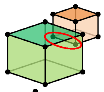
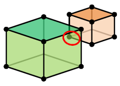
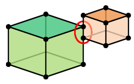
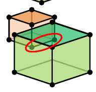
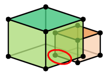
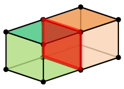
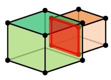
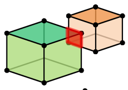
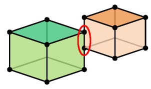
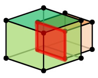
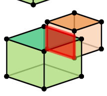
Primitives		**1 EQ**	**2 IN/CN**	**3 OV**	**4 TC**
1	$sd_1.v - sd_2.v$		–	–	–
2	$sd_1.v - sd_2.s$	–		–	
3	$sd_1.v - sd_2.p$	–		–	
4	$sd_1.v - sd_2.vl$	–		–	–
5	$sd_1.s - sd_2.v$	–	as 2	–	as 2
6	$sd_1.s - sd_2.s$				
7	$sd_1.s - sd_2.p$	–			
8	$sd_1.s - sd_2.vl$	–			–
9	$sd_1.p - sd_2.v$	–	as 3	–	as 3
10	$sd_1.p - sd_2.s$	–	as 7	as 7	as 7
11	$sd_1.p - sd_2.p$				
12	$sd_1.p - sd_2.vl$	–			–
13	$sd_1.vl - sd_2.v$	–	as 4	–	–
14	$sd_1.vl - sd_2.s$	–	as 8	as 8	–
15	$sd_1.vl - sd_2.p$	–	as 12	as 12	–

Fig. 10 All possible scenarios to be considered to evaluate the existence of a topological relation between two solids. The following symbols are used: *sd* solid, *sd.p*, *sd.s*, and *sd.v* a patch, segment, or vertex of a solid *sd*. Finally, *sd.vl* indicates the interior of a solid *sd*.

- $\forall(v,s) \in \mathbf{VR}_v(sd_1) \times \mathbf{VR}_s(sd_2) \implies R_{topo}(v,s) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(v,p) \in \mathbf{VR}_v(sd_1) \times \mathbf{VR}_p(sd_2) \implies R_{topo}(v,p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{TC}\}$
- $\forall(s,v) \in \mathbf{VR}_s(sd_1) \times \mathbf{VR}_v(sd_2) \implies R_{topo}(s,v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(s_1,s_2) \in \mathbf{VR}_s(sd_1) \times \mathbf{VR}_s(sd_2) \implies R_{topo}(s_1,s_2) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(s,p) \in \mathbf{VR}_s(sd_1) \times \mathbf{VR}_p(sd_2) \implies R_{topo}(s,p) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(p,v) \in \mathbf{VR}_p(sd_1) \times \mathbf{VR}_v(sd_2) \implies R_{topo}(p,v) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{TC}\}$
- $\forall(p,s) \in \mathbf{VR}_p(sd_1) \times \mathbf{VR}_s(sd_2) \implies R_{topo}(s,p) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(p_1,p_2) \in \mathbf{VR}_p(sd_1) \times \mathbf{VR}_p(sd_2) \implies R_{topo}(p_1,p_2) \in \{\mathsf{DJ}, \mathsf{EQ}, \mathsf{IN}, \mathsf{CN}, \mathsf{OV}, \mathsf{TC}\}$
- $\forall(v,vl) \in \mathbf{VR}_v(sd_1) \times sd_2.vl \implies R_{topo}(v,vl) \in \{\mathsf{DJ}, \mathsf{IN}\}$
- $\forall(s,vl) \in \mathbf{VR}_s(sd_1) \times sd_2.vl \implies R_{topo}(s,vl) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}\}$
- $\forall(p,vl) \in \mathbf{VR}_p(sd_1) \times sd_2.vl \implies R_{topo}(p,vl) \in \{\mathsf{DJ}, \mathsf{IN}, \mathsf{OV}\}$
- $\forall(vl,v) \in sd_1.vl \times \mathbf{VR}_v(sd_2) \implies R_{topo}(vl,v) \in \{\mathsf{DJ}, \mathsf{CN}\}$
- $\forall(vl,s) \in sd_1.vl \times \mathbf{VR}_s(sd_2) \implies R_{topo}(vl,s) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{OV}\}$
- $\forall(vl,p) \in sd_1.vl \times \mathbf{VR}_p(sd_2) \implies R_{topo}(vl,p) \in \{\mathsf{DJ}, \mathsf{CN}, \mathsf{OV}\}$

These scenarios are exactly the ones reported in Tab. 10, except for DJ which is not included for not cluttering the presentation.

6 Topological Relation Tests for 3D Geometries

Tab. 1 in Sect. 3 contains six topological relations, each one corresponding to a set of matrix configurations due to the different definition of boundary and interior for each geometric type. In order to simplify the presentation, this section starts by showing the implementation of the tests regarding the significant matrix cells, i.e.: $A^\circ \cap B^\circ$, $A^\circ \cap \partial B$, $A^\circ \cap B^-$, $\partial A \cap \partial B$ and $\partial A \cap B^-$. Then, the implementation of each topological relation test can be obtained by specifying a conjunction of a subset of these tests. For not cluttering the discussion, only the proof of Prop. 7 is given, while the proof for Prop. 8-11 can be obtained in a similar way and are reported in App. A.

Proposition 7 *The* ***Interior/Interior test*** *($A^\circ \cap B^\circ$) has to be specialized for each combination of the geometric types of the involved geometries A and B. Tab. 4-5 report the tests regarding the cases: curve/curve, curve/surface, curve/solid, surface/surface, surface/solid and solid/solid. They are expressed by means of the operations and predicates of Def. 3 and Def. 4, respectively.*

Proof The proof shows that each specialization of the intersection tests ($IT^{\circ\circ}$) reported in Tab. 4-5 cover all the possible scenarios between A and B. The complete enumeration of all scenarios is obtained by considering the geometric primitives that compose A and B and the relations among them that imply the truth of the considered test. These tests are denoted as follows:

$$IT^{\circ\circ}_{type(g_1),type(g_2)}(g_1, g_2)$$

where g_1, g_2 are two geometries and $type(g) \in \{cv, sf, sd\}$ denotes curve, surface and solid, respectively.

Table 4 Implementation of tests for Interior/Interior intersection $IT^{\circ\circ}$ for the cases curve/curve, curve/surface, curve/solid. To simplify reading, the OR logical operators connecting alternative conditions have been highlighted with a gray background and consecutive lines have been numbered with the same font color.

