HAL
open science

Analyse de forme appliquée à des modèles CAO B-Rep pour extraire des symétries locales et globales

Ke Li

To cite this version:

Ke Li. Analyse de forme appliquée à des modèles CAO B-Rep pour extraire des symétries locales et globales. Autre. Université de Grenoble, 2011. Français. NNT : 2011GRENI092 . tel-00849146

HAL Id: tel-00849146

https://theses.hal.science/tel-00849146

Submitted on 30 Jul 2013

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

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

UNIVERSITÉ DE GRENOBLE

THÈSE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité Génie Industriel:conception et production

Arrêté ministériel : 7 août 2006

PRÉSENTÉE PAR

Ke LI

THÈSE DIRIGÉE PAR **Jean-Claude LÉON** ET

CO-DIRIGÉE PAR **Gilles FOUCAULT**

PRÉPARÉE AU SEIN DU

Laboratoire G-SCOP (Grenoble - Sciences pour la Conception, l'Optimisation et la Production)

DANS L'**École Doctorale**: **Ingénierie - Matériaux Mécanique Energétique Environnement Procédés Production**

Shape Analysis of B-Rep CAD Models to Extract Partial and Global Symmetries

THÈSE SOUTENUE PUBLIQUEMENT LE **10 Novembre 2011**,

DEVANT LE JURY COMPOSÉ DE:

Mme Bianca, FALCIDIENO

Directeur de Recherches, CNR IMATI-GE, Italie, Présidente

Mme Claire, LARTIGUE

Professeur, Université Paris Sud 11, Rapporteur

M Dominique, MICHELUCCI

Professeur, Université de Bourgogne, Rapporteur

M Jean-Philippe, PERNOT

Maître de conférences, Arts et Métiers Paris Tech, Examinateur

M Jean-Claude, LÉON

Professeur, Grenoble-INP, Directeur de thèse

M Gilles, FOUCAULT

Maître de conférences, Université Joseph Fourier, co-Directeur de thèse

- Acknowledgments -

First of all, I would like to thank all the members of Laboratory G-SCOP who explicitly or implicitly made the completion of this work a possibility.

I would like to thank all the members of the jury for the attention they have paid to my work and for their constructive remarks and questions. They are:

- Bianca FALCIDIENO who made me the honor of being the president of this jury,
- Claire LARTIGUE and Dominique MICHELUCCI who have accepted the difficult task of being my reviewers,
- Jean-Philippe PERNOT who has examined carefully this manuscript.

I am heartily thankful to my director Jean-Claude LÉON of thesis, whose encouragement and guidance from the initial to the final level enabled me to study and progress about the subject. His passion and motivation encouraged me during my research time. Except the scientific quality directions, he also gave me many helps for my life in France and the language.

I would like to thank Gilles FOUCAULT, my co-director of thesis. He is well on computer science and gave me many helps on the programme development and advices for the subject.

I am happy to thank all my colleagues and my friends around me during these years. Not only for work but also for life, we have had a good time together.

And finally, I would like to thank my family and my girl friend, for their love, for their encouragements, for their support from China all along these years.

Ke LI

Table of contents

Notations **1**

Introduction **3**

1 Symmetry applications and related researches **7**
1.1 Applications of symmetry in PDP 7
1.2 Symmetry basics and PDP requirements 11
1.3 Digital model representations 14
1.3.1 Boundary Representation (B-Rep) models 14
1.3.2 Feature based modeling 15
1.3.3 Discrete and faceted geometric models used for geometric analysis 16
1.4 Symmetry detection methods 20
1.4.1 Symmetry detection applied to enumerative models 20
1.4.2 Symmetry detection applied to B-Rep CAD models 31
1.5 Conclusion 34

2 Principle of the symmetry analysis approach and hypotheses **37**
2.1 Principles of the symmetry analysis approach 37
2.2 Hypotheses and shape range 40
2.2.1 Shape categories 40
2.2.2 Range of objects addressed 42
2.2.3 Shape geometry and reference surfaces 43
2.3 Maximal Surfaces and Maximal Edges Model 44
2.4 Preliminary processes to the effective symmetry analysis 46
2.5 An overview of the symmetry analysis process 47
2.6 Conclusion 49

3 Basic Input Data: B-Rep CAD model and STEP format **51**
3.1 Introduction 51
3.2 Overview of the STEP format for volume description 51

3.2.1 Structure of the major entities contributing to the definition of a B-Rep volume entity . . . 53
3.2.2 Some insight about orientation information . . . 55
3.3 From STEP format to B-Rep modeler datastructures . . . 59
3.4 Conclusion . . . 61

4 Object boundary description using a hypergraph datastructure 63
4.1 Introduction . . . 63
4.2 Hypergraph description . . . 64
4.3 Dual graph of a graph embedded in a surface . . . 66
4.4 Hypergraph creation from a B-Rep model . . . 67
4.5 Hypergraphs transformation operators . . . 71
4.5.1 Generic hypergraph transformation operators . . . 72
4.5.2 Non-manifold configuration detection . . . 75
4.5.3 Non-manifold vertex splitting . . . 79
4.5.4 Faces merging operator . . . 79
4.5.5 Edge merging operator . . . 80
4.5.6 Manifold vertex splitting operator . . . 82
4.5.7 Maximal face and maximal edge merging criteria . . . 85
4.6 Vertex splitting at regular and non-regular points . . . 89
4.7 Conclusion . . . 101

5 Symmetry planes of point sets defined from surfaces and curves 103
5.1 Introduction . . . 103
5.2 Reflective symmetry properties of the five reference surfaces . . . 104
5.3 Symmetry constraints originated from face boundary loops . . . 109
5.3.1 Symmetry constraints derived from loops over planar faces . . 114
5.3.2 Symmetry constraints derived from loops over cylindrical faces 115
5.3.3 Symmetry constraints derived from loops over conical faces . . 119
5.3.4 Symmetry constraints derived from loops over toroidal faces . . 120
5.3.5 Symmetry constraints derived from loops over spherical faces . 121
5.3.6 Synthesis about loops and loop symmetry CSP . . . 121
5.4 Self symmetry planes of a boundary loop . . . 124
5.4.1 Symmetries defined through a Candidate Symmetry Axis (CSA) 125
5.4.2 O-CSP of a maximal edge through the analysis its two adjacent faces . . . 126
5.4.3 CSAs generation . . . 130
5.4.4 Incorporating edge bounds constraints in CSP generation . . . 132

5.4.5 Generation of loop bisector symmetry planes (LB-CSP) of a face by analyzing the surrounding surfaces at a vertex 135
5.4.6 Characterizing the used area of a reference face using the orientation index 138
5.5 CSP between loops bounding a surface (LS-CSP) 141
5.6 Bisector symmetry plane between two surfaces (BS-CSP) 145
5.6.1 Processing multiple solutions of BS-CSP 147
5.7 Defining the status of face boundary loops 151
5.7.1 Deriving the status of face boundary loops from maximal edges without vertices 152
5.7.2 Deriving the status of face boundary loops from composite maximal edges 167
5.8 Conclusion 172

6 The propagation of symmetry properties 175
6.1 Introduction 175
6.2 The symmetric area surrounding a CSP or a CSA 177
6.3 First level propagation and CSP chains 183
6.4 First level Propagation with faces bounded by multiple loops 191
6.5 Axisymmetry processing during the first level propagation 193
6.5.1 Interactions between CSAs and CSPs 193
6.5.2 CSA propagation process 195
6.6 Synthesis of the first level propagation process 197
6.7 Two sides propagation 200
6.7.1 Principle of the 'two sides' propagation process 200
6.7.2 Introducing the SS-CSP category 203
6.7.3 Processing faces with multiple loops 204
6.8 Highlighting asymmetry in M_{MAX} 207
6.9 Conclusion 210

7 The examples and applications for reflective symmetry detection 213
7.1 Introduction 213
7.2 Global symmetry detection 214
7.3 Highlighting asymmetry and modification suggestions 225
7.4 Axisymmetry modification under asymmetry suggestions 230
7.5 Conclusion 234

8 Conclusions and perspectives 235

A Combinatorial analysis of producing CSPs 241

B Details about the loop status analysis **253**
B.1 Cylinder and cone 253
B.2 Cylinder and torus 254
B.3 Cone and cone 260
B.4 Cone and torus, torus and torus 264

C More examples and statistic **265**

Notations

Here are gathered together the different notations and major acronyms used all along the document.

S	geometric surface of a B-Rep model supporting a face F
Σ	geometric curve of a B-Rep model supporting an edge E
P	geometric point of a B-Rep model supporting a vertex V
F	topological face in a hypergraph describing a B-Rep model (can be used to designate face of M_{MAX} or M_I)
E	topological edge in a hypergraph describing a B-Rep model (can be used to designate edge of M_{MAX} or M_I)
V	topological vertex in a hypergraph describing a B-Rep model
Γ	loop bounding a face F
N	node in a hypergraph
A	arc in a hypergraph
R	rank of a node or an arc in a hypergraph
$[\ \ ]$	entity set
$[F_{\Omega V_i}]$	faces set surrounding vertex V_i
$\widetilde{F}$	homologous face of F
M_I	B-Rep CAD model input from a STEP file
M_{MI}	Shape as M_I after maximal faces generation
M_{MAX}	Shape as M_I describing with maximal edges and faces
P	Plane
Cy	Cylinder
Co	Cone
Sp	Sphere
To	Torus
CSP	Candidate symmetry plane
CSA	Candidate symmetry axis
GSP	Global symmetry plane
GSA	Global symmetry axis
BS-CSP	Bisector CSP
LB-CSP	Loop bisector CSP
LS-CSP	Loop symmetry CSP
O-CSP	Orthogonal CSP
SS-CSP	Surface symmetry CSP

Introduction

Digital Product Development (DPD) is a digital platform of traditional Product Development Processes (PDPs). Under the help of more powerful computers, DPD is more efficient and more accurate. A digital product document is easy to use for knowledge management and product management. In this case, recently, with the progressing of the computer technology and the needs of Product Lifecycle Management (PLM), DPD has been in rapid development and wide use.

Whatever the top-down or bottom-up design, it always needs the processes of digital model generation, model analysis and model modification (see Figure 1). And these processes should normally be repeated several times. As in the DPD, the digital model is translated and modified over time. A good digital model should stay convenient for the whole development and production processes.

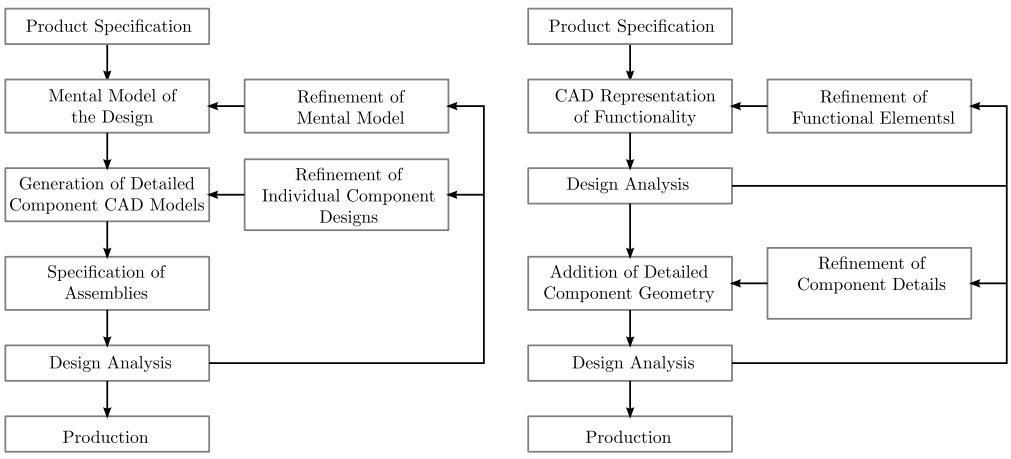


Figure 1: Bottom-up and top-down design [60].

Whether in natural objects or man made products, symmetry is ubiquitous. Symmetry properties of components or products or, in a larger extent shapes, have many applications. Symmetry information has been used for human face recognition [56] [55], 3D model storage and retrieval [43]. In a PDP, using symmetries can help sim-

plifying a structure subjected to a FEA or retrieving similar models from a database. Also, symmetry properties can be useful when modifying an object or a product to characterize some of its shape features. Manufacturing operations and tool path trajectories could take advantage of symmetries in an object, e.g. to define parting planes in stamping or molding processes. Also, shape theory strongly relies on symmetry to analyze them and connect with a generative process [34].

Generally, some symmetry information can be attached to a component such as mirror functions, copy operations with respect to a plane and pattern operations when a designer models an object. However, if these operations lead to symmetries, they are embedded in the object related to a specific design software but they are not explicit. Hence, it is difficult to take advantage of them. Anyhow, not all the symmetries are available because the designer can generate symmetries with standard extrude or pocket operations. Here, the purpose is to make explicit the symmetry properties of an object, whatever the modeling operators used during its design process.

Otherwise, not all the objects are absolutely and globally symmetric. Approximate symmetry is more general. During the numerous analyses in a PDP, such as FEA, the details which makes asymmetry may not be important, but create problems. Based on the symmetry analysis, highlighting asymmetries of a component is useful for its simplification and/or modifications. On the other hand, for completely asymmetric models, the partial symmetry is also very useful for the same reasons. If the model is asymmetric and the asymmetric area is not a detail, detecting the partial symmetry is also interesting.

Nowadays, there are some researches related to symmetry detection based on different kinds of digital models. However, these researches are weak on calculation complexity and accuracy. Also, they cannot locate asymmetry nor detect partial symmetries. So, CAD softwares don't contain a symmetry analysis function.

The considerations above support the main motivation of the present work. We propose an algorithm to generate symmetry planes of 3D models using the B-Rep model of CAD volumes. Design processes of volume models strongly rely on extrusion and revolution primitives from sketches containing essentially straight line segments and circular arcs. Hence, the boundary surfaces considered are planes, cylinders, cones, tori and spheres. Global geometry properties help characterizing the symmetry planes that can initiate the global symmetry planes of the object. The intersection curves of these reference surfaces provide strong properties to characterize possible global symmetry planes or a global symmetry axis of the object. Then, the algorithm starts from analyzing the symmetry properties of the faces followed by the intersection curves between them. Subsequently, the candidate symmetry planes set up contains all the possible global symmetry planes. Finally, the properties of neighboring surfaces and the use of intrinsic parameters of surfaces help determining robustly the global symmetry planes, whether there is a finite number or an infinite number (axisymme-

try) that produces a global symmetry axis.

The manuscript structure is as follows:

- **Chapter 1:** gives an introduction about the applications of symmetry properties during a PDP and the definition about **Reflective Symmetry**. Using illustrations of symmetry detections, different approaches valid on different formats of digital models are reviewed: point sets models, mesh models and B-Rep models. Some of their limitations are highlighted and analyzed;

- **Chapter 2:** proposes the hypotheses of our symmetry analysis approach. It is valid for 2-manifold B-Rep models and their surface is restricted to the combination of the five reference surfaces: plane, cylinder, cone, sphere and torus. Each face and edge of the object boundary is regarded as a infinite point set and the proposed approach concentrates on the symmetry properties of these point sets and their extension to the whole object. Globally, the symmetry analysis appears as a divide and conquer process starting from a neutral representation of a B-Rep model;

- **Chapter 3:** presents an analysis of the structure of the STEP format, a neutral B-Rep model representation. It is an ISO standard for model transformation widely used as digital model format for all CAD softwares. The five reference surfaces are directly accessible in the STEP format, which give enough support for the symmetry detection process. Their intrinsic parameters are the only parameters used during the symmetry anamysis;

- **Chapter 4:** introduces the model preparation process that is needed to produce a boundary decomposition independent from any modeling process and that must be compatible with the symmetry analysis. It appears that a data structure based on hypergraphs is needed to meet these requirements. As a graph representation, it has graph operators which are convenient for element merging and splitting and can be used to produce a boundary decomposition into maximal faces and edges. It is created from a STEP datastructure and keeps a link with the parameters of this structure. A sub-graph named **Dual Graph** is also introduced. Hypergraphs support the whole symmetry detection algorithm;

- **Chapter 5:** describes the symmetry properties attached to faces and edges of a B-Rep model. These faces and edges being regarded as infinite point sets. The corresponding symmetry properties produce Candidate Symmetry Planes. It is the divide phase of the symmetry analysis. CSPs are generated solely from the basic symmetry properties of the five reference surfaces. There are categories of CSPs. The **Orthogonal CSP** (O-CSP) is orthogonal to an edge and created by the two neighboring faces. The **Loop Bisector CSP** (LB-CSP) is attached to

a vertex and reflects a symmetry of its two adjacent edges. It is a bisector plane of a loop constrained by its two neighboring faces. The **Loop Symmetry CSP** (LS-CSP) is the symmetry information within a face containing multiple loops. The **Bisector CSP** (BS-CSP) coincides with an edge contained in a bisector plane produced by its adjacent faces. The O-CSP evolves into a **Candidate Symmetry Axis** (CSA) when the intersection between two defines an infinite number of symmetry planes;

- **Chapter 6:** introduces the conquer phase of the symmetry analysis. It is founded on two levels of propagation. In a first place, the symmetric area of a CSP propagates to cover the largest possible area of the object boundary. This process generates CSP chains. In a second place, the propagation expands over the boundary of the object on both sides of each CSP chain to cover the whole boundary. If so, a global symmetry plane or a global symmetry axis is obtained. The overall algorithmic complexity of the divide and conquer process appears as mostly linear with respect to faces, edges and vertices of the model. This process produces also local symmetry information. Most of the symmetry and asymmetry over the object can be obtained within the same process;

- **Chapter 7:** gives some illustrations of the symmetry analysis and its divide and conquer algorithm. The efficiency of the algorithm is analyzed. The analysis shows clearly the areas of symmetry and asymmetry. The use of symmetry analysis for model modification or simplification is also highlighted.

- **Chapter 8:** discusses the advantage and drawbacks of the approach. A synthesis of the proposed approach is also given. Then, perspectives are also argued.

Chapter 1

Applications of symmetry properties and related researches about symmetry detection

This chapter describes the context of the manuscript, investigating the application of symmetry properties during a product development process. For different product development processes need different digital models, they are reviewed with respect to symmetry properties extraction. Under the reflective symmetry definition, there are some researches about symmetry detection based on different input models. This chapter analyses these methods. Most of the researches focus on enumerative input models, their limitations relate to their detailed representations and the high computational complexity to extract symmetry information. B-Rep model input for symmetry detection is also addressed. It has the advantage of containing all the details of a model but faces limitations regarding the recognition of boundary surfaces and current approaches about symmetry detection mainly rely on heuristics.

1.1 Applications of symmetry in a product development process and in the large

The product development process (PDP) deals with all the stages from design to manufacturing, assembly, maintenance aspects of mechanical components. In the context of a PDP, symmetries can be used to simplify the computational domain of a shape subjected to structural analysis, such as Finite Element Analysis (FEA). In another context of shape modifications, symmetry properties can be useful to modify an object or a product while maintaining the symmetry of some of its form features. In the context of a manufacturing process preparation, manufacturing operations and tool path trajectories could take advantage of symmetries in an object, e.g. to define parting planes in stamping or molding processes. Also, shape theory strongly relies on symmetry to analyze them and connect with a generative process [34].

When a designer uses a CAD modeler to create a 3D shape representation for a manufactured product, some symmetry information can create some local symmetries

in a shape such as mirror operations, pattern copy operations, revolution of sketches (axisymmetry). However, if these operations lead to shape construction features that are symmetric (see Figure 1.10), these construction features are embedded in the object but they are not explicit, hence it is difficult to take advantage of them. Anyhow, not all the symmetries are available because the designer can generate symmetries with standard extrude or pocket operations. In the context of this manuscript, the purpose is to make explicit the symmetry properties of an object, whatever the modeling operators used during its design process.

In FEA, the shape of parts has to be adapted by the mechanical engineer to take into account the mechanical behavior hypotheses of the analysis. The shape adaptation procedure combines the following operations:

- The removal of form features that are considered as details because they can be removed while respecting the targeted accuracy of the mechanical simulation;
- The cut of the shape along some of its symmetry planes that are relevant for the analysis (see Figure 1.1). A problem is symmetric according to a reflective symmetry plane P when all the following requirements are met: the shape is symmetric with respect to P, the boundary conditions are symmetric with respect to P;
- The transformation of a 3D simulation problem that features a symmetry axis into a 2D problem that is revolved around this axis.

When the shape is symmetric, while the boundary conditions of the analysis don't share the same symmetry properties, the mesh generation process can take advantage of the shape symmetry as follows:

- The mesh of a shape featuring one or more reflective symmetry planes can be generated on one of the symmetry planes, and copied by reflective symmetry operations to complete the mesh. The main advantage of this approach is a reduction of the time required to mesh the part;
- The mesh of a shape featuring a symmetry axis can be generated by the revolution of the planar section of the mesh around the symmetry axis. The quality of mesh element shapes is optimal while the time of the mesh generation process is drastically reduced (see Figure 1.2).

Many research works focused on the application of symmetry properties of shapes in other contexts than a PDP. Even though they were devoted to other contexts, some of them could be relevant in a PDP context. Many works [13][55][56] focused on the detection of human face features and structuring features in photographs using generalized symmetry properties. Other works focused on the semantic enrichment and

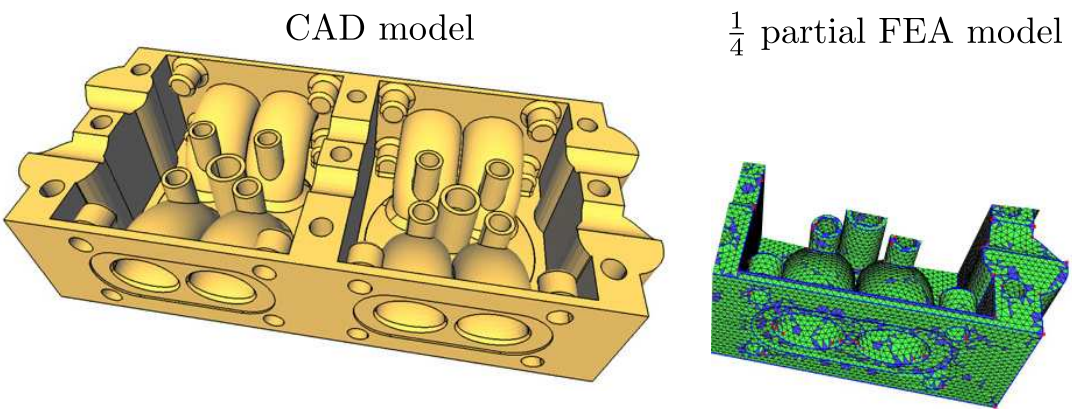


Figure 1.1: From a CAD model to its symmetric partial mesh model.

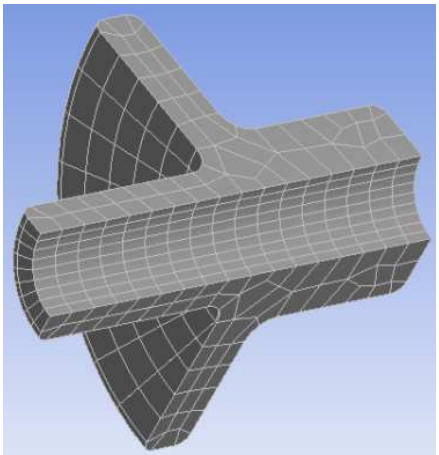


Figure 1.2: A hexahedral mesh generated by the revolution of the 2D section around the symmetry axis.

digital storage of 3D models by generating a hierarchical structure of symmetries on mesh models [62], facilitating the compression and indexing of symmetric 3D models for storage and retrieval (see Figure 1.4). In the context of a PDP, component search and retrieval is also of interest to exploit large industrial databases, e.g. in the aeronautic and automotive industries. In Mitra's [44] work, an algorithm based on curvature invariants under symmetry transformations was proposed to detect partial reflective symmetries, rotational symmetries, and scalings of local features in 3D triangulated models (see Figure 1.3). These properties are also useful in the engineering context even though the challenge is to process CAD models rather than meshes. In biomedicine, many biological structures are symmetric or approximately symmetric. Symmetry detection can cover symmetry/asymmetry information from regions of the head and explore its implication with respect to positive clinical findings [37].

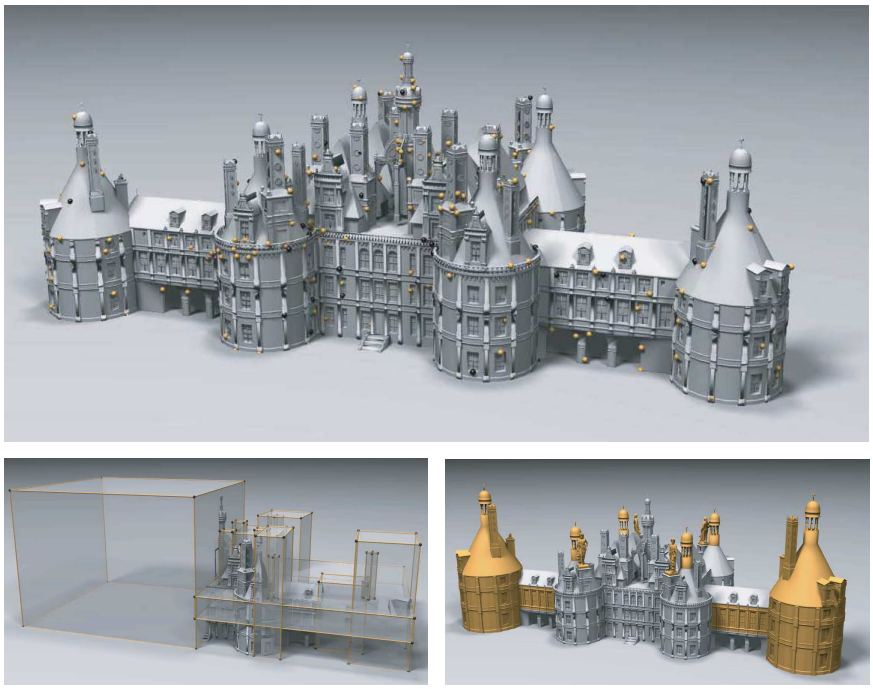


Figure 1.3: The compression of geometric model using its symmetry properties [44].

Symmetry is an intrinsic property of the nature or man made object. The symmetry detection and symmetry modelling operations are ubiquitous in product design, shape analysis, shape recognition, similarity matching, synthesis and reconstruction[54]. Symmetry can be broken by some anomaly or abnormal behavior[53]. In most configurations, symmetry is not strict and accurate[38]. The self-similarity is the most general case. During the simulation, modifying the asymmetry part to get a symmetric model is often required to simplify the problem. Therefore, the detection of approximative symmetries is important in numerous practical applications, including PDP, crystallography, medical imaging, face recognition, 3D model storage, semantic enrichment of 3D models, to mention a few[52].