Test	#L	Implementation
$IT^{\circ\circ}_{cv,cv}$		$IT^{\circ\circ}_{cv,cv}(cv_1, cv_2) \equiv$
	1.	$\exists v \in cv_1.\mathsf{vert}()(v \notin cv_1.\mathsf{bnd}() \wedge v \in cv_2.\mathsf{vert}() \wedge v \notin cv_2.\mathsf{bnd}())$ $\vee$
	2.	$\exists v \in cv_1.\mathsf{vert}()(v \notin cv_1.\mathsf{bnd}() \wedge \exists s \in cv_2.\mathsf{seg}()(s.\mathsf{cnt}_3(v)))$ $\vee$
	3.	$\exists v \in cv_2.\mathsf{vert}()(v \notin cv_2.\mathsf{bnd}() \wedge \exists s \in cv_1.\mathsf{seg}()(s.\mathsf{cnt}_3(v)))$ $\vee$
	4.	$\exists s_1 \in cv_1.\mathsf{seg}()(\exists s_2 \in cv_2.\mathsf{seg}()(s_1.\mathsf{eq}_3(s_2) \vee$
	5.	$(s_1.\mathsf{cop}(s_2) \wedge (s_1.\mathsf{in}_2(s_2) \vee s_2.\mathsf{in}_2(s_1) \vee s_1.\mathsf{ov}_2(s_2)))))$
$IT^{\circ\circ}_{cv,sf}$		$IT^{\circ\circ}_{cv,sf}(cv, sf) \equiv$
	1.	$\exists s \in cv.\mathsf{seg}()(\exists p \in sf.\mathsf{pat}()(p.\mathsf{int}_3(s) \vee (p.\mathsf{cop}(s) \wedge \neg p.\mathsf{dj}_2(s) \wedge \neg p.\mathsf{tc}_2(s))))$ $\vee$
	2.	$\exists s \in cv.\mathsf{seg}()(\exists s_0 \in sf.\mathsf{intSeg}()(s.\mathsf{cop}(s_0) \wedge \neg s.\mathsf{dj}_2(s_0) \wedge \neg s.\mathsf{tc}_2(s_0)))$ $\vee$
	3.	$\exists s \in cv.\mathsf{seg}()(\exists v_0 \in sf.\mathsf{intVert}()($
		$s.\mathsf{cnt}_3(v_0) \vee v_0.\mathsf{eq}_3(s.\mathsf{start}()) \vee v_0.\mathsf{eq}_3(s.\mathsf{end}())))$ $\vee$
	4.	$\exists v \in cv.\mathsf{vert}()(v \notin cv.\mathsf{bnd}() \wedge (\exists p \in sf.\mathsf{pat}()(p.\mathsf{cnt}_3(v))))$ $\vee$
	5.	$\exists v \in cv.\mathsf{vert}()(v \notin cv.\mathsf{bnd}() \wedge$
		$(\exists s_0 \in sf.\mathsf{intSeg}()(s_0.\mathsf{cnt}_3(v)) \vee \exists v_0 \in sf.\mathsf{intVert}()(v.\mathsf{eq}_3(v_0))))$
$IT^{\circ\circ}_{cv,sd}$		$IT^{\circ\circ}_{cv,sd}(cv, sd) \equiv$
	1.	$\exists v \in cv.\mathsf{vert}()(\mathsf{mod}_2(v.\mathsf{ray}_3(sd)) = 1)$ $\vee$
	2.	$\exists s \in cv.\mathsf{seg}()($
	3.	$\exists p \in sd.\mathsf{pat}()(p.\mathsf{int}_3(s))$ $\vee$
	4.	$\exists s_1, s_2 \in sd.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge s.\mathsf{cop}(s_1) \wedge s.\mathsf{cop}(s_2) \wedge s.\mathsf{cr}_2(s_1) \wedge s.\mathsf{cr}_2(s_2) \wedge$
	5.	$\mathsf{mod}_2(\mathsf{mid}_3(s.\cap_3(s_1), s.\cap_3(s_2)).\mathsf{ray}_3(sd)) = 1)$ $\vee$
	6.	$\exists s_1, s_2 \in sd.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge s.\mathsf{cop}(s_1) \wedge s.\mathsf{cop}(s_2) \wedge s.\mathsf{cr}_2(s_1) \wedge s.\mathsf{tc}_2(s_2) \wedge$
	7.	$\exists v \in s.\mathsf{bnd}()(s_2.\mathsf{cnt}_3(v) \wedge$
	8.	$\mathsf{mod}_2(\mathsf{mid}_3(s.\cap_3(s1), v).\mathsf{ray}_3(sd)) = 1))$ $\vee$
	9.	$\exists s_1, s_2 \in sd.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge s.\mathsf{cop}(s_1) \wedge s.\mathsf{cop}(s_2) \wedge s.\mathsf{tc}_2(s_1) \wedge s.\mathsf{tc}_2(s_2) \wedge$
	10.	$\exists v_1, v_2 \in s.\mathsf{bnd}()(s_1.\mathsf{cnt}_3(v_1) \wedge s_2.\mathsf{cnt}(v_2) \wedge$
	11.	$\mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 1))$ $\vee$
	12.	$\exists s_1 \in sd.\mathsf{seg}()(s.\mathsf{cop}(s_1) \wedge s.\mathsf{cr}_2(s_1) \wedge$
	13.	$\exists v \in sd.\mathsf{vert}()((s.\mathsf{cnt}_3(v) \vee v \in s.\mathsf{bnd}()) \wedge$
	14.	$\mathsf{mod}_2(\mathsf{mid}_3(s.\cap_3(s1), v).\mathsf{ray}_3(sd)) = 1))$ $\vee$
	15.	$\exists s_1 \in sd.\mathsf{seg}()(s.\mathsf{cop}(s_1) \wedge s.\mathsf{tc}_2(s_1) \wedge \exists v_1 \in s.\mathsf{bnd}()(s_1.\mathsf{cnt}_3(v_1) \wedge$
	16.	$\exists v_2 \in sd.\mathsf{vert}()((s.\mathsf{cnt}_3(v_2) \vee v_2 \in s.\mathsf{bnd}()) \wedge$
	17.	$\mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 1))$ $\vee$
	18.	$\exists v_1, v_2 \in sd.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2) \wedge (s.\mathsf{cnt}_3(v_1) \vee v_1 \in s.\mathsf{bnd}()) \wedge$
	19.	$(s.\mathsf{cnt}_3(v_2) \vee v_2 \in s.\mathsf{bnd}()) \wedge \mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 1))$
		where $\mathsf{mod}_2(x)$ returns the rest of the division by 2.

– $IT^{\circ\circ}_{cv,cv}(g_1, g_2)$ is true if at least one segment of g_1 intersects the interior of g_2, excluding the intersections involving the start and end point of g_1 and/or g_2. Sufficient conditions for obtaining this result are produced in the following possible scenarios of Fig. 3: cells $(4, 1)$, $(5, 2)$, $(6, 1)$, $(6, 2)$, $(6, 3)$ and $(6, 4)$. In the proposed test (1st row of Tab. 4) scenarios $(4, 1)$ is covered by formula at line 1, $(5, 2)$ is covered by formulas at line 2 and 3, $(6, 1)$, $(6, 2)$ and $(6, 3)$ are covered by formula at lines 4 and 5. Scenarios $(6, 4)$ produces a sufficient condition for an interior intersection only in the cases that are already detected by cells $(4, 1)$ and $(5, 2)$, thus line 1, 2 and 3 cover also $(6, 4)$.

Table 5 Implementation of tests for Interior/Interior intersection $IT^{\circ\circ}$ for the cases surface/surface, surface/solid and solid/solid. To simplify reading, the OR logical operators connecting alternative conditions have been highlighted with a gray background and consecutive lines have been numbered with the same font color.