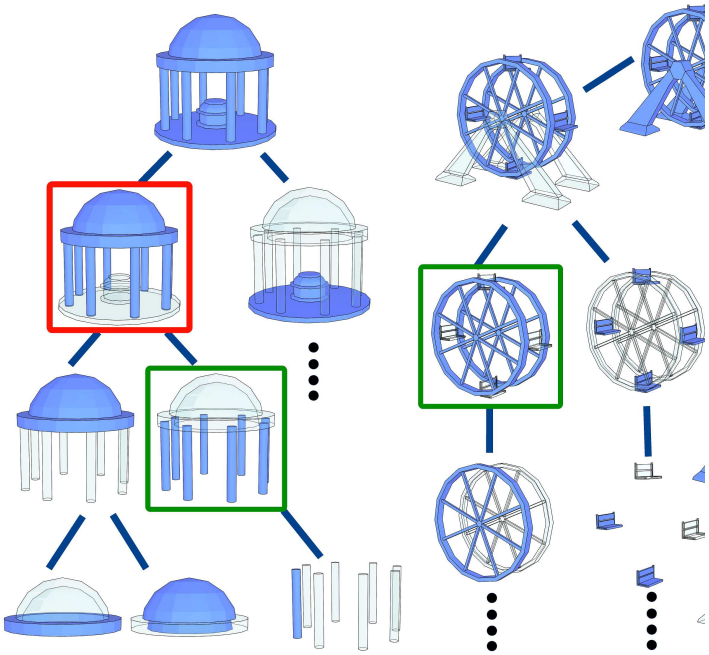


Figure 1.4: Hierarchical structure providing an organization of a 3D model [62].

1.2 Symmetry basics and PDP requirements

Formally, symmetry is an isometric property between sets of Euclidean groups of 2D or 3D shapes in Euclidean space. In $I\!R^2$, reflection and rotation are two types of symmetries which characterize axes and points as symmetry entities, (see Figure 1.5). Reflective symmetry can be further refined with self reflective symmetry as a specific subset of reflective symmetries. In $I\!R^3$, reflective, axial, and spherical symmetries are three types of symmetries which characterize planes, axes, and points as symmetry entities (see Figure 1.6).

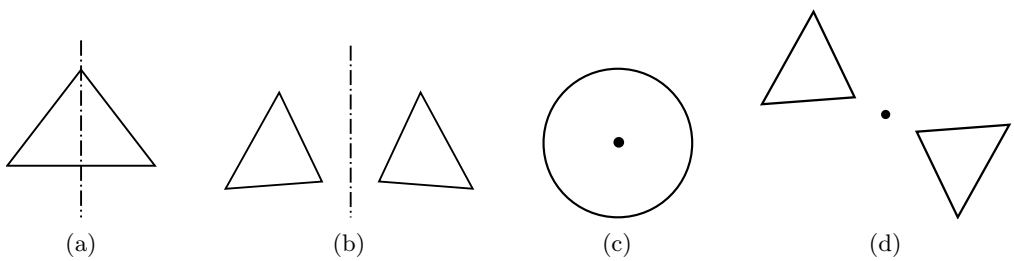
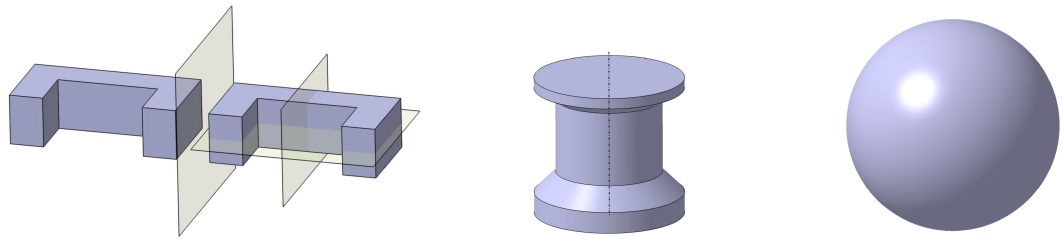


Figure 1.5: Reflective and rotational symmetry examples in 2D: (a) self reflective symmetry, (b) reflective symmetry of two contours with respect to an axis, (c) rotational symmetry, (d) rotational symmetry of two contours with respect to a point.



(a) (b) (c)

Figure 1.6: Reflective and rotational symmetry examples in 3D: (a) reflective and self reflective symmetry, (b) axial symmetry, (c) spherical symmetry.

It has to be noticed that Figures 1.5c, 1.6b and c illustrate configurations where there are infinite numbers of reflective symmetry entities, i.e. axes in 2D, planes in 3D.

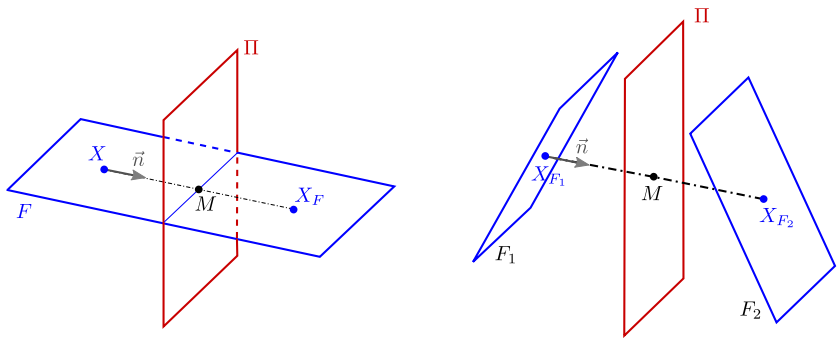
During a PDP, the simulation behavior of components is increasingly important and symmetry properties of components are always used to simplify the domain of study when the boundary conditions applied on a component satisfy the same symmetry properties. Axisymmetric configurations are also of particular interest to generate even simpler simulation models but they rely on reflective symmetry configurations where there are an infinite number of symmetry planes (see Figure 1.6b). Similarly, when designing an object, symmetry properties are difficult to incorporate because the designer is not always able to plan the symmetry operators needed during an object modeling process. Additionally, shape modifications required during a PDP can generate symmetries where they were none. Detecting symmetries can be helpful when dimensioning, tolerancing either for functional or manufacturing purposes even though their modeling process does not contain this symmetry information.

Summarizing the above observations, reflective and rotational symmetries are the main categories of symmetries useful for some PDP stages, i.e. central symmetry [44] does not show up. Reflective symmetry appears as prominent to generate simulation models and a strong contributor to other models used in a PDP. Rotational symmetry is useful to set up periodic boundary conditions in simulations but is much less frequent than reflective symmetry. As a first step, this research focuses on reflective symmetry.

Definition 1 *: Self reflective symmetry*

Given an arbitrary point $X \in I\!R^3$ on a reference surface F and a symmetry plane Π, there exists a unique point X_F such that X_F belongs to F and satisfies:

$$\overrightarrow{MX_F} = -\overrightarrow{MX} = -\|\overrightarrow{MX_F}\| \cdot \vec{n} \tag{1.1}$$



(a) (b)

Figure 1.7: Illustration of symmetry definitions with respect to a plane Π: (a) the self symmetry; (b) the reflective symmetry of two surfaces.

where F denotes a bounded reference surface, M is the normal projection of X on Π , $\vec{n}$ is the normal to X at M. X_F is called the image of F, i.e. the symmetric point of X in F. Conversely, if any point X of F has a symmetric point X_F lying in F through Π, then Π is a symmetry plane for F.

For two surfaces, the definition is similar.

Definition 2 *: Reflective symmetry of two surfaces with respect to a plane* Π

Let two bounded surfaces be F_1 and F_2 then, for every point $X_{F_1} \in F_1$, $X_{F_1} \subset I\!R^3$ and a symmetry plane Π, there exists a unique point X_{F_2} such that $X_{F_2} \in F_2$ and a point $M \in \Pi$ satisfying:

$$\overrightarrow{MX_{F_2}} = -\overrightarrow{MX_{F_1}} = -\|\overrightarrow{MX_{F_2}}\| \cdot \vec{n} \tag{1.2}$$

where F_1 and F_2 denote two reference surfaces each one bounded by one loop at least, M is the normal projection of X_{F_1} on Π, $\vec{n}$ is the normal to M going through X_{F_1}. X_{F_2} is called the image of X_{F_1} on F_2, i.e. the symmetric point of X_{F_1} in F_2. Because this property holds for the infinite point set defined by F_1, F_2 is called the image of F_1. Conversely, if any point $X_{F_1} \in F_1$ has a symmetric point X_{F_2} lying in F_2 through Π, then Π is a symmetry plane for F_1 and F_2. Similarly, this definition applies also to space curves, too. These two definitions of reflective symmetry support this research.

1.3 Digital model representations

In this section, we introduce the major categories of geometric models that are used for the description of components in a PDP, as well as other models that are used by algorithms to identify the symmetry properties. We highlight some geometric transformations that are applied on the models of manufactured products to simulate their mechanical behavior with FEA or used to convert them into discrete geometric models such as point sets, voxelizations and meshes.

1.3.1 Boundary Representation (B-Rep) models

B-Rep models are widespread in current commercial CAD modelers to represent volume objects, otherwise called solids. B-Rep models represent solids by describing their boundary surface with faces, edges, and vertices that separate the interior domain from the exterior of the solid.

This model combines two types of entities:

- **Topological entities,** which describe how geometric entities are connected in the solid model (solids, face, edge, vertex, co-face, co-edge, shell, loop);
- **Geometric entities,** which define surfaces (cylinder, sphere, cone, torus, plane, NURBS), curves (line, circle, ellipse, parabola, NURBS), and points.

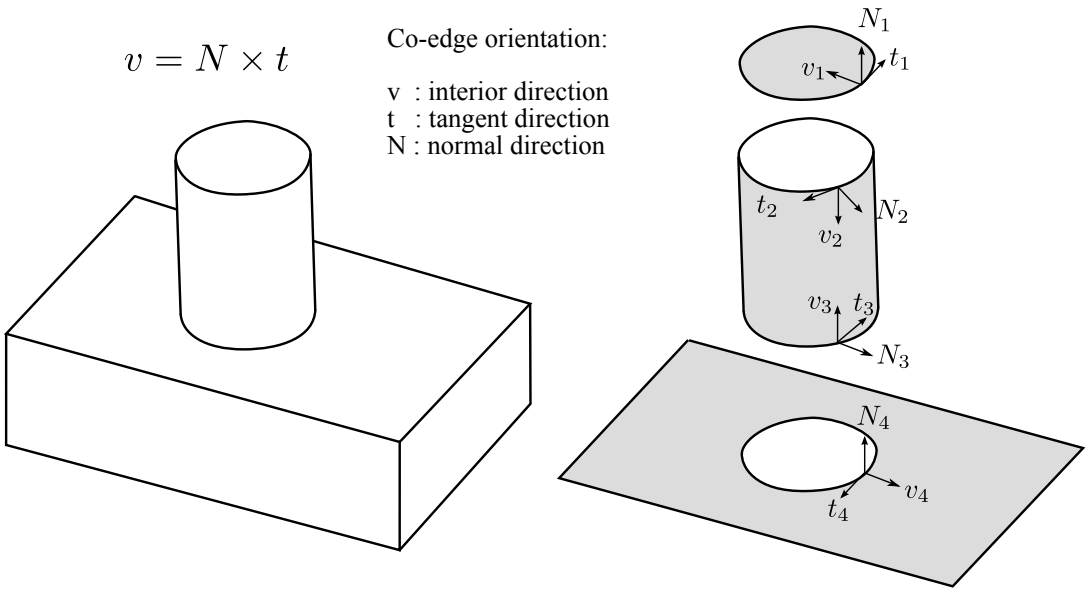


Figure 1.8: Right-hand rule of co-edges in B-Rep models.

A **solid** is bounded by a set of closed surfaces called shells and organized as follows:

- Each **shell** is a connex set of co-faces (faces oriented with the normal towards the exterior of the solid);
- Each **co-face** is a face oriented with its normal towards the exterior of the solid;
- Each **face** is a surface patch whose boundary is defined by a set of loops;
- Each **loop** is a cycle of co-edges representing a boundary of a face;
- Each **co-edge** is an oriented edge defining a part of the boundary of a face. As shown in Figure 1.8, co-edges are oriented such that $\vec{t}$ (tangent vector) and $\vec{N}$ (face normal) give $\vec{v}$ (direction orthogonal to the co-edge and towards the interior side of the face): $\vec{v} = \vec{N} \wedge \vec{t}$;
- Each **edge** is the portion of a curve whose bounds are set by two vertices;
- Each **vertex** is a point bounding one or several edges.

Here, the focus is set on B-Rep CAD models that contain only canonical surfaces such as cylinders, cones, planes, tori, spheres, and free-form surfaces of NURBS type.

Some conditions must hold to ensure that a B-Rep model represents a valid solid. These conditions summarize:

- Faces may intersect only at common edges;
- Each edge is shared by exactly two faces;
- Faces around each vertex can be arranged in a cyclical sequence such that each consecutive pair shares an edge incident to that vertex.

1.3.2 Feature based modeling

While B-Rep models enable the accurate representation of complex shapes, they do not convey semantic information about the object (a B-Rep model only represents geometric and topological information). Manufactured parts are designed from primitive shapes having a functional meaning such as holes, slots, etc. To incorporate more meaning, B-Rep models have been enriched by attaching 'feature' entities to groups of topological entities. Features aim at representing form aspects or manufacturing aspects or other attributes attached to sub domains that constitute the shape of objects. Initially, features where restricted to manufacturing entities (holes, chamfers, etc.). CAD modelers adopted generic and application-independent features to create B-Rep objects through a construction tree approach.

In the feature based modeling approach, the designer combines form features available in a reference library (see Figure 1.9):

- **reference geometric entities**: planes, axis, points;
- **2D sketching primitives**: creation of planar sketches composed of free-form curves, points, axes, lines, circles, etc;
- **extrusion/revolution of 2D sketches to add/remove material:** 2D sketches representing a plane bounded by a set of contours are swept along an axis (extrusion) or a circle (revolution) to add or remove material to the object;
- **surface operators:** edge filleting operation is a kind of surface blend operation that replaces sharp edges by curved surfaces having a constant radius of curvature. Surface filling operations fill a hole in a surface with a patch that can be tangent up to curvature continuous with the contours;
- **symmetry operators** (see Figure 1.10): 3D features can be duplicated using a symmetry plane, a linear pattern monitored by a direction and an offset distance, a circular pattern monitored by an arc of circle and an angle.

The parametric representation of features enables changes in the dimensions, geometric constraints and attributes of features. Because the geometric constraints of each step of a feature construction declared in a CAD modeler is defined with respect to the pre-existing entities of this B-Rep model, each feature depends upon its parent features that represent the construction steps of the entities underlying the geometric constraints of the current construction step. The consistency of the model is difficult to maintain when modification occur on features having many children attached to them.

1.3.3 Discrete and faceted geometric models used for geometric analysis

Discrete and faceted geometric models contrast with continuous geometric models such as CAD B-Rep models by the fact that they represent solids with a finite set of elements (for example: points, grid cells, triangles).

Point sets models represent a set of points in $\mathbb{R}^3$, obtained by sampling the surface of a digital or a real object:

- Points can be generated randomly on the surface of B-Rep models or triangulations, until the stopping criterion is met (for example, the point count or the target distance between neighbors) (see Figure 1.11);
- Points can be obtained by scanning real objects with a 3D scanner like coordinate measuring machines (CMM) or laser scanners.

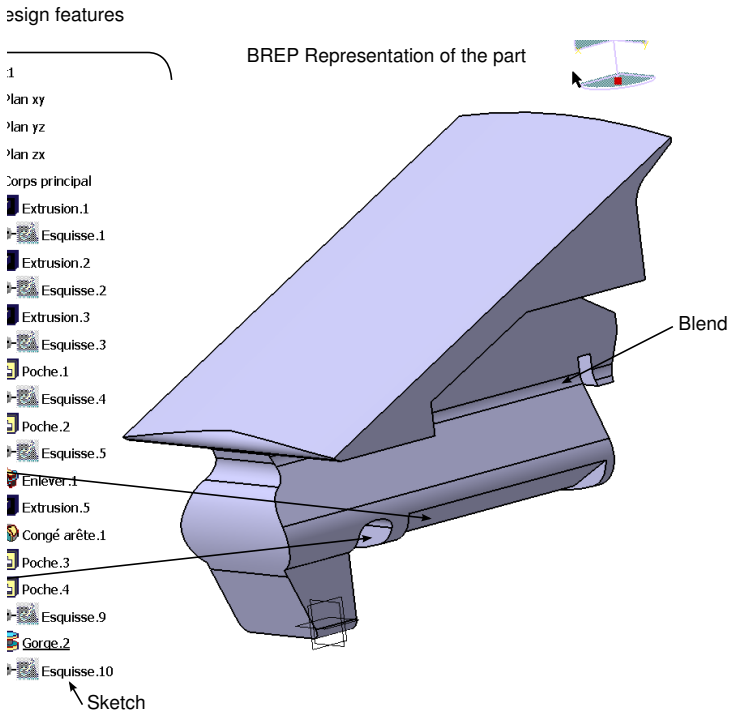


Figure 1.9: Example of a feature based modeling process in CATIA V5 CAD modeler.

Voxel models represent solids using a regular grid where each cell is assigned a boolean information: full or empty (see Figure 1.12).

A mesh model is a collection of vertices, edges, and faces that defines the shape of a polyhedron enclosing a volume (see Figure 1.13). Faces can be triangles, quadrilaterals or other simple convex polygons. Faces are bounded by edges and each edge is bounded by two vertices. Consequently, it is a category of B-Rep model where faces are planar.

FEA uses surface meshes to simulate the behavior of shells or plates, and volume meshes composed of hexahedrons or tetrahedrons to simulate the behavior of solids.

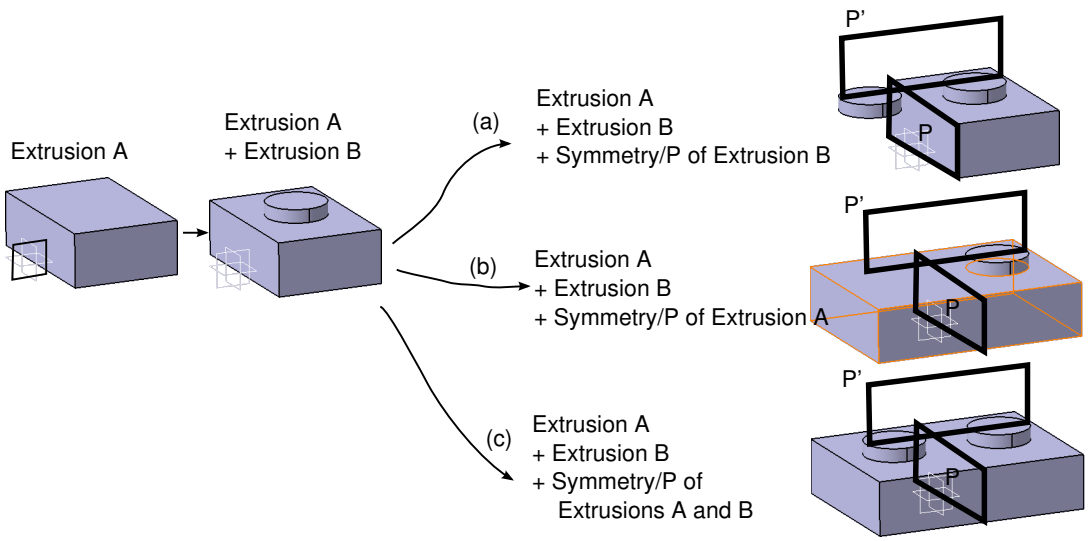
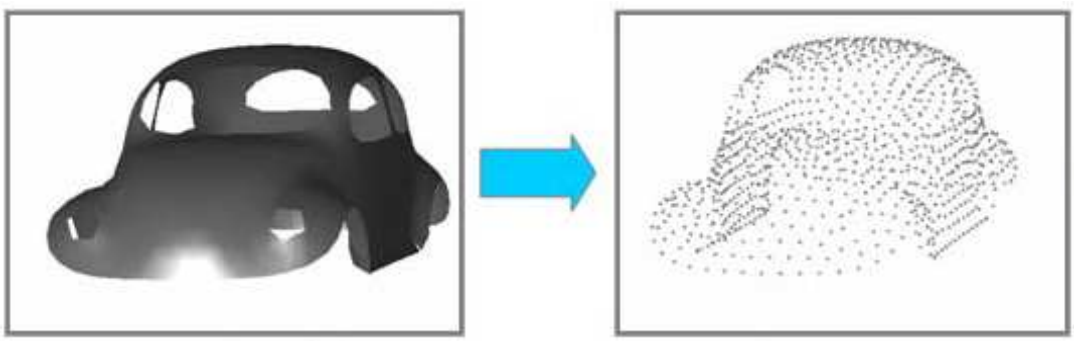


Figure 1.10: Example of the application of a symmetry feature in CATIA V5 CAD modeler: (a) and (b) the model is partially symmetric; (c) the model is globally symmetric. None of the feature based models represent explicitly the symmetry plane P'.



Real object Point set model

Figure 1.11: An example of point set model [10].

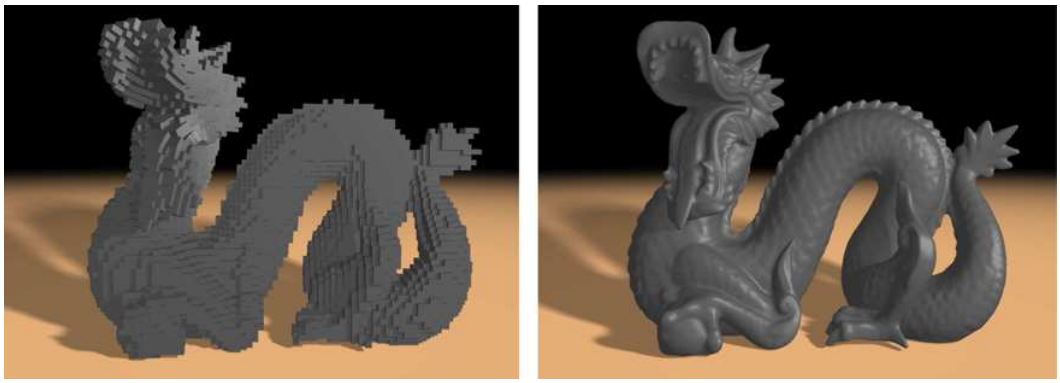


Figure 1.12: An example of voxel model [16].

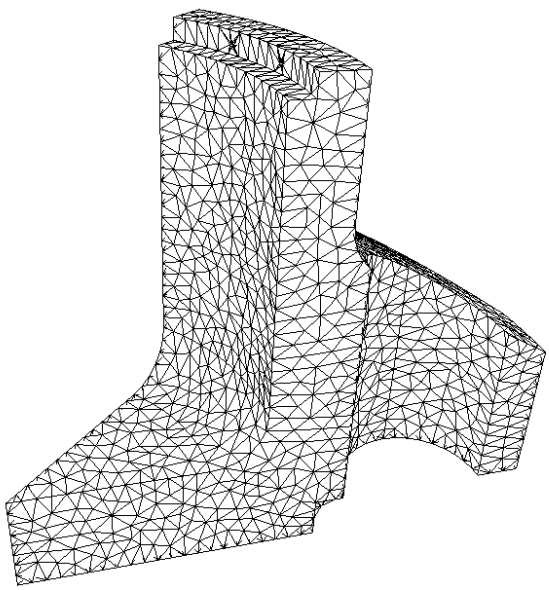


Figure 1.13: An example of mesh model.

1.4 Symmetry detection methods

As illustrated through the previous sections, the symmetry properties of a shape find many applications during a PDP as well as in the large. With the development of computer aided geometric modeling and CAD technology, the research about symmetry detection and analysis is still an up to date topic. So, symmetry detection focuses on digital models and the corresponding algorithms have to adapt to the geometric and topological properties of these models as well as to their data structures. Consequently, for different categories of digital models, the symmetry detection and analysis algorithms differ. It is the purpose of this section to address some of them.

1.4.1 Symmetry detection applied to enumerative models

Related to data storage techniques, early research about symmetry detection started from datasets of 2D points [4]. The input data are point located on contours created by geometric entities such as: segments, circles, points, etc. Regarding the adjacency relationships between these entities, they help naming these relations with respect to geometric properties and transforming them into a string. Then, closing a chain derived from these strings to form a loop characterizes the shape symmetry properties. With a similar algorithm, Wolter [64] extended the method to polygons and polyhedrons. Within his method, the points of the objects are structured into cycles, then each cycle is encoded into a string of symbols. At the end, the algorithm tests the symmetry of the encoded strings.

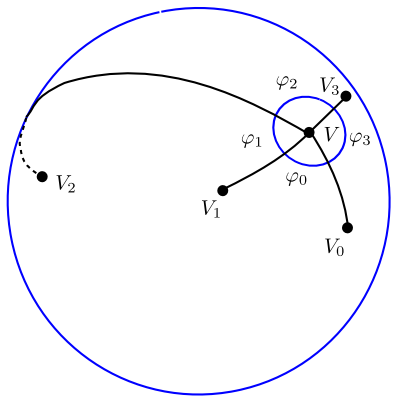


Figure 1.14: The projection of a 3D point on a unit sphere producing a point V with two identifiers at V: the distance between the 3D point and the centroid; the angle ϕ_i on the sphere between V and its neighbors: V_i, V, V_{i+1}.

Later, some reports have addressed approximate symmetry detection for 3D do-

mains, where input data were 3D point sets. The symmetry information is derived from a bipartite graph comparison method. Alt [3] reported an algorithm that estimates the congruency of two point sets A and B, their similarity index and symmetry properties. This algorithm was projecting 3D points on a unit sphere whose origin is located at the centroid of each point set, each point V thus obtained was assigned a distance and an angle parameter (see Figure 1.14). All the points V are vertices of convex hulls representing A and B.

Then, considering each convex hull as a labeled planar graph, the label of each point V containing two identifiers (see Figure 1.14), the algorithm could find out whether or not the two input point sets A and B were isomorphic. Then, the authors gave some criteria for grouping the vertices in order to detect symmetry or congruence structures.

Iwanowski [27] showed a reduction algorithm based on a ϵ-approximative index to group the point set as previous work to the symmetry detection in order to reduce the cost of the approximate symmetry detection. The approximate symmetry analysis being an NP-hard problem. Later, Brass [9] made some optimization but the basic theory has not changed. The complexity of the method for symmetry detection is $n \log n$.