Test	#L	Implementation
$IT^{\circ\circ}_{sf,sf}$		$IT^{\circ\circ}_{sf,sf}(sf_1, sf_2) \equiv$
	1.	$\exists p_1 \in sf_1.\mathsf{pat}()(\exists p_2 \in sf_2.\mathsf{pat}()(p_1.\mathsf{int}_3(p_2) \vee$ $(p_1.\mathsf{cop}(p_2) \wedge \neg p_1.\mathsf{dj}_2(p_2) \wedge \neg p_1.\mathsf{tc}_2(p_2)))) \vee$
	2.	$\exists s_1 \in sf_1.\mathsf{intSeg}()(\exists p_2 \in sf_2.\mathsf{pat}()(p_2.\mathsf{cop}(s_1) \wedge \neg p_2.\mathsf{dj}_2(s_1) \wedge \neg p_2.\mathsf{tc}_2(s_1))) \vee$
	3.	$\exists s_2 \in sf_2.\mathsf{intSeg}()(\exists p_1 \in sf_1.\mathsf{pat}()(p_1.\mathsf{cop}(s_2) \wedge \neg p_1.\mathsf{dj}_2(s_2) \wedge \neg p_1.\mathsf{tc}_2(s_2))) \vee$
	4.	$\exists v_1 \in sf_1.\mathsf{intVert}()(\exists p_2 \in sf_2.\mathsf{pat}()(p_2.\mathsf{cnt}_3(v_1))) \vee$
	5.	$\exists v_2 \in sf_2.\mathsf{intVert}()(\exists p_1 \in sf_1.\mathsf{pat}()(p_1.\mathsf{cnt}_3(v_2))) \vee$
	6.	$\exists s_1 \in sf_1.\mathsf{intSeg}()(\exists s_2 \in sf_2.\mathsf{intSeg}()(s_1.\mathsf{eq}_3(s_2) \vee$
	7.	$s_1.\mathsf{int}_3(s_2) \vee (s_1.\mathsf{cop}(s_2) \wedge \neg s_1.\mathsf{dj}_2(s_2) \wedge \neg s_1.\mathsf{tc}_2(s_2)))) \vee$
	8.	$\exists v_1 \in sf_1.\mathsf{intVert}()(\exists s_2 \in sf_2.\mathsf{intSeg}()(s_2.\mathsf{cnt}_3(v_1)) \vee$ $\exists v_2 \in sf_2.\mathsf{intVert}()(v_1.\mathsf{eq}_3(v_2))) \vee$
	9.	$\exists v_2 \in sf_2.\mathsf{intVert}()(\exists s_1 \in sf_1.\mathsf{intSeg}()(s_1.\mathsf{cnt}_3(v_2)))$
$IT^{\circ\circ}_{sf,sd}$		$IT^{\circ\circ}_{sf,sd}(sf, sd) \equiv$
	1.	$\exists v \in sf.\mathsf{vert}()(\mathsf{mod}_2(v.\mathsf{ray}_3(sd)) = 1) \vee$
	2.	$\exists s \in sf.\mathsf{seg}()(IT^{\circ\circ}_{cv,sd}(s, sd)) \vee$
	3.	$\exists p \in sf.\mathsf{pat}()(\forall s \in p.\mathsf{bnd}()(\exists p_0 \in sd.\mathsf{pat}()(p_0.\mathsf{cop}(s) \wedge s.\mathsf{in}_2(p_0)) \vee$ $\exists s_0 \in sd.\mathsf{seg}()(s.\mathsf{eq}_3(s_0) \vee (s.\mathsf{cop}(s_0) \wedge s.\mathsf{in}_2(s_0)))) \wedge$
	4.	$\exists v_1, v_2 \in p.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2) \wedge \mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 1)) \vee$
	5.	$\exists p_1 \in sf.\mathsf{pat}()(\exists p_2 \in sd.\mathsf{pat}()(p_1.\mathsf{int}_3(p_2)))$ where $\mathsf{mod}_2(x)$ returns the rest of the division by 2.
$IT^{\circ\circ}_{sd,sd}$		$IT^{\circ\circ}_{sd,sd}(sd_1, sd_2) \equiv$
	1.	$\exists v_1 \in sd_1.\mathsf{vert}()(\mathsf{mod}_2(v_1.\mathsf{ray}_3(sd_2)) = 1) \vee$
	2.	$\exists v_2 \in sd_2.\mathsf{vert}()(\mathsf{mod}_2(v_2.\mathsf{ray}_3(sd_1)) = 1) \vee$
	3.	$\exists s \in sd_1.\mathsf{seg}()(IT^{\circ\circ}_{cv,sd}(s, sd_2)) \vee \exists s \in sd_2.\mathsf{seg}()(IT^{\circ\circ}_{cv,sd}(s, sd_1)) \vee$
	4.	$\exists p \in sd_1.\mathsf{pat}()(IT^{\circ\circ}_{sf,sd}(p, sd_2)) \vee \exists p \in sd_2.\mathsf{pat}()(IT^{\circ\circ}_{sf,sd}(p, sd_1)) \vee$
	5.	$(\forall v_1 \in sd_1.\mathsf{vert}()(\exists v_2 \in sd_2.\mathsf{vert}()(v_1.\mathsf{eq}_3(v_2))) \wedge$
	6.	$\forall v_2 \in sd_2.\mathsf{vert}()(\exists v_1 \in sd_1.\mathsf{vert}()(v_2.\mathsf{eq}_3(v_1))))$

- $IT^{\circ\circ}_{cv,sf}(g_1, g_2)$ is true if at least one segment of g_1 intersects the interior of g_2; sufficient conditions for obtaining this result are produced in the following possible scenarios of Fig. 4: cells $(1,1)$, $(3,2)$, $(5,2)$, $(6,2)$, $(8,1)$, $(8,2)$, $(8,3)$, $(10,2)$ and $(10,3)$. Cells $(3,4)$, $(5,4)$, $(6,4)$ and $(8,4)$ are already considered by cell $(1,1)$, and cell $(10,4)$ by cell $(5,2)$. In the proposed test (2nd row of Tab. 4) scenarios $(10,2)$ and $(10,3)$ are covered by formula at line 1, $(8,1)$, $(8,2)$ and $(8,3)$ by formula at line 2, $(6,2)$, $(6,4)$ by formula at line 3, $(5,2)$ by formula at line 4, $(3,2)$ and $(1,1)$ by formula at line 5.
- $IT^{\circ\circ}_{cv,sd}(g_1, g_2)$ is true if at least one segment of g_1 intersects the interior of the solid g_2; sufficient conditions for obtaining this result are produced in the following possible scenarios of Fig. 5: cells (10,2), (11,2), (12,2) and (12,3). In the proposed test (3rd row of Tab. 4) scenarios (10,2) and (11,2) are covered by formula at line 1, scenarios described by cell (12,2) that do not imply the containment in the solid volume of a segment vertex (scenarios (10,2) and (11,2)) are covered by formulas at lines 9, 10 and 11,

by formulas at lines 15, 16 and 17 and by formulas at lines 18 and 19 for the case in which the segment starts and ends at solid vertices. Finally, scenarios described by cell (12,3) that do not imply the containment in the solid volume of a segment vertex (scenarios (10,2) and (11,2)) are covered by formulas at lines 4 and 5, by formulas at lines 6, 7 and 8 and by formulas at lines 12, 13 and 14.

– $IT^{\circ\circ}_{sf,sf}(g_1, g_2)$ is true if exists at least one primitive covering the interior of g_1 that intersects one primitive covering the interior of g_2. Sufficient conditions for obtaining this result are produced in the following possible scenarios of Fig. 6-7: cells $(1, 1)$, $(5, 2)$, $(9, 2)$, $(13, 1)$, $(13, 2)$, $(13, 3)$, $(17, 2)$, $(19, 2)$, $(19, 3)$, $(21, 1)$, $(21, 2)$, $(21, 3)$, $(24, 2)$ and $(24, 3)$. Cells $(5, 4)$ and $(9, 4)$ are already considered by cell $(1, 1)$; cells $(13, 4)$ and $(17, 4)$ are already considered by cells $(1, 1)$, $(5, 2)$ and $(9, 2)$; cells $(19, 4)$ and $(21, 4)$ are already considered by cells $(1, 1)$, $(13, 1)$, $(13, 2)$ and $(13, 3)$. In the proposed test (1st row of Tab. 5) scenarios $(1, 1)$ and $(5, 2)$ are covered by formula at line 8, $(9, 2)$ by formula at line 9, $(13, 1)$ by formula at line 6, $(13, 2)$ and $(13, 3)$ by formula at line 7, $(17, 2)$ by formula at line 5, $(19, 2)$ and $(19, 3)$ by formula at line 3, $(21, 1)$, $(21, 2)$ and $(21, 3)$ by formula at line 1, $(22, 2)$ by formula at line 4, $(24, 2)$ and $(24, 3)$ by formula at line 2.

– $IT^{\circ\circ}_{sf,sd}(g_1, g_2)$ is true if at least one patch of g_1 intersects the interior of the solid g_2; sufficient conditions for obtaining this result are produced in the following possible scenarios of Fig. 8-9: cells $(16, 2)$, $(17, 2)$, $(18, 2)$, $(18, 3)$, $(19, 2)$, $(19, 3)$, $(20, 2)$ and $(20, 3)$. In the proposed test (2nd row of Tab. 5) scenarios $(16, 2)$ and $(17, 2)$ are covered by formula at line 1, while scenarios $(18, 2)$, $(18, 3)$, $(19, 2)$ and $(19, 3)$ are covered by formula at line 2. Finally, scenario $(20, 2)$ is covered by formula at lines 3-4 and scenario $(20, 3)$ is covered both by formula at line 1 and formula at line 5.

– $IT^{\circ\circ}_{sd,sd}(g_1, g_2)$ is true if the interior of g_1 intersects at least one primitive defining g_2 or is contained in the volume of g_2, or viceversa. Sufficient conditions for obtaining this result are produced in the following possible scenarios of Tab. 10: $(4, 2)$, $(8, 2)$, $(8, 3)$, $(12, 2)$, $(12, 3)$, $(13, 2)$, $(14, 2)$, $(14, 3)$, $(15, 2)$ and $(15, 3)$. In the proposed test (3rd row of Tab. 5) scenarios $(4, 2)$ and $(13, 2)$ are directly covered by formulas at line 1 and 2, scenarios $(8, 2)$, $(8, 3)$ and $(14, 2)$, $(14, 3)$ are covered by formula at line 3, scenarios $(12, 2)$, $(12, 3)$ and $(15, 2)$, $(15, 3)$ are covered by formula at line 4, finally, formulas at lines 5 and 6 covers the case of exact equality between sd_1 and sd_2. □

Proposition 8 *The* ***Interior/Boundary test*** *($A^\circ \cap \partial B$) has to be specialized for each combination of the geometric types of the involved geometries (A and B). Tab. 6 the tests regarding the following cases are reported: curve/curve, curve/surface, curve/solid, surface/surface, surface/solid and solid/solid. They are expressed by means of the operations and predicates of Def. 3 and Def. 4, respectively.*

Proposition 9 *The* ***Interior/Exterior test*** *($A^\circ \cap B^-$) has to be specialized for each combination of the geometric types of the involved geometries*

Table 6 Implementation of tests for Interior/Boundary intersection $IT^{\circ\partial}$.

Test	#L	Implementation
$IT^{\circ\partial}_{cv,cv}$		$IT^{\circ\partial}_{cv,cv}(cv_1, cv_2) \equiv$
	1.	$\exists v_1 \in cv_1.\mathsf{vert}()(\neg v_1 \in cv_1.\mathsf{bnd}() \wedge \exists v_2 \in cv_2.\mathsf{bnd}()(v_1.\mathsf{eq}(v_2)))\ \vee$
	2.	$\exists s_1 \in cv_1.\mathsf{seg}()(\exists v_2 \in cv_2.\mathsf{bnd}()(s_1.\mathsf{cnt}_3(v_2)))$
$IT^{\circ\partial}_{cv,sf}$		$IT^{\circ\partial}_{cv,sf}(cv, sf) \equiv IT^{\circ\circ}_{cv,cv}(cv, sf.\mathsf{bnd}())$
$IT^{\circ\partial}_{cv,sd}$		$IT^{\circ\partial}_{cv,sd}(cv, sd) \equiv IT^{\circ\circ}_{cv,sd}(cv, sd.\mathsf{bnd}())$
$IT^{\circ\partial}_{sf,sf}$		$IT^{\circ\partial}_{sf,sf}(sf_1, sf_2) \equiv IT^{\circ\circ}_{cv,sf}(sf_2.\mathsf{bnd}(), sf_1)$
$IT^{\circ\partial}_{sf,sd}$		$IT^{\circ\partial}_{sf,sd}(sf, sd) \equiv IT^{\circ\circ}_{sf,sf}(sf, sd.\mathsf{bnd}())$
$IT^{\circ\partial}_{sd,sd}$		$IT^{\circ\partial}_{sd,sd}(sd_1, sd_2) \equiv IT^{\circ\circ}_{sf,sd}(sd_2.\mathsf{bnd}(), sd_1)$

(A and B). In Tab. 7-8 the tests regarding the following cases are reported: curve/curve, curve/surface, curve/solid, surface/surface, surface/solid and solid/solid. They are expressed by means of the operations and predicates of Def. 3 and Def. 4, respectively.

Proposition 10 *The* ***Boundary/Boundary test*** *($\partial A \cap \partial B$) has to be specialized for each combination of the geometric types of the involved geometries (A and B). In Tab. 9 the tests regarding the following cases are reported: curve/curve, curve/surface, curve/solid, surface/surface, surface/solid and solid/solid. They are expressed by means of the operations and predicates of Def. 3 and Def. 4, respectively.*

Proposition 11 *The* ***Boundary/Exterior test*** *($\partial A \cap B^-$) has to be specialized for each combination of the geometric types of the involved geometries (A and B). In Tab. 10 the tests regarding the following cases are reported: curve/curve, curve/surface, curve/solid, surface/surface, surface/solid and solid/solid. They are expressed by means of the operations and predicates of Def. 3 and Def. 4, respectively.*

Given the previous propositions regarding the cells of the 9IM, the following final proposition shows how to combine them in order to obtain the necessary test for a given topological relation defined by a matrix pattern.

Proposition 12 *Given the intersection tests shown in Prop. 7, 8, 9, 10 and 11, the implementation of the test for a specific topological relation $R(A, B)$ can be obtained by considering: (i) the types $type(A)$ and $type(B)$ of the geometries A and B; (ii) the matrix pattern of the 9IM that corresponds to the relation R when evaluated on geometries with types $type(A)$ and $type(B)$ (see Tab. 1 of Sect. 3). Once the matrix pattern $[I_1I_2I_3 - B_1B_2B_3 - E_1E_2E_3]$ has been identified, it can be possibly simplified considering that geometries are embedded in the 3D space, and then translated into a conjunction of intersection tests*

Table 7 Implementation of tests for Interior/Exterior intersection $IT^{\circ-}$ for the cases curve/curve, curve/surface and curve/solid.

Test	#L	Implementation
$IT^{\circ-}_{cv,cv}$		$IT^{\circ-}_{cv,cv}(cv_1, cv_2) \equiv$
	1.	$\exists v_1 \in cv_1.\mathsf{vert}()(\neg v_1 \in cv_1.\mathsf{bnd}() \wedge \forall v_2 \in cv_2.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2)) \wedge$
	2.	$\forall s \in cv_2.\mathsf{seg}()(\neg s.\mathsf{cnt}_3(v_1))) \vee$
	3.	$\exists s_1 \in cv_1.\mathsf{seg}()(\forall s_2 \in cv_2.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge \neg(s_1.\mathsf{cop}_3(s_2) \wedge s_1.\mathsf{in}_2(s_2)))$
$IT^{\circ-}_{cv,sf}$		$IT^{\circ-}_{cv,sf}(cv, sf) \equiv$
	1.	$\exists v_1 \in cv.\mathsf{vert}()(\neg v_1 \in cv.\mathsf{bnd}() \wedge \forall v_2 \in sf.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2)) \wedge$
	2.	$\forall s \in sf.\mathsf{seg}()(\neg s.\mathsf{cnt}_3(v_1)) \wedge$
	3.	$\forall p \in sf.\mathsf{pat}()(\neg p.\mathsf{cnt}_3(v_1))) \vee$
	4.	$\exists s_1 \in cv.\mathsf{seg}()(\forall s_2 \in sf.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge \neg(s_1.\mathsf{cop}_3(s_2) \wedge s_1.\mathsf{in}_2(s_2))) \wedge$
	5.	$\forall p \in sf.\mathsf{pat}()(\neg(p.\mathsf{cop}_3(s_1) \wedge s_1.\mathsf{in}_2(p))))$
$IT^{\circ-}_{cv,sd}$		$IT^{\circ-}_{cv,sd}(cv, sd) \equiv$
	1.	$\exists v \in cv.\mathsf{vert}()(\mathsf{mod}_2(v.\mathsf{ray}_3(sd)) = 0) \vee$
	2.	$\exists s \in cv.\mathsf{seg}()($
	3.	$\exists p \in sd.\mathsf{pat}()(p.\mathsf{int}_3(s)) \vee$
	4.	$\exists s_1, s_2 \in sd.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge s.\mathsf{cop}(s_1) \wedge s.\mathsf{cop}(s_2) \wedge s.\mathsf{cr}_2(s_1) \wedge s.\mathsf{cr}_2(s_2) \wedge$
	5.	$\mathsf{mod}_2(\mathsf{mid}_3(s.\cap_3(s_1), s.\cap_3(s_2)).\mathsf{ray}_3(sd)) = 0) \vee$
	6.	$\exists s_1, s_2 \in sd.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge s.\mathsf{cop}(s_1) \wedge s.\mathsf{cop}(s_2) \wedge s.\mathsf{cr}_2(s_1) \wedge s.\mathsf{tc}_2(s_2) \wedge$
	7.	$\exists v \in s.\mathsf{bnd}()(s_2.\mathsf{cnt}_3(v) \wedge$
	8.	$\mathsf{mod}_2(\mathsf{mid}_3(s.\cap_3(s1), v).\mathsf{ray}_3(sd)) = 0)) \vee$
	9.	$\exists s_1, s_2 \in sd.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge s.\mathsf{cop}(s_1) \wedge s.\mathsf{cop}(s_2) \wedge s.\mathsf{tc}_2(s_1) \wedge s.\mathsf{tc}_2(s_2) \wedge$
	10.	$\exists v_1, v_2 \in s.\mathsf{bnd}()(s_1.\mathsf{cnt}_3(v_1) \wedge s_2.\mathsf{cnt}(v_2) \wedge$
	11.	$\mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 0)) \vee$
	12.	$\exists s_1 \in sd.\mathsf{seg}()(s.\mathsf{cop}(s_1) \wedge s.\mathsf{cr}_2(s_1) \wedge$
	13.	$\exists v \in sd.\mathsf{vert}()((s.\mathsf{cnt}_3(v) \vee v \in s.\mathsf{bnd}()) \wedge$
	14.	$\mathsf{mod}_2(\mathsf{mid}_3(s.\cap_3(s1), v).\mathsf{ray}_3(sd)) = 0)) \vee$
	15.	$\exists s_1 \in sd.\mathsf{seg}()(s.\mathsf{cop}(s_1) \wedge s.\mathsf{tc}_2(s_1) \wedge \exists v_1 \in s.\mathsf{bnd}()(s_1.\mathsf{cnt}_3(v_1) \wedge$
	16.	$\exists v_2 \in sd.\mathsf{vert}()((s.\mathsf{cnt}_3(v_2) \vee v_2 \in s.\mathsf{bnd}()) \wedge$
	17.	$\mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 0)) \vee$
	18.	$\exists v_1, v_2 \in sd.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2) \wedge (s.\mathsf{cnt}_3(v_1) \vee v_1 \in s.\mathsf{bnd}()) \wedge$
	19.	$(s.\mathsf{cnt}_3(v_2) \vee v_2 \in s.\mathsf{bnd}()) \wedge \mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 0)))$