The Hausdorff distance is widely used as a measure of shape similarity in computational geometry [66][67][50][21][46] since it is commutative.

Given two sets of points X and Y, the definition of the Hausdorff distance is:

$$d_H(X,Y) = max\{\underset{x \in X}{sup} \ \underset{y \in Y}{inf} \ d(x,y), \underset{y \in Y}{sup} \ \underset{x \in X}{inf} \ d(y,x)\} \tag{1.3}$$

where sup represents the superior bound and inf the inferior bound of the values (see Figure 1.15). If the Hausdorff distance is 0, the two point sets are congruent.

Hausdorff distance measurement has been used to determine the degree to which two shapes differ from one another in pattern recognition and machine vision [25][24][12]. The definition shows that shape matching algorithms can evaluate efficiently two point sets with a tolerance control but they give results that are still approximations with respect to real objects. In addition, these results are obtained at expensive computational costs. To determine the congruence between two planar point sets of size n using a Hausdorff distance criterion, the computational cost in time is $O(n^5)$ [31]. In Krishnan's work, leveraged FFT-based techniques for string matching were combined with Hausdorff distance methods to compute the symmetry and approximate symmetry of point sets. Here, string matching can be performed in a time complexity of $s \log s$ where s is the total length of the string [31] even though the connection between string length and n is harder to establish. Complementary, the memory storage complexity of the strings can be yielding an overall bound of $O(n^3 \log n)$ but there is

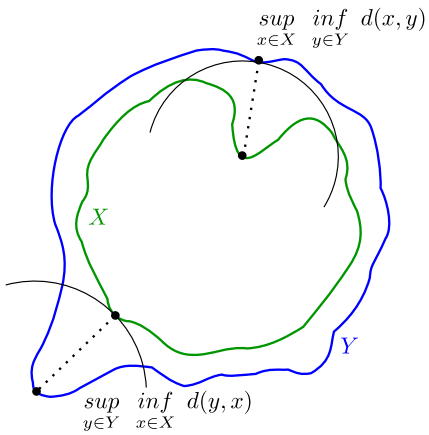


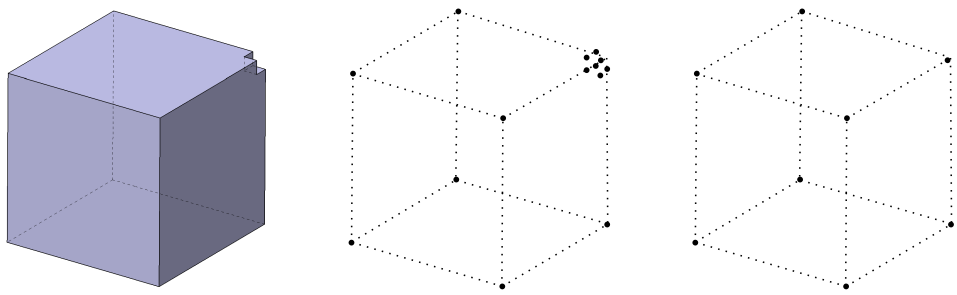
Figure 1.15: The Hausdorff distance between X and Y [63].

the mandatory price for FFT computations to be taken into account.

Whether the Hausdorff distance or the methods mentioned before form the basis of symmetry detection algorithms, point sets symmetry detection has limitations. The first one is that the comparison process needs two point sets as input. There, many informations and details are lost compared to the B-Rep CAD model of a component or the physical object. Then, the algorithm results in an approximate symmetry detection. In the context of shape recognition, approximate symmetry properties are expected, but with applications in the field of mechanical engineering and PDPs, approximate results don't meet the accuracy requirements of the simulations taking place in that context. To be useful, symmetry properties must be produced at the accuracy level of B-Rep CAD models and in interactive time. A solution to improve the accuracy is to increase the density of sampled points over the digital model. A second limitation shows up that is computational complexity which becomes prohibitive on large point sets. Indeed, these two limitations interact with each other.

Subsequently, B-Rep models are considered as input. Mills et al. [41][33] tried to use key vertices instead of point sets model. This operation can efficiently reduce the number of points (see Figure 1.16b). Next, the author replaces certain groups of points by their centroid to reduce further the amount of points to be processed for symmetry detection. Then, the algorithm finds the approximate symmetries of the collection of points remaining after this grouping. The time performance of the algorithm in the worst case is bounded by $O(n^{3.5} \log^4 n)$ where n is the number of key points. Due to the reduced point set extracted, the extracted symmetries lack of robustness.

Ming Li et al. [35][36] proposed a similar symmetry detection method. The original input data is a B-Rep model and it is translated into a discrete model as a point set.



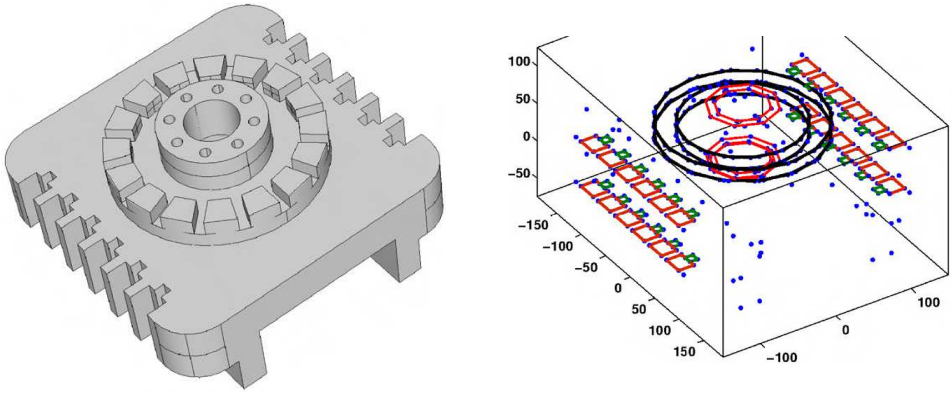
(a) (b) (c)

Figure 1.16: The major steps of Mills operation to extract key vertices [33]: (a) B-Rep model input; (b) key points replacing the B-Rep model; (c) key points after a grouping operation.

The point set model is the same as the one based on key points as defined by Mills (see Figure 1.17). Then, the method looks for symmetric cycles. The time complexity is evaluated and divided into two parts. Incomplete cycles are determined in $O(p \times s)$ time for a point set of size p and s initial seed sets. Then, the clustering of c cycles takes $O(c^2 \log c)$.

Whatever Mills' model or Ming Li's point set models derived from a B-Rep CAD model, their purpose is to reduce the boundary of the model that define an infinite point set into a finite one to describe the initial model. The B-Rep model gives them a good opportunity to translate a smooth model into a discrete one. But when ignoring the curves and surfaces informations, the discrete point model looses important symmetry information, e.g. infinite numbers of symmetry planes forming axisymmetry is not available and the positions of symmetry planes become approximate. In this framework, many different shapes can have the same discrete point sample. Again, these algorithms are only for approximate symmetry detection. For example, in Figure 1.16, the cube misses one corner, after grouping the key points, thus removing the shape detail. In the engineering context, such a shape detail, if removed, must be monitored by an engineering criterion, e.g. size of finite elements, rather than a sampling criterion that may not be accessible to the engineer.

Berner's algorithm of symmetry detection expects a point sampled representation of the manifold model used as input [5][6][7][8]. The input model is also a point set model and the first step is to translate the model into a point sampled model, which is the same as Ming Li's input model. Then, the algorithm creates segments linking adjacent points sampled on the surface of the model, which are named for use during a feature detection process. This produces a feature graph with edges and points: $G = (\nu, \epsilon)$. The task is now to find a subset of graph G: $U_G \subseteq G$, and a set of discrete



(a) (b)

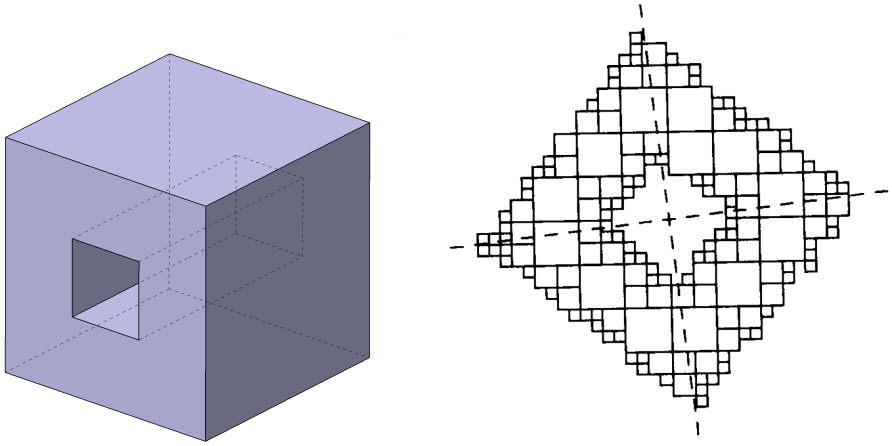
Figure 1.17: Li Ming's model translation: (a) B-Rep model input; (b) point sets model derived from the B-Rep one [36].

mapping functions, $f_{G(i)}$, that map vertices and edges of this subset to corresponding subgraphs $U_G^{(i)}$ that have approximately of the same structure [5] and characterize symmetry properties. The subgraph matching algorithm starts with a random edge and its neighborhood. Within the graph G, it looks for matching a graph under graph structure criteria: edge length, geodesic distances, intrinsic angles and geodesic curvature. Regarding the running time discussion, because the algorithm is based on a sampled point model, the execution time depends on the details of the model that are preserved through the sampling process. Under this possible model adjustment, the amount data for the matching process can be kept low to reduce the time spend. But, as said by the author, the most expensive processes are the feature detection step and sub graph alignment [5]. Within this work, the author enhanced the point set with features as a way to get an intuition of typical results. However, it is not getting significantly more robust than the previous approaches.

As observed through the above analysis, symmetry detection for point set models as input addresses a wide variety of algorithms. Because point set models, even if they are derived from a B-Rep CAD model cannot represent a volume shape accurately, the symmetry detection algorithms proposed can only find approximate results. This is not suitable for the symmetry applications that were mentioned previously in a PDP or engineering contexts.

Minovic et al. detected symmetry properties using an octree representation [42]. The octree decomposition of the object is set up using some intrinsic parameters of the volume input: the reference frame coincides with its center of gravity and the reference

axes are aligned with its principal inertia axes. Each octree cell can have three states: inside of the object, outside the object and intersect the object boundary. Then, the intersecting cell is divided into the next octree level. The symmetry evaluation is obtained by analyzing the mass of octree cells recursively: two cells having the same mass and symmetric inertia axes represent a symmetry property (see Figure 1.18).



(a) (b)

Figure 1.18: The octree method for symmetry detection [42]: (a) a B-Rep type volume used as input, (b) a section of the octree decomposition used to extract symmetry properties of the volume.

In the research of Zabrodsky [70], he introduced a concept of Symmetry Distance (SD) and a symmetry transform operation. This operation transforms the vertices on a 2D figure to a location where the whole figure is symmetric. SD is a quantifier of the minimum 'effort' required to transform a given shape into a symmetric shape. Its general definition enables the evaluation of both distance and rotational angle required to transform an asymmetric figure into a symmetric one. The SD is attached to the center of gravity of the figure. SD is obtained by measuring the moving distance of each vertex and defined as the minimum mean squared distance over all the displacements of these vertices. If the SD is 0, the initial figure is already symmetric (see Figure 1.19). The proposed work was limited to 2D figures or contours even though the SD can be applied to higher dimensions. Also, he noticed that the proposed method is valid for continuous features, but there is a need to discretize the input figure first and then, to use this sampled representation as basis to compute the SD (see Figure 1.20).

Podolak et al. extended the SD measurement algorithm to 3D shapes with a planar reflective symmetry transform. The input model is now 3D meshes. The proposed transform extends previous work on global symmetries with respect to the center

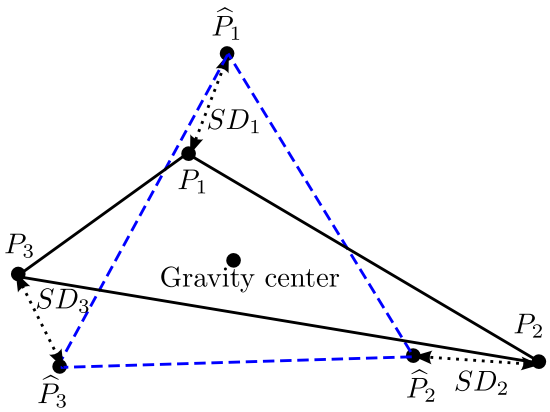


Figure 1.19: Symmetry distance measurement algorithm [70].

of gravity of an object as well as the work of Zabrodsky. Because there is a very large number of planes which could be symmetry planes, he also provides an iterative refinement algorithm that uses Monte Carlo sampling algorithm [51].

Mitra presented an algorithm to detect partial and approximate Euclidean symmetry transforms (reflections, scalings, rigid transformations) on closed triangular meshes[43][44].

His algorithm is based on the following steps:

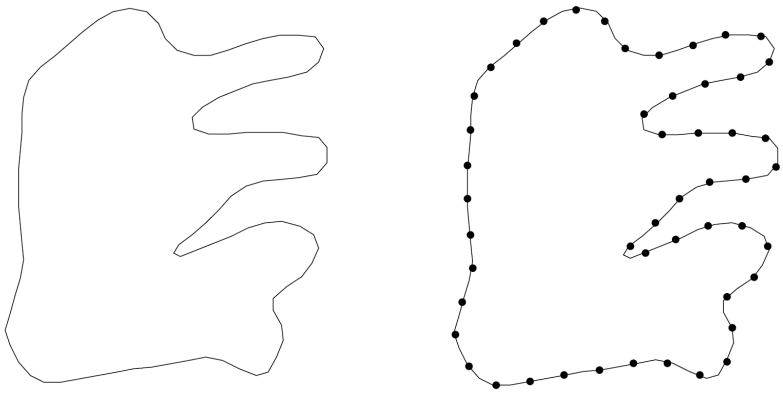
- **Points sampling and curvature analysis:** A curvature tensor is evaluated on a set of points $p_i \in P$ sampled on the triangulated surface S with the algorithm proposed in [2] to approximate the two principal curvatures $\kappa_{i,1}$ and $\kappa_{i,2}$ and principal directions $c_{i,1}$ and $c_{i,2}$;

- **Points pruning:** A restricted point set $P' \subset P$ is generated by removing quasi-umbilic points from the initial vertices, that is points p_i such that $|\kappa_{i,1}/\kappa_{i,2}| < 0.75$. Points pruning not only reduces the complexity of the analysis, but also removes points where the determination of principal directions is uncertain;

- **Pairing:** a signature of each point of P' is generated. This signature contains seven values that are invariant under one or more transformations:

  - The ratio of principal curvatures $\sigma_7(p_i) = \kappa_{i,1}/\kappa_{i,2}$ is invariant under reflections, scaling, and rigid transformations;
  - The principal curvatures are invariants under rigid transformations: $\sigma_6(p_i) = (\kappa_{i,1}, \kappa_{i,2})$.

  The signatures are stored in spatial proximity data structures (*kd-tree*) to identify candidate pairs for reflections, scalings, rigid transformations;



(a) (b)

Figure 1.20: Continuous feature to discrete feature [70]: (a) the initial 2D figure as a continuous contour, (b) the discretized contour used to evaluate the symmetry distance.

- **Clustering:** point pairs computed at the previous step are used to extract symmetries between larger areas, i.e. groups of pairs with a similar transformation that corresponds to symmetric areas of the model surface. The symmetry transforms of each pair, noted Γ_i, are identified as a 7D transformation representing one scale factor, three translation components, and three rotation components. Groups of similar transformations are then identified in the set of transformations Γ such that a group is formed with adjacent pairs in 3D having their transformations that are close in 7D.

The complexity of the points sampling, curvature analysis, and pruning is linear, while the pairing and clustering processes complexity is of $O(n' \log(n'))$ time complexity where $n' = |P'|$.

In the example showed in Figure 1.21, the model has been sampled with 100 points (black spheres) and 500 points (yellow spheres), leading to 280 and 1262 points in Γ, respectively.

The symmetries of large areas of the model are identified robustly, even if the accuracy of the identified transformations is limited for identified point pairs (pairs don't exactly share the identified symmetry).

A strong limitation is that symmetric areas that are smaller than the distance between points cannot be identified. Figure 1.21a shows that every window area is sampled with less than two points, and no self-reflection symmetry is identified in window areas in Figure 1.21d. One could easily imagine that the number of sampled

points would drastically increase to be able to detect symmetries on small features in the example presented above.

Finally, the algorithm could hardly be extended to support axial symmetries, that are viewed as an infinite number of reflective symmetry planes in the approach.

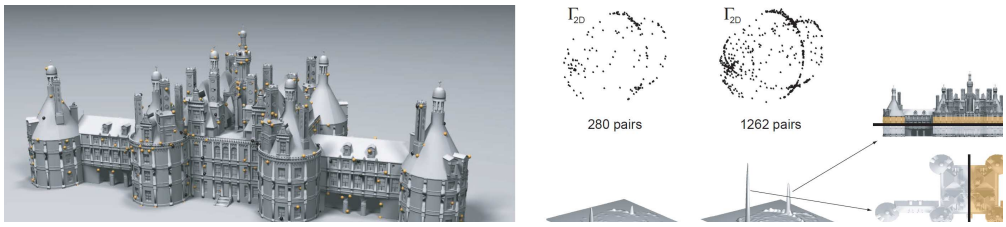


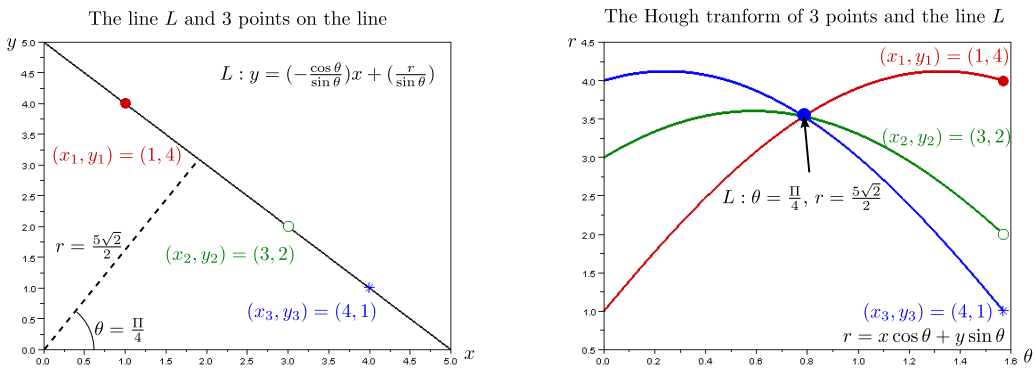
Figure 1.21: Results obtained in [44] for a model of Chambord castle: (a) input model with random surface samples drawn from a total of 2254 samples, (b) Γ_i points in 7D transformation space projected in 2D and associated density plots; the symmetries corresponding to the largest two modes are shown on the right, (c) successive reductions by taking out symmetric patches and resulting bounding box hierarchy, (d) advanced editing using the extracted symmetry relations.

Another similar local reflective symmetry detection algorithm was described which uses Hough transform [10][68][69]. Hough transform is widely used in image analysis and computer vision. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes through a voting procedure. Hough transform is also adapted at detecting simple shapes. For example in 2D space, a line can be described as [17]:

$$y = -(\frac{\cos\theta}{\sin\theta})x + (\frac{r}{\sin\theta}) \tag{1.4}$$

where θ and r are the reference parameters of the line. This line equation can be rearranged as:

$$r = x\cos\theta + y\sin\theta. \tag{1.5}$$



(a) (b)

Figure 1.22: The Hough transform of a line: (a) is a line and points in $x - y$ space; (b) after Hough transform, the same line and points in the $r - \theta$ parametric space.

It means that a point (x, y) in 2D space becomes a curve in the (θ, r) parametric space and a line in 2D space becomes a point in (θ, r) parametric space (see Figure 1.22). When processing range data or point set models, Hough transform can also be used, thus helping process 3D shapes. Similarly, the parameters of a function defining a plane in 3D space contains two parameters $(\vec{n}, d)$. Each pair of points $(\vec{v}, \vec{v}')$ in the point set model can define a candidate symmetry plane as:

$$\vec{n} = \frac{\vec{v}' - \vec{v}}{\|\vec{v}' - \vec{v}\|}, \tag{1.6}$$

$$d = \vec{n} \cdot (\frac{\vec{v} + \vec{v}'}{2}). \tag{1.7}$$

Finally, the planes derived from the remaining candidate pairs of points are inserted in a 3D Hough transform where their evaluations are accumulated into bins regularly distributed over the transform space [11]. Because the preparation phase is similar to that of Mitra, the time complexity of this method is the same as Mitra's, too.

Extended Gaussian Image (EGI) which was reported by Horn [23] was used for 3D meshes symmetry detection in the paper of Sun and Sherrah [59]. EGI is a mapping of surface normals of an object onto the unit sphere (Gaussian sphere). For computer computation purposes, the Gaussian sphere is tessellated. The center of the Gaussian sphere coincides with the center of gravity of the object. Then, a spherical histogram

is generated which maps each facet of the 3D mesh to the unit sphere. Then, the symmetry planes can be calculated using the symmetry properties of the spherical histogram because similar symmetry properties are preserved between the 3D mesh and its EGI. Non convex objects however, cannot be analyzed robustly with this approach. This method can successfully detect symmetries of symmetric shapes but with some special asymmetric shapes, the EGI mapping could be symmetric, which reduces the robustness of the approach. Also, the accuracy of symmetry detection depends on the tessellated sphere and original meshes.

Kazhdan et al. [28][29][30] reported another method to describe the input digital model. The input model is a voxel grid model. A reference center is placed at the center of gravity of the model or at the center of its bounding box. Then, the algorithm uses spherical harmonic functions to represent each voxel of the model and to characterize its symmetry properties on the sphere (see Figure 1.23). By comparing the angle parameter of the shape descriptors, the method can detect the symmetries of the model. With the property of the spherical harmonics function and the help of Fourier Descriptor method, the method also can detect axisymmetry and can be used with a multi-resolution approach.

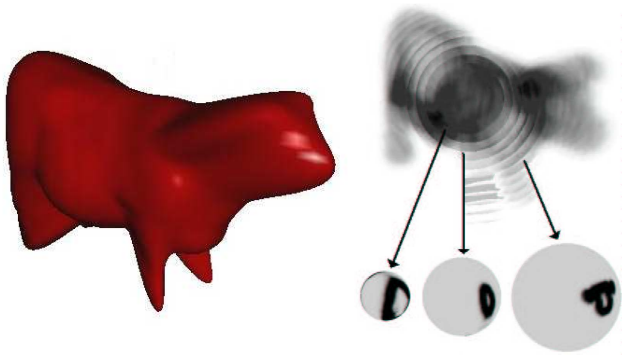


Figure 1.23: The spherical harmonics function description [28][29] with the shape descriptor of the object on the left and the illustration of the range of spheres attached to the object and used to collect its symmetry properties.

Martinet in his report [39][40] used the concept of generalized moments to characterize the symmetries of an object. He found that the symmetry properties of a shape and its generalized moments of order $2p$ in a direction ω behave similarly. Using this property, the symmetry detection is translated into a set of linear equations that has to be solved. In his approach, he also uses spherical harmonics function. The model input must be a continuous and closed shell to be able to compute the generalized moments. It also needs that each surface be discretized so that it can be represented by linear equations. With these constraints, mesh models are a good candidate cat-

egory of input models. Because mesh models are closed B-Rep models, these models can be derived from B-Rep CAD models but each surface must be discretized, which still incorporates approximations whose effect can be difficult to master.

In all these methods, if the mesh model is effectively symmetric, the result of symmetry detection is good too. As the introduction given at section 1.1, mesh models, whether for engineering purposes or in the field of computer graphics, are generated through a transformation of a shape model, often from a B-Rep CAD model. The problem originates from the fact that mesh models are not generated with symmetry preserving properties of their input B-Rep CAD models. As put forward by the explanations at the beginning of this chapter, mesh models are essentially generated for FEA purposes or other analyses in the engineering context. With asymmetric details or even with a symmetric original B-Rep model, mesh model generation does not always create a symmetric mesh model. That is why users need a symmetry preserving property during mesh model generation. This observation applies also to steps of a PDP where symmetry properties of a component or product in important. It has also to be pointed out that approximated symmetry properties can hardly be acceptable from a user point of view because hidden tessellation operations can influence the result, leaving the user with an weak process where he, resp. she, cannot rely on these properties. Such a robustness is a key property in an industrial process.

1.4.2 Symmetry detection applied to B-Rep CAD models

The symmetry detection of B-Rep CAD models has also been addressed by researchers. Early on, Davis used candidate axes to detect symmetry properties of 2D polygons [14]. Considering two adjacent segments, an axis is their angle bisector or, considering one segment, an axis is orthogonal to this segment and located at its middle point. In case of two separated segments, there are four possible axes. The candidate symmetry axes are defined hierarchically and different symmetry axes levels contain different parts of the polygon input. This approach has been also extended to continuous curves using an average middle line so it reduces to a polygonal approach, too. Parui detected symmetry of 2D polygons [49]. The method was considering 2D loops as polygons. The location of segment extremities are the reference of symmetry axes. This work however is too basic to address the shape diversity of engineering components.

Parray-Barwick and Bowyer present a Woodwark's Algorithm to recognize features which supports a multi-dimensional set theoretic context [47][48]. The key point is a template matching process. A partial shape or an existing shape is considered as template and then using its center of gravity, length of contour or other properties, the algorithm recognizes if the target object is the same or not. If a rotational parameter is added, this approach could detect symmetries. Woodwark's algorithm is devoted

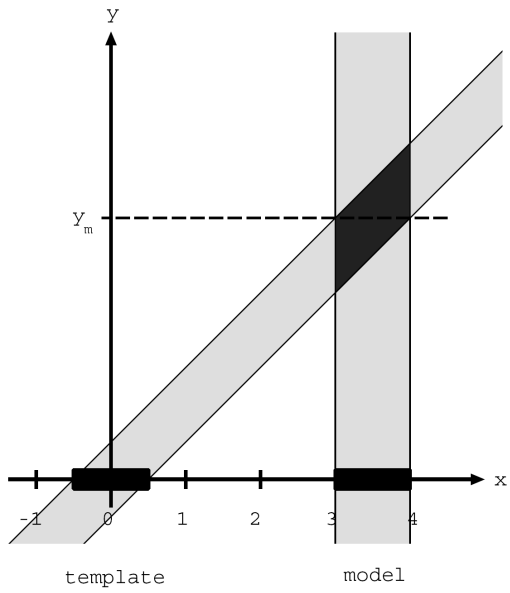


Figure 1.24: A one-dimensional model matching process in (x, y) plane [47][48]. This illustrates the principle of the matching mechanism where a dimension of the template intersects (matches) a range of a parameter of the input model.

to feature recognition when there is a template already available (see Figure 1.24). In a shape analysis context, how to define a template and the matching parameters are among the major issues. The method scans every dimension defining the shape (see Figure 1.24). With an increase in the number of dimensions, the search space for the matching process increases too. Consequently, the matching process will take much more time. Indeed, reducing the extent of the area where could be located a symmetry axis (plane) is a general problem for all symmetry detection methods. As an example, Martin summarized that for a whole object, to find an axis, the object has to be translated so that its centroid is at the origin because the symmetry axes pass through the origin [38].