(IT) according to the following rule:

$$TT([I_1 I_2 I_3 - B_1 B_2 B_3 - E_1 E_2 E_3], gt_1, gt_2) = \bigwedge_{i=1}^{3} \tau_{I,i}(I_i, gt_1, gt_2) \wedge \bigwedge_{i=2}^{3} \tau_{B,i}(B_i, gt_1, gt_2)$$

where:

$$\tau_{I,i}(b, gt_1, gt_2) = \begin{cases} \mathit{true} & \mathit{if}\ i = * \\ (b \implies IT^{\circ\circ}_{gt_1,gt_2}()) \wedge (\neg b \implies \neg IT^{\circ\circ}_{gt_1,gt_2}()) & \mathit{if}\ i = 1 \\ (b \implies IT^{\circ\partial}_{gt_1,gt_2}()) \wedge (\neg b \implies \neg IT^{\circ\partial}_{gt_1,gt_2}()) & \mathit{if}\ i = 2 \\ (b \implies IT^{\circ-}_{gt_1,gt_2}()) \wedge (\neg b \implies \neg IT^{\circ-}_{gt_1,gt_2}()) & \mathit{if}\ i = 3 \end{cases}$$

Table 8 Implementation of tests for Interior/Exterior intersection $IT^{\circ-}$ for the cases surface/surface, surface/solid and solid/solid.

Test	#L	Implementation
$IT^{\circ-}_{sf,sf}$		$IT^{\circ-}_{sf,sf}(sf_1, sf_2) \equiv$
	1.	$\exists v_1 \in sf_1.\mathsf{vert}()(\forall v_2 \in sf_2.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2)) \wedge$
	2.	$\forall s \in sf_2.\mathsf{seg}()(\neg s.\mathsf{cnt}_3(v_1)) \wedge$
	3.	$\forall p \in sf_2.\mathsf{pat}()(\neg p.\mathsf{cnt}_3(v_1))) \vee$
	4.	$\exists s_1 \in sf_1.\mathsf{seg}()(\forall s_2 \in sf_2.\mathsf{seg}()(\neg s_1.\mathsf{eq}_3(s_2) \wedge \neg(s_1.\mathsf{cop}_3(s_2) \wedge s_1.\mathsf{in}_2(s_2)))\wedge$
	5.	$\forall p \in sf_2.\mathsf{pat}()(\neg(p.\mathsf{cop}_3(s_1) \wedge s_1.\mathsf{in}_2(p)))) \vee$
	6.	$\exists p_1 \in sf_1.\mathsf{pat}()(\forall p_2 \in sf_2.\mathsf{pat}()(\neg(p_1.\mathsf{cop}(p_2) \wedge (p_1.\mathsf{eq}_2(p_2) \vee p_1.\mathsf{in}_2(p_2)))))$
$IT^{\circ-}_{sf,sd}$		$IT^{\circ-}_{sf,sd}(sf, sd) \equiv$
	1.	$\exists v_1 \in sf.\mathsf{vert}()(\forall v_2 \in sd.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2)) \wedge$
	2.	$\forall s \in sd.\mathsf{seg}()(\neg s.\mathsf{cnt}_3(v_1)) \wedge$
	3.	$\forall p \in sd.\mathsf{pat}()(\neg p.\mathsf{cnt}_3(v_1)) \wedge$
	4.	$\mathsf{mod}_2(v_1.\mathsf{ray}_3(sd)) = 0) \vee$
	5.	$\exists s \in sf.\mathsf{seg}()(INT^{\circ-}_{cv,sd}(s, sd)) \vee$
	6.	$\exists p \in sf.\mathsf{pat}()(\forall s \in p.\mathsf{bnd}()(\exists p_0 \in sd.\mathsf{pat}()(p_0.\mathsf{cop}(s) \wedge s.\mathsf{in}_2(p_0)) \vee$
		$\exists s_0 \in sd.\mathsf{seg}()(s.\mathsf{eq}_3(s_0) \vee (s.\mathsf{cop}(s_0) \wedge s.\mathsf{in}_2(s_0)))) \wedge$
	7.	$\exists v_1, v_2 \in p.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2) \wedge$
		$\mathsf{mod}_2(\mathsf{mid}_3(v_1, v_2).\mathsf{ray}_3(sd)) = 0)) \vee$
	8.	$\exists p_1 \in sf.\mathsf{pat}()(\exists p_2 \in sd.\mathsf{pat}()(p_1.\mathsf{int}_3(p_2)))$
$IT^{\circ-}_{sd,sd}$		$IT^{\circ-}_{sd,sd}(sd_1, sd_2) \equiv$
	1.	$\exists v_1 \in sd_1.\mathsf{vert}()(\mathsf{mod}_2(v_1.\mathsf{ray}_3(sd_2)) = 0) \vee$
	2.	$\exists v_2 \in sd_2.\mathsf{vert}()(\mathsf{mod}_2(v_2.\mathsf{ray}_3(sd_1)) = 0) \vee$
	3.	$\exists s_1 \in sd_1.\mathsf{seg}()(IT^{\circ-}_{cv,sd}(s_1, sd_2)) \vee \exists s_2 \in sd_2.\mathsf{seg}()(IT^{\circ-}_{cv,sd}(s_2, sd_1)) \vee$
	4.	$\exists p_1 \in sd_1.\mathsf{pat}()(IT^{\circ-}_{sf,sd}(p_1, sd_2)) \vee \exists p_2 \in sd_2.\mathsf{pat}()(IT^{\circ-}_{sf,sd}(p_2, sd_1))$

Table 9 Implementation of tests for Boundary/Boundary intersection $IT^{\,\partial\partial}$.

Test	#L	Implementation
$IT^{\,\partial\partial}_{cv,cv}$		$IT^{\,\partial\partial}_{cv,cv}(cv_1, cv_2) \equiv$
	1.	$\exists v_1 \in cv_1.\mathsf{bnd}()(\exists v_2 \in cv_2.\mathsf{bnd}()(v_1.\mathsf{eq}_3(v_2)))$
$IT^{\,\partial\partial}_{cv,sf}$		$IT^{\,\partial\partial}_{cv,sf}(cv, sf) \equiv$
	1.	$\exists v \in cv.\mathsf{bnd}()(\exists r \in sf.\mathsf{bnd}()(\exists s \in r.\mathsf{seg}()(s.\mathsf{cnt}_3(v) \vee$
	2.	$\exists v_s \in s.\mathsf{bnd}()(v_s.\mathsf{eq}_3(v)))))$
$IT^{\partial\partial}_{cv,sd}$		$IT^{\partial\partial}_{cv,sd}(cv, sd) \equiv$
	1.	$\exists v \in cv.\mathsf{bnd}()(\exists p \in sd.\mathsf{pat}()(p.\mathsf{cnt}_3(v) \vee$
	2.	$\exists s \in p.\mathsf{seg}()(s.\mathsf{cnt}_3(v)) \vee$
	3.	$\exists v_p \in p.\mathsf{vert}()(v_p.\mathsf{eq}_3(v))))$
$IT^{\partial\partial}_{sf,sf}$		$IT^{\partial\partial}_{sf,sf}(sf_1, sf_2) \equiv IT^{\circ\circ}_{cv,cv}(sf_1.\mathsf{bnd}(), sf_2.\mathsf{bnd}())$
$IT^{\partial\partial}_{sf,sd}$		$IT^{\partial\partial}_{sf,sd}(sf, sd) \equiv IT^{\circ\circ}_{cv,sd}(sf.\mathsf{bnd}(), sd.\mathsf{bnd}())$
$IT^{\partial\partial}_{sd,sd}$		$IT^{\partial\partial}_{sd,sd}(sd_1, sd_2) \equiv IT^{\circ\circ}_{sf,sf}(sd_1.\mathsf{bnd}(), sd_2.\mathsf{bnd}())$

Table 10 Implementation of tests for Boundary/Exterior intersection $IT^{\partial -}$.