A different approach was proposed by Kulkarni [32]. The method reduced the 3D shape analyzed to its 2D skeleton with the help of the Medial Axis Transform. The medial axis of a 2D (3D) object is the locus of the centers of all maximal inscribed circles (spheres) [20]. Kulkarni's work addressed 2D objects only. It is a complex task to extent to 3D because the calculation of a skeleton is expensive and it is obtained through a discretization of the object boundary, which may influence the result. In addition, a large disruption in the skeleton can happen when the object boundary changes a little. Therefore, symmetric elements of the object are not necessarily represented in the skeleton [58].

The algorithm reported by Tate is based on a B-Rep CAD model input [58][60]. Here, the boundary surface of the model is limited to five categories: plane, cylinder, cone, sphere, and torus. The symmetry detection process is broken down into five steps:

- Compute loop properties;
- Identify matching loops;
- Construct axes and planes of symmetry;
- Rationalize axes and planes of symmetry;
- Extract the primary symmetry axes from the resulting set.

The loop properties used are: surface type, loop type, loop area, loop centroid, surface normal at the loop centroid and the number of edges in the loop. If all the properties of two loops are identical, the two loops with their underlying surfaces are considered as congruent. The candidate symmetry planes and axes are located from the pair of loops centroids. By comparing the locations of candidate symmetry planes and axes, finally, the primary symmetry planes or axes can be extracted. The computational complexity of this algorithm, in the worst case scenario, is $O(n^4)$, where n is the number of loops of the model. It has also to be pointed out that this algorithm requires rather complex treatments such as computing loop area and loop centroid, which increases the computational effort.

With some model shapes, the method can detect the symmetry properties. But as summarized by the author, this approach faces four limitations. The first one relates to those faces where the intersecting features have some asymmetric details that may be ignored. This problem can originate from the object boundary decomposition because of the influence of the modeling process (see Chapter 4). The second one is that there are configurations where the two properties of two loops are identical but these loops are not congruent, which is illustrated in Figure 1.25. The third one relates to loops subjected to rotations. Rotated loops have the same properties, the corresponding asymmetry created cannot be recognized. The last limitation relates to the nature of features in terms of protrusions and depressions. Also, within this work, the method uses Djinn solid modeling Application Procedural Interface (API) and the ACIS Solid Modeler from Spatial Technology to perform the extraction of loop properties. This environment raises several questions. One is how to monitor the accuracy of computations, which is limited by ACIS. The second one is that it needs to assign the status of internal or external as type to a loop but the loop type definition is ambiguous, hence not robust. The third one relates to the decomposition of surfaces and curves. Not all the surfaces of the B-Rep are maximal. In fact, it is a

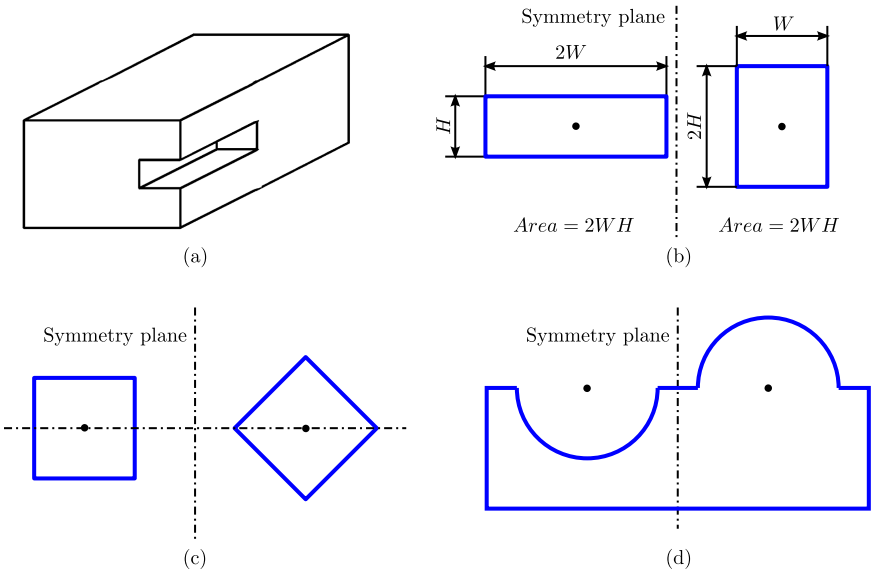


Figure 1.25: Some of the limitations of Tate's algorithm [60]: (a) asymmetry of a detail producing the same global parameters; (b) different loops with same properties; (c) rotation problem; (d) no distinction between protrusion and depression.

general problem, because a B-Rep model always describes surfaces of revolution with two or more pieces, which breaks the object symmetry (see Chapter 4).

Compared to the other approaches, Tate's method is the one that uses effectively a B-Rep model, as an infinite point set model. As discussed in Section 1.1, B-Rep models are used in a design process as basic digital models and widely subjected to shape transformation processes. Therefore, using a B-Rep model that directly matches exactly its boundary, avoids problems of referring to its faceted representation. It is a significant advantage compared to other approaches.

1.5 Conclusion

The symmetry properties of an object have many important applications for different processes in a PDP, which can make the simulations, search, storage, etc. more efficient. Because of this impact, during the development of CAD tools, the research about symmetry detection has always been active. Many algorithms are reported for enumerative or discrete models, such as point set models, voxel models or mesh models. But because their references are discrete models, the result of symmetry detection is only approximate. Improving the accuracy and tracking more details, increases significantly the computational complexity of the symmetry detection. It is no longer

applicable in the engineering context.

In a PDP context, B-Rep CAD models are more useful and more important than the others but they are infinite point set representation and the main group of researches uses a discretized version of these models as input. This does not provide a better answer to the engineering needs. Tate's work is an entirely new approach that directly takes a B-Rep CAD model as input. With the description of loop properties, the candidate symmetry informations are created but this method has many limitations that restrict the results' validity to special cases. This is inherently due to the use of global parameters attached to areas of the object boundary that cannot reflect precisely their spatial configuration. Finally, after many years of research, symmetry detection on B-Rep CAD models is still an issue for its use in a PDP and other engineering applications.

Chapter 2

Principle of the symmetry analysis approach and hypotheses

This chapter gives a overview of the symmetry analysis approach and sets the objectives of the present work. The symmetry analysis is devoted to engineering applications with CAD based models to process shapes close to real objects. The proposed approach is a semi global one since an object boundary is decomposed into patches and curves defining the boundary of these patches. The objects are described in STEP format, hence they are decomposed into patches and their symmetry properties are not readily available. Surfaces and curves bounding objects stands for infinite point sets and they form the basis of the approach. In this case, maximal surfaces and curves generation is mandatory before the symmetry analysis and performed with the help of hypergraphs, which form the main datastructure of the algorithm. Then, candidate symmetry properties are extracted from each entity of the hypergraphs to initiate a divide phase of a divide and conquer process. The conquer phase consists in propagation processes producing global as well as local symmetry information about the object analyzed.

2.1 Principles of the symmetry analysis approach

First of all, it has to be recalled that reflexive symmetry is, by definition, a point based property (see eq. 1.1 and eq. 1.2), which has been exploited essentially for discrete (point sets) and piecewise linear (mesh based or faceted models) representations of objects. Consequently, the algorithms currently available have a rather high complexity (polynomial of high degree) and have to be applied to a large amount of entities due to the discrete representation of the objects processed. In addition, many of them produce an approximate symmetry information because of these discrete representations. Effectively, these representations cannot represent all the details of a physical object like engineering components. Axisymmetry cannot be extracted from discrete representations and the location of symmetry planes stays approximative as a function of a shape discretization, which is often not acceptable for engineering applications. Indeed, engineering applications require symmetry properties evaluation at the level of accuracy of manufactured components to be useful in a PDP. It is the purpose of the proposed approach to address this issue.

Consequently, the purpose is to set up an approach of higher level such that infinite point sets rather than finite ones can be processed, allowing for a precise description of objects while enabling a much faster processing. In that sense the proposed approach is a semi global approach since an object boundary is decomposed into patches and curves defining the boundary of these patches. There are two categories of infinite point sets: curves defining the boundary of patches and patches defining a subset of the object boundary. The goal of the approach is to analyze an object symmetry without limitation of discrete representations: the resolution of point sets and the chordal deviation between a mesh and the precise model of this object. As a result, the input model can contain all the details necessary to get very close to real objects if needed. Regarding the applications of symmetry properties in a PDP, the proposed approach forms a basis to add symmetry information to several of its steps, e.g. restructuring a modeling tree to incorporate the object symmetries since a design process rarely expresses all the object symmetries; exploiting symmetry properties to simplify an object for finite element simulations, to structure an assembly process, to improve the trajectory planning of manufacturing processes, to compare objects and characterize similarities in databases, etc. Whatever, the step considered, symmetry properties being intrinsic to an object, they must be independent from the boundary decomposition of this object. Indeed, this decomposition is the result of modeling and modification processes in a PDP whereas it must be intrinsic to the object symmetries. It is part of the principle of the proposed approach to set up an intrinsic framework to analyze and exploit the symmetry properties of an object.

Tate's approach [58][60], is among the closest to the proposed one. The input model is of type B-Rep and it can contain a combination of planes, cylinders, cones, spheres and tori with possible spline surfaces. This is an advantage but the symmetry detection does not rely on an intrinsic decomposition, hence ambiguous parameters and heuristics have been used and reduce the efficiency of this approach. Here, the purpose is to preserve the intrinsic framework throughout the symmetry analysis process to obtain a reliable and robust process. According to the contribution of different models into a PDP as addressed at chapter 1, the B-Rep NURBS is used by many famous CAD software to generate digital models often regarded as reference ones by companies. B-Rep NURBS models can contain many shape details as defined by engineers and technicians and can be combined with CSG operators.

However, patch boundaries of these models are often resulting from intersection computations, hence they are approximated and their symmetry properties can be perturbed and difficult to obtain robustly like faceted representations are approximations of smooth objects through chordal deviation. Here, the approach aims at favoring the use of intrinsic parameters all through the symmetry analysis. To this end, the intrinsic parameters of surfaces should be used as much as possible to derive symmetry properties and addressing the geometry of intersection curves is avoided to preserve the robustness of the analysis.

If the proposed approach could be integrated in industrial CAD platforms directly or in tight connection with them, this would be very helpful for the design and product development processes. Modification suggestions could be easily exploited in the original CAD software because it would be connected to the modeling tree of objects. Because of the commercial protection, these modeling tree and B-Rep datastructures are internal, hence difficult to access and hardly generic. Even if some of these datastructures are partly accessible, it holds in one software environment and can be transposed to another one only with a fair amount of software development effort.

Another possible integration focuses on shape transformations often taking place between product views in a PDP. It can be associated with all CAD softwares and all simulation platforms when B-Rep NURBS datastructures rely on standard formats. In this context, object shapes can be generated from an original industrial CAD software and may contain numerous details, which justify modification requirements, e.g. simplifications for finite element analyses, for digital review visualization, etc. In the present approach, the STEP format [26][71] has been selected because it is an ISO standard where an object boundary is described with surfaces and curves forming infinite sets of points and analytic surfaces can be explicitly available too, i.e. planes, cylinders, cones, etc. The topology of volume boundaries is explicit and can be used to robustly extend symmetry properties attached to patches and their boundaries. As far as symmetry analysis is concerned, both categories of integration are part of the current approach since symmetry is an intrinsic shape property, which relies on datastructures and processes rather independent from CAD modeling issues. Both categories of integration will be observed here.

Analytic surfaces are widespread in mechanical engineering applications, which entails the description of a wide range of components while benefiting well known symmetry properties to initiate a high level approach for symmetry analysis. Consequently, the main idea of the approach is to analyze object symmetries from the level of patches, seen as infinite point sets, and to extend progressively these point sets to neighboring ones. This can be described as a divide-and-conquer process:

- Every surface patch is described by its intrinsic parameters and its location in space. This is independent of their parametric or implicit equations that could be used and it concentrates on the embedding of each patch in $I\!R^3$ where the symmetry properties take place;
- Every surface patch has self symmetry properties that can be combined with the symmetry constraints of its adjacent surfaces to take into account its boundary as patch boundary of a B-Rep model: this produces Candidate Symmetry Planes (CSPs). It is the result of a division phase;
- Then, a propagation process can extend the validity of each CSP over the largest

possible area of a model boundary to structure the global symmetry properties of the object: it is the conquer phase.

Apart from symmetry analysis, asymmetry is also very interesting and is more useful for design and PDPs since many components are not globally symmetric, rather symmetry exists only at the level of a boundary subset and the loss of symmetry can be used as a means to evaluate shape transformations that could be useful for simplification purposes. The symmetry analysis proposed in the current approach intend to address this issue since the propagation process helps identify and structure areas where symmetry properties are valid.