Test	**#L**	**Implementation**
$IT^{\partial -}_{cv,cv}$		$IT^{\partial -}_{cv,cv}(cv_1, cv_2) \equiv$
	1.	$\exists v_1 \in cv_1.\mathsf{bnd}()(\forall v_2 \in cv_2.\mathsf{vert}()(\neg v_1.\mathsf{eq}_3(v_2))\wedge$
	2.	$\forall s_2 \in cv_2.\mathsf{seg}()(\neg s_2.\mathsf{cnt}_3(v_1)))$
$IT^{\partial -}_{cv,sf}$		$IT^{\partial -}_{cv,sf}(cv, sf) \equiv$
	1.	$\exists v \in cv.\mathsf{bnd}()(\forall v_s \in sf.\mathsf{vert}()(\neg v.\mathsf{eq}_3(v_s))\wedge$
	2.	$\forall s \in sf.\mathsf{seg}()(\neg s.\mathsf{cnt}_3(v))\wedge$
	3.	$\forall p \in sf.\mathsf{pat}()(\neg p.\mathsf{cnt}_3(v)))$
$IT^{\partial -}_{cv,sd}$		$IT^{\partial -}_{cv,sd}(cv, sd) \equiv$
	1.	$\exists v \in cv.\mathsf{bnd}()(\forall v_s \in sd.\mathsf{vert}()(\neg v_s.\mathsf{eq}_3(v))\wedge$
	2.	$\forall s \in sd.\mathsf{seg}()(\neg s.\mathsf{cnt}_3(v))\wedge$
	3.	$\forall p \in sd.\mathsf{pat}()(\neg p.\mathsf{cnt}_3(v))\wedge$
	4.	$\mathsf{mod}_2(v.\mathsf{ray}_3(sd)) = 0)$
$IT^{\partial -}_{sf,sf}$		$IT^{\partial -}_{sf,sf}(sf_1, sf_2) \equiv IT^{\circ -}_{cv,sf}(sf_1.\mathsf{bnd}(), sf_2)$
$IT^{\partial -}_{sf,sd}$		$IT^{\partial -}_{sf,sd}(sf, sd) \equiv IT^{\circ -}_{cv,sd}(sf.\mathsf{bnd}(), sd)$
$IT^{\partial -}_{sd,sd}$		$IT^{\partial -}_{sd,sd}(sd_1, sd_2) \equiv IT^{\circ -}_{sf,sd}(sd_1.\mathsf{bnd}(), sd_2)$

$$\tau_{B,i}(b, gt_1, gt_2) = \begin{cases} true & \text{if } b = * \\ (b \implies IT^{\partial\circ}_{gt_1,gt_2}()) \wedge (\neg b \implies \neg IT^{\partial\circ}_{gt_1,gt_2}()) & \text{if } i = 1 \\ (b \implies IT^{\partial\partial}_{gt_1,gt_2}()) \wedge (\neg b \implies \neg IT^{\partial\partial}_{gt_1,gt_2}()) & \text{if } i = 2 \\ (b \implies IT^{\partial -}_{gt_1,gt_2}()) \wedge (\neg b \implies \neg IT^{\partial -}_{gt_1,gt_2}()) & \text{if } i = 3 \end{cases}$$

Proof The correctness of the test obtained by applying the proposed rule derives straightforward from the correctness of the intersection tests (Prop. 7, 8, 9, 10 and 11) and from the following observation. The translation functions $\tau_{I,i}$ and $\tau_{B,i}$ when applied to the corresponding cell of the matrix pattern return: (i) true, when the cell is equal to $*$; (ii) the intersection test that corresponds to the position of the cell in the matrix (for example $IT^{\circ\circ}_{gt_1,gt_2}()$), when the cell is equal to T (true); (ii) the negation of the same intersection test (for example $IT^{\circ\circ}_{gt_1,gt_2}()$), when the cell is equal to F (false). Then the conjunction of these tests are considered, thus, it correctly tests the considered topological relation according to the definition of Sec. 3. □

7 Conclusion

The evaluation of 3D topological relations between geometries of different types is still a challenging task in current GIS systems. Systems like PostGIS usually supports the definition of 3D data types, such as line-strings, polyhedral surfaces and solids as closed polyhedral surfaces, but they currently provide a very limited set of operations on them. In particular, the checking of 3D topological relations traditionally requires the definition of ad-hoc implementations. This paper proposes a different approach for overcoming such problem, in particular instead of producing ad-hoc procedure from scratch whenever needed, it tries to provide a generic solution by exploiting the operation already implemented in the currently available GIS system.

More specifically, the contribution provided by this paper is twofold: (1) the definition of an abstract 3D vector model on which geometric types and topological relations are defined, (2) the definition of a set of procedures for testing 3D topological relations based on the abstract 3D vector model.

As regards to the first point, starting from the geometric model proposed in the ISO Standard 19136 GML, this paper introduces a vector model that provides a general environment representing an abstraction of the current technology. This model is composed of a set of basic 3D vector types together with predicates and operations, that are necessary in order to implement the topological relation tests. Such 3D topological relations have been defined by adopting the well-known 9-intersection model (9IM) [5]. Subsequently, as regards to the second point, for each 3D topological relation and each combination of geometric types, the necessary test is given by means of a logic formula that represents the condition to be satisfied by the geometric primitives in order to satisfy the relation.

More specifically, a regards to the second point: we start by identifying all the possible relations that can exists between the components of two vector representations. This is done by a set of propositions that given the vector representation of two geometric types, identify all possible scenarios to be considered to evaluate the existence of a topological relation between them. Given the identification of such situations, we are able to provide a set of tests for checking topological relations. In particular, we start by considering the structure of the 9IM, then we show how each significant cell of the matrix can be tested, by considering the possible situations previously identified. The test implementation of each matrix cell can be considered a basic test, since it is a building block of our procedure, and is defined as a logical formula. Given such basic tests, the test of particular topological relation can be obtained by specifying a conjunction of such basic tests. Proofs of both basic and complete tests have been provided. A prototypical implementation of such test in PostGIS is available at [1].

The main benefit of the proposed approach is that it provides a general solution easily achievable in exiting GIS systems, because it relies on functions and 2D topological relations commonly available. Moreover, it is particularly suitable for integration with approaches like the one in [2], since the proposed tests can be used for automatically validate topological integrity constraints specified by means of templates.

Future work will regard: (i) the implementation of validator tools for datasets describing urban contexts, which are stored in spatial DBMS; (ii) the testing of the proposed approach on huge datasets using a map-reduce approach, as done in [16] for 2D spatial constraint templates.

References

1. Belussi, A., Migliorini, S., Negri, M.: (2019). https://github.com/smigliorini/spatialdbgroup Last accessed Dec. 2019
2. Belussi, A., Migliorini, S., Negri, M., Pelagatti, G.: A template-based approach for the specification of 3D topological constraints. GeoInformatica **under review**
3. Belussi, A., Migliorini, S., Negri, M., Pelagatti, G.: Validation of Spatial Integrity Constraints in City Models. In: Proc. of the 4th ACM SIGSPATIAL Int. Workshop on Mobile Geographic Information Systems, MobiGIS '15, pp. 70–79 (2015)
4. Bieri, H.: Nef polyhedra: A brief introduction. Computing Supplementum **10**, 43–60 (1995)
5. Egenhofer, M.J.: Topological relations in 3D. Tech. rep., University of Maine, USA (1995)
6. Egenhofer, M.J., Franzosa, R.: Point-set topological spatial relations. International Journal of Geographic Information Systems **2**(5), 161–174 (1991)
7. Hachenberger, P., Kettner, L., Mehlhorn, K.: Boolean operations on 3D selective Nef complexes: Data structure, algorithms, optimized implementation and experiments. Computational Geometry **38**(1–2), 64–99 (2007)
8. Open Geospatial Consortium Inc.: OpenGIS Implementation Standard for Geographic Information - Simple feature access - Part 1: Common architecture (2011). Version 1.2.1 http://www.opengeospatial.org/standards/sfa
9. Open Geospatial Consortium Inc.: OGC City Geography Markup Language (CityGML) Encoding Standard (2012). Version 2.0 http://www.opengeospatial.org/standards/citygml
10. ISO: ISO 19107 Geographic Information – Spatial Schema (2003). http://www.iso.org/iso/catalogue_detail.htm?csnumber=26012
11. ISO: ISO 19125 Geographic Information – Simple Feature Access – Part 1: Common architecture (2004). https://www.iso.org/standard/40114.html
12. ISO: Geographic information – Geography Markup Language (GML) (ISO 19136:2007 (2007). http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32554
13. Khuan, C., Abdul-Rahman, A., Zlatanova, S.: 3D Solids and Their Management in DBMS. In: Advances in 3D Geoinformation Systems, pp. 279–311. Springer Berlin Heidelberg (2008)
14. Ledoux, H.: On the validation of solids represented with the international standards for geographic information. Computer-Aided Civil and Infrastructure Engineering **28**(9), 693–706 (2013)
15. Migliorini, S.: Enhancing cidoc-crm models for geosparql processing with mapreduce. In: Proceedings of 2nd Workshop On Computing Techniques For Spatio-Temporal Data in Archaeology And Cultural Heritage, pp. 45–59 (2018). URL http://coarch18.di.univr.it/wp-content/uploads/2018/02/paper_03.pdf
16. Migliorini, S., Belussi, A., Negri, M., Pelagatti, G.: Towards Massive Spatial Data Validation with SpatialHadoop. In: Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial'16, pp. 18–27 (2016)
17. OSGeo: PostGIS 3.0 Manual. Open Source Geospatial Foundation (2019). https://postgis.net/stuff/postgis-3.0.pdf
18. Pelagatti, G., Negri, M., Belussi, A., Migliorini, S.: From the Conceptual Design of Spatial Constraints to Their Implementation in Real Systems. In: 17th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems, pp. 448–451 (2009)
19. Tet-Khuan, C., Abdul-Rahman, A., Zlatanova, S.: 3D Spatial Operations in Geo DBMS Environment for 3D GIS. In: O. Gervasi, M.L. Gavrilova (eds.) Computational Science and Its Applications – ICCSA 2007, pp. 151–163 (2007)
20. Wagner, D., Wewetzer, M., Bogdahn, J., Alam, N., Pries, M., Coors, V.: Geometric-Semantical Consistency Validation of CityGML Models. In: Progress and New Trends in 3D Geoinformation Sciences, pp. 171–192 (2013)
21. Woo, T.C., Thomasma, T.: An algorithm for generating solid elements in objects with holes. Computers & Structures **18**(2), 333–342 (1984)

22. Xu, D.: Design and Implementation of Constraints for 3D Spatial Database: Using Climate City Campus Database as an Example. Master's thesis, OTB Research Institute for the Built Environment (2011)
23. Xu, D., van Oosterom, P., Zlatanova, S.: A methodology for modelling of 3D spatial constraints. In: Advances in 3D Geoinformation, pp. 95–117 (2016)
24. Xu, D., Zlatanova, S.: An approach to develop 3D geo-DBMS topological operators by re-using existing 2D operators. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences **II-2/W1**, 291–298 (2013)
25. Zlatanova, S.: On 3D topological relationships. In: Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on, pp. 913–919 (2000)
26. Zlatanova, S., Abdul-Rahman, A., Shi, W.: Topological models and frameworks for 3D spatial objects. Computers & Geosciences **30**(4), 419–428 (2004)

A Proofs of the Test Implementations

This section reports the detailed proofs for Prop. 8-11 in Sect. 6.

Proof for Proposition 8 ($IT^{\circ\partial}$)

The proof shows that each specialization of the intersection test ($IT^{\circ\partial}$) reported in Tab. 6 covers all the possible scenarios between A and B. The complete enumeration of all scenarios is obtained as in Prop. 7 in Sect. 6.

- $IT^{\circ\partial}_{cv,cv}(g_1, g_2)$ is true if at least one vertex of the boundary of g_2 intersects g_1; sufficient conditions for obtaining this result are produced in the following scenarios of Tab. 3: cells $(1, 1)$, $(2, 1)$ and $(3, 2)$. Scenario $(3, 4)$ is already considered by cell $(1, 1)$. In the proposed test (1st row of Tab. 6) scenarios $(1, 1)$ and $(2, 1)$ are covered by formula at line 1, while scenario $(3, 2)$ is covered by formula at line 2.
- all other tests can be converted in a test $IT^{\circ\circ}_{*,*}()$ as shown in Tab. 6. This can also be done because the boundary of a surface or a solid is always a cycle, i.e. it has an empty boundary. □

Proof for Proposition 9 ($IT^{\circ-}$)

The proof shows that each specialization of the intersection test $IT^{\circ-}$ covers all the possible scenarios between A and B. The complete enumeration of all scenarios is obtained as explained in Prop. 7 in Sect. 6.

- $IT^{\circ-}_{cv,cv}(g_1, g_2)$ is true if at least one internal vertex or segment of g_1 intersects the exterior of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 3: (i) when it exists at least one internal vertex of g_1 for which the scenarios of cells $(4, 1)$ and $(5, 2)$ do not occur; (ii) when it exists at least a segment of g_1 such that for all segments of g_2 scenarios of cells $(6, 1)$ and $(6, 2)(IN)$ do not occur.
  In the proposed test (1st row of Tab. 7) the first case involving scenarios $(4, 1)$ and $(5, 2)$ is covered by formula at lines 1-2, while the second one involving scenarios $(6, 1)$ and $(6, 2)$ is covered by formula at line 3.
- $IT^{\circ-}_{cv,sf}(g_1, g_2)$ is true if at least one internal vertex or segment of g_1 intersects the exterior of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 4: (i) when it exists at least one internal vertex of g_1 for which all the scenarios of cells $(1, 1)$, $(2, 1)$, $(3, 2)$, $(3, 4)$, $(4, 2)$, $(4, 4)$, $(5, 2)$ and $(5, 4)$ do not occur; (ii) when it exists at least a segment of g_1 such that for all segments and patches of g_2 scenarios of cells $(8, 1)$, $(8, 2)$, $(9, 1)$, $(9, 2)$ and $(10, 2)$ do not occur.
  In the proposed test (2nd row of Tab. 7) the first case regarding cells $(1, 1)$, $(2, 1)$, $(3, 4)$, $(4, 4)$ and $(5, 4)$, is covered by the formula at line 1, regarding cells $(3, 2)$ and $(4, 2)$, by the formula at line 2 and regarding cell $(5, 2)$ by formula at line 3; the second case regarding cells $(8, 1)$, $(8, 2)$, $(9, 1)$ and $(9, 2)$, is covered by the formula at line 4 and, regarding cell $(10, 2)$ by formula at line 5.

- $IT^{\circ -}_{cv,sd}(g_1, g_2)$ is true if at least one vertex or segment of g_1 intersects the exterior of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 5: (i) when it exists a vertex of g_1 such that no one of the scenario of cells $(1,1)$, $(2,2)$, $(2,4)$, $(3,2)$, $(3,4)$, $(4,1)$, $(5,2)$, $(5,4)$, $(6,2)$, $(6,4)$, $(10,2)$ and $(11,2)$ occurs; (ii) when it exists a segment of g_1 such that the scenario of cell $(9,3)$ occurs provided that the intersection between the segment and the patch is a point; (iii) when it exists a segment of g_1 such that no one of the scenarios of cells $(8,1)$, $(8,2)$(IN), $(9,2)$, $(12,2)$ and $(9,3)$ together with $(12,3)$ (the segment is contained in the union of a patch and the interior of the solid) occurs.
  In the proposed test (third row of Tab. 7) the first case, regarding all cells, is covered by the formula at line 1, the second case is covered by the formula at line 2 combined with line 3, finally the third case is covered by the formula at line 2 combined with lines 4-19.
- $IT^{\circ -}_{sf,sf}(g_1, g_2)$ is true if at least one vertex, segment or patch of g_1 intersects the exterior of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 6-7: (i) when it exists a vertex of g_1 such that no one of the scenario of rows $1-8$, 22 and 23 occurs; (ii) when it exists a segment of g_1 such that no one of the scenarios of cells $(13,1)$, $(13,2)$(IN), $(14,1)$, $(14,2)$(IN),$(15,1)$, $(15,2)$(IN), $(16,1)$, $(16,2)$(IN), $(24,2)$ and $(25,2)$ occurs; (iii) when it exists a patch of g_1 such that no one of the scenarios of cells $(21,1)$ and $(21,2)$(IN) occurs.
  In the proposed test (1st row of Tab. 8) the first case, regarding cells of rows $1-4$, is covered by the formula at line 1, regarding cells of rows $5-8$, is covered by the formula at line 2, regarding cells of rows 22 and 23, is covered by the formula at line 3; the second case, regarding cells $(13,1)$, $(13,2)$(IN), $(14,1)$, $(14,2)$(IN),$(15,1)$, $(15,2)$(IN), $(16,1)$, $(16,2)$(IN), is covered by the formula at line 4, regarding cells $(24,2)$ and $(25,2)$, is covered by formula at line 5; finally the third case is covered by formula at line 6.
- $IT^{\circ -}_{sf,sd}(g_1, g_2)$ is true if at least one vertex or segment of g_1 intersects the exterior of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 8-9: (i) when it exists a vertex of g_1 such that no one of the scenario of rows $1-6$, 16 and 17 occurs; (ii) when it exists a segment of g_1 such that no one of the scenarios of cells $(8,1)$, $(8,2)$(IN), $(9,2)$, $(11,1)$, $(11,2)$(IN),$(12,2)$, $(18,2)$ and $(19,2)$ occurs; (iii) when it exists a patch of g_1 such that no one of the scenarios of cells $(15,1)$, $(15,2)$(IN) and $(20,2)$ occurs.
  In the proposed test (2nd row of Tab. 8) the first case, regarding cells $(1,1)$, $(2,4)$ and $(4,1)$ $(5,4)$, is covered by the formula at line 1, regarding cells $(2,2)$, $(3,4)$ and $(5,2)$, $(6,4)$, is covered by the formula at line 2, regarding cells $(3,2)$ and $(6,2)$, is covered by the formula at line 3, regarding cells $(16,2)$ and $(17,2)$, is covered by the formula at line 4; the second case is covered by formula at line 5; finally, the third case, regarding cell $(15,1)$, is covered by formula at line 8 (and partially by formula at line 5), regarding cell $(15,2)$, is covered by formula at line 8 and, regarding cell $(20,2)$, is covered by formula at line 7 and 8.

- $IT^{\circ-}_{sd,sd}(g_1, g_2)$ is true if at least one vertex or segment or patch of g_1 intersects the exterior of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 10: (i) when it exists a vertex of g_1 such that no one of the scenario of rows $1-4$ occurs; (ii) when it exists a segment of g_1 such that no one of the scenarios of cells $(6,1)$, $(6,2)$(IN), $(7,2)$ and $(8,2)$ occurs; (iii) when it exists a patch of g_1 such that no one of the scenarios of cells $(11,1)$, $(11,2)$(IN) and $(12,2)$ occurs. Since sd_2 cannot have holes then, there are no other cases to consider, indeed, in order for s_1 to intersect the exterior of s_2, it is necessary that at least a patch of s_1 intersects the exterior of s_2.
  In the proposed test (3rd row of Tab. 8) the first case is covered by formula at line 1 and 2; the second case is covered by formula at line 3 and the third on by formula at line 4.

Proof for Proposition 9 ($IT^{\partial\partial}$)

In the proof we show that each specialization of the intersection test ($IT^{\partial\partial}$) covers all the possible scenarios between A and B. The complete enumeration of all scenarios is obtained as explained in Prop. 7 in Sect. 6.

- $IT^{\partial\partial}_{cv,cv}(g_1, g_2)$ is true if at least one vertex of the boundary of g_1 is equal to a vertex of the boundary of g_2; sufficient conditions for obtaining this result are produced in the following case of Tab. 3: cell $(1,1)$. In the proposed test (1st row of Tab. 9) this case is covered by the formula at line 1.
- $IT^{\partial\partial}_{cv,sf}(g_1, g_2)$ is true if at least one vertex of the boundary of g_1 intersects the curve representing the boundary of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 4: cells $(2bis,1)$, $(4bis,2)$ and $(4bis,4)$. In the proposed test (2nd row of Tab. 9) scenario $(4bis,2)$ is covered by the formula at line 1, while scenarios $(2bis,1)$ and $(4bis,4)$ are covered by the formula at line 2.
- $IT^{\partial\partial}_{cv,sd}(g_1, g_2)$ is true if at least one vertex of the boundary of g_1 intersects the surface representing the boundary of g_2; sufficient conditions for obtaining this result are produced in the following cases of Tab. 5: cells $(1,1)$, $(2,2)$, $(2,4)$, $(3,2)$ and $(3,4)$. In the proposed test (third row of Tab. 9) scenarios $(1,1)$, $(2,4)$ and $(3,4)$(intersection on the patch vertices) are covered by the formula at line 3; scenarios $(2,2)$ and $(3,4)$(intersection on the patch segments) are covered by formula at line 2; finally scenario $(3,2)$ is covered by formula at line 1.
- All other tests can be converted in a test $IT^{\circ\circ}_{*,*}()$ as shown in Tab. 9. This can be done also because the boundary of a surface or a solid is always a cycle, i.e. it has an empty boundary.

Proof for Proposition 9 ($IT^{\partial-}$)

In the proof we show that each specialization of the intersection test ($IT^{\partial-}$) covers all the possible scenarios between A and B. The complete enumeration of all scenarios is obtained as explained in Prop. 7 in Sect. 6.

- $IT^{\partial -}_{cv,cv}(g_1, g_2)$ is true if at least one vertex of the boundary of g_1 intersects the exterior of g_2; the sufficient condition for obtaining this result is produced when there exists at least one vertex of the boundary of g_1 for which the scenarios of cells $(1,1)$, $(2,1)$, $(3,2)$ and $(3,4)$ of Tab. 3 do not occur. In the proposed test (1st row of Tab. 10) scenarios $(1,1)$, $(2,1)$ and $(3,4)$ are covered by the formula at line 1, while scenario $(3,2)$ is covered by formula at line 2.
- $IT^{\partial -}_{cv,sf}(g_1, g_2)$ is true if at least one vertex of the boundary of g_1 intersects the exterior of g_2; the sufficient condition for obtaining this result is produced when there exists at least one vertex of the boundary of g_1 for which the scenarios of cells: $(1bis,1)$, $(2bis,1)$, $(3bis,2)$, $(3bis,4)$, $(4bis,2)$, $(4bis,4)$, $(5bis,2)$ and $(5bis,4)$ of Tab. 4 do not occur. In the proposed test (2nd row of Tab. 10) scenarios $(1bis,1)$, $(2bis,1)$, $(3bis,4)$, $(4bis,4)$ and $(5bis,4)$ (intersection on the patch vertices) are covered by the formula at line 1; scenario $(3bis,2)$ and $(5bis,4)$ (intersection on the patch segments) is covered by formula at line 2; finally scenario $(5bis,2)$ is covered by formula at line 3.
- $IT^{\partial -}_{cv,sd}(g_1, g_2)$ is true if at least one vertex of the boundary of g_1 intersects the exterior of g_2; sthe sufficient condition for obtaining this result is produced when there exists at least one vertex of the boundary of g_1 for which the scenarios of cells: $(1,1)$, $(2,2)$, $(2,4)$, $(3,2)$, $(3,4)$ and $(10,2)$ of Tab. 5 do not occur. In the proposed test (3rd row of Tab. 10) scenarios $(1,1)$, $(2,4)$, $(3,4)$(intersection on the patch vertices) are covered by the formula at line 1; scenarios $(2,2)$ and $(3,4)$(intersection on the patch segments) are covered by formula at line 2; scenario $(3,2)$ is covered by formula at line 3; finally, scenario $(10,2)$ is covered by formula at line 4.
- All other tests can be converted in a test $IT^{\circ -}_{*,*}()$ as shown in Tab. 10. This can be done also because the boundary of a surface or a solid is always a cycle, i.e. it has an empty boundary.