Finally, the objectives of the approach can be summarized as the algorithms to answer the following questions:

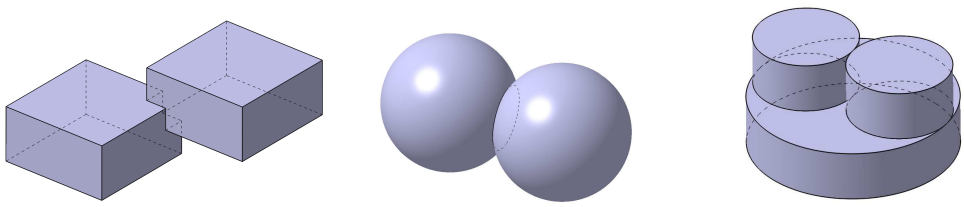
1. Is a B-Rep NURBS model symmetric with respect to some symmetry planes or symmetry axes? If so, where are located these symmetry planes and symmetry axes?
2. If a B-Rep NURBS model has no global symmetry property, doest it benefit local symmetry planes or axes? Where are located these symmetry planes or axes and what are their extents of validity over the model boundary?
3. How does these symmetry properties can be obtained at various steps of the design and PDPs? Into which extent they can be obtained with a process intrinsic to the object shape and how robust is this process under a wide diversity of shapes?

2.2 Hypotheses and shape range

The previous objectives and context fits into a PDP, which is a very complex in terms shapes and models available. The purpose of this section is to define accurately the ranges of models and PDP configurations addressed. First of all, a PDP may rely on a wide range of geometric shapes that fit into several geometric and topological categories such as manifold or non-manifold models, volume model, etc. The hypothesis addresses this categories of models.

2.2.1 Shape categories

The proposed approach of symmetry analysis focuses on volume models. In $\mathbb{R}^3$ space, a volume is a finite and connected subspace. A volume divides the 3D space into two partitions: one is open and outside of the volume and the other one is bounded and defines the interior and the boundary of the volume (see Figure 2.1). It means that any



(a) (b) (c)

Figure 2.1: Illustrations of example objects part of the volume category addressed by the current approach.

point located strictly inside the volume, i.e. not on its boundary, has a neighborhood defined as a ball. Then, any point located on the boundary of the volume has a neighborhood that is topologically equivalent to a disk. The corresponding class of objects is designated as 2-manifolds. The Euler-Poincaré theorem is applicable to them and extends the notion of volume under the form:

$$\#V - \#E + \#F = 2(\#s - \#h), \tag{2.1}$$

where the quantities are respectively, the numbers of vertices, edges and faces forming the boundary of the object and the numbers of partitions and holes of this boundary. In particular, $\#s$ extends the concept to objects dividing the 3D space into more than two partitions. Industrial CAD modelers behave differently regarding this extension, some allowing a subset with 'cavities' only, others conform exactly to this general concept.

Here, as a first step of the proposed approach, volumes have been restricted to 'manufactured' objects only, i.e. when they are bounded by one partition only.

If there exist point neighborhoods of the object boundary that are topologically equivalent to several disks or to a disk and segments (see Figure 2.2), the corresponding class of objects is of type non-manifold. This class of objects is often appearing in geometric models devoted to simulations like finite element ones. Here, they are out of the scope of the proposed approach. However, non-manifold configurations have to be distinguished from that of Figure 2.1c where the two small cylinders are geometrically tangent to each other but not connected through an edge of the object boundary. This configuration is indeed manifold and within the scope of the proposed approach. In Figure 2.1c, the two top cylinders are tangent. Between them, if at their common tangent linear edge and vertex exist, they must be two instances of the same group of edge and vertices superimposed and forming an open chain, otherwise the model is non-manifold.

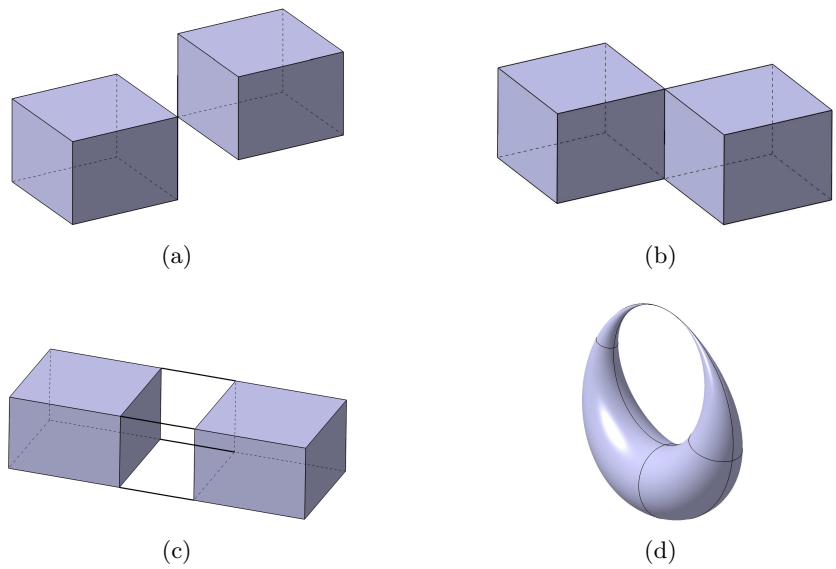


Figure 2.2: Illustrations of object examples falling outside the scope.

2.2.2 Range of objects addressed

As stated in the previous section, the shape category addressed is of type volume and follows the restrictions stated there. Industrial CAD modelers however, can produce different object categories, sometime under the same computer type they designate as volume, which is not accurate enough for the purpose of the proposed approach. Here, the goal is to define the object range which is not part of the previously strictly defined volume category but that can be transformed into this category without perturbing the symmetry analysis.

As a refinement of the non-manifold category illustrated before, let us consider configurations where non-manifold singularities occur at vertices only and are further restricted to configurations where the neighborhood of these points are formed by sets of disks. Indeed, such a range of configurations is named pseudo-manifolds and is able to describe volumes as stated previously even if they incorporate these non-manifold singularities. Pseudo-manifold can appear as 'extreme' configurations in CAD software and STEP files. For the sake of completeness, these configurations can be converted into manifold ones through vertex duplication and fall in the scope of the proposed approach without changing the object shape, hence without influence on the symmetry properties of the object.

The analysis can be extended to non-manifold configurations where the singular entity is an edge. Figure 2.1c can be interpreted in that way if a unique edge exists as common generatrix of the two cylinders. Consequently, if the duplication of these

edges and some of their extreme vertices modify the boundary decomposition of the object without modifying the number of partitions, i.e. $\#s = 1$, these transformed objects have preserved their initial geometry and now fall into the object category addressed here.

2.2.3 Shape geometry and reference surfaces

The previous sections have concentrated on the topological aspects of the shape range addressed. Here, the purpose is to focus on the geometry of the models covered in the proposed approach. The main context of the present work concentrates on PDPs and mechanical engineering applications. There, objects are generally created through constructive approaches using sketches and simple primitives. Line segments and arc of circles are combined to form most of the sketches content and they are translated or rotated to be extruded or revolved to form primitive volumes that can added to or removed from an existing one. Processing geometry that way forms already a wide range of manufactured objects whose boundary surfaces are derived from the previous observations. Such faces are now designated as *Reference faces*.

In the framework of CAD software, 2D sketches in arbitrary planes as basis for extrusion or revolution operators form the main method to create 3D volumes. Blends and chamfers come afterwards as local modification of an object boundary. As a result, the boundary surfaces of 3D volumes fall into the following configurations when combining segments and arcs with extrusion (translation) and revolution (rotation):

- **Planes**: they can originate from the extrusion of a segment belonging to a sketch, from the revolution of a segment orthogonal to the rotation axis or from a closed planar contour of a sketch defining a face of a volume primitive;
- **Cylinders**: they can be obtained by revolution of a segment parallel to the rotation axis or as extrusion of a circular arc orthogonally to the plane containing the arc;
- **Cones**: they are generated by rotating a segment that is not parallel to the rotation axis;
- **Spheres**: they are obtained by revolving a circular arc whose center is located on the revolution axis;
- **Tori**: they originate from revolving a circular arc whose center does not lie on the revolution axis.

As a result, assuming that sketches only contain segments and circular arcs covers a large amount of mechanical objects. Hence, the five surface types listed above

form the basic configurations of 3D volume boundary surfaces. These five surfaces are called reference surfaces and noted S. Because it covers a large diversity of CAD volumes, it is assumed that 3D volume boundaries considered here combine only these reference surfaces. There is no restriction placed on intersection curves between these surfaces. Consequently, configurations on blending radii with constant or variable radius located on arbitrary intersection curve between reference surfaces can generate free-form surfaces and are not part of the present approach.

This restriction is a trade-off between a wide enough range of objects covered and the complexity of the approach. Perspectives still exist to widen further the current object range.

2.3 Maximal Surfaces and Maximal Edges Model

Having specified in section 2.2 the range of input shapes covered by the proposed approach, its principles have to be set in adequacy with the input model. First of all, it has to be recalled that the symmetry properties of an object are intrinsic to it whereas the B-Rep decomposition is subjected to constraints originated from the:

- Topological properties needed to describe the object, e.g. an edge must have exactly two extreme vertices;
- Modeling kernel where the boundary surfaces need to be processed in a specific way to meet topological requirements, e.g. a cylindrical surface must be decomposed into two half cylinders at least to correctly embed a topological space describing the object boundary (see Figure 2.3b);
- Modeling process as set up by a designer at each stage of a PDP, e.g. the content of sketches and their influence on the surface generation may generate unnecessary edges and faces, the successive modifications of designs, the transformations of models between CAD and simulation software can modify the boundary decomposition of an object without changing its 3D shape.

The purpose of the approach being to derive global symmetry properties from that of infinite point sets, surfaces, curves and points are the entities forming the input model and the categories of point sets processed. The corresponding faces, edges and vertices will then be the support of the divide and conquer process to analyze the object symmetries. As a result of their embedding in 3D, reference surfaces can be bounded (sphere, torus) or unbounded (plane, cylinder, cone). The purpose of curves is to restrict or bound these surfaces. Similarly to surfaces, curves can be bounded (circle, ellipse, etc.) or unbounded (straight line, parabola, etc.). Similarly, the purpose of vertices is to restrict or bound these curves.

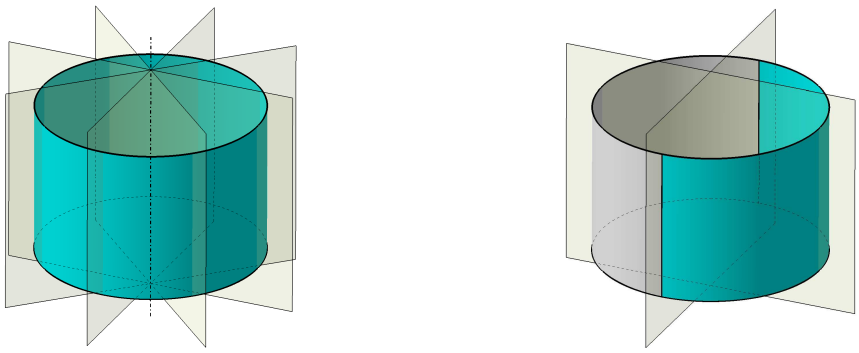
As a result, the infinite point sets must be as large as possible not to interfere with the symmetry properties of the object. This leads to the concept of maximal faces and maximal edges that form the basis of the divide phase when CSPs are attached to them as starting point of the conquer phase. More precisely, the concepts of maximal faces and edges can be explained as follows.

Maximal surfaces: During a modeling process, several surfaces S of a B-Rep model, adjacent to each other through a common edge, can share the same surface type and parameters (same axis and radius or same normal and reference point) because of the successive sketches defined by the user and the type of operator applied. In addition, if the shape boundary contains circular areas such as a full cylinder or a full torus, all CAD software either divide these surfaces into more than one piece or open their boundary to meet the topological requirements of 2-manifold surface models.

Because symmetry properties are global for a geometric domain, there are no longer identical for an entire, non decomposed, surface and the same surface subdivided into a set of adjacent domains. This is illustrated in Figure 2.3 where the cylinder halved has two symmetry planes passing through the axis (Figure 2.3b) whereas the full cylinder has an infinite number of symmetry planes (see Figure 2.3a). In order to obtain the right symmetry information, faces must be of maximum area even though they can no longer satisfy the B-Rep topology description, i.e. an object formed by a sphere can be described with a unique face and no other edge or vertex. Consequently, surfaces may be merged through edges or vertices to meet the area maximization and two adjacent maximal surfaces S_i and S_j have to have different characteristics (type, location, axis location, radius, normal, etc.).

Maximal curves: Similarly to maximal surfaces S, each boundary curve Γ, i.e. a loop, of each surface S needs to be of maximal length so that the symmetry properties of Γ can be analyzed robustly and don't propagate errors when analyzing S. Γ decomposes into a set of maximal curves Σ_k containing at least one curve. Every curve Σ_k is an intersection curve between two adjacent maximal faces (S_i, S_j) . A merging algorithm is used to generate Σ_k that takes into account the characteristics of the two faces S_i and S_j because intersections curves are explicitly used. Because of the five type of surfaces, maximal curves Γ are the loci of G^1 or G^2 discontinuities. Similarly to surfaces, maximal curves can be closed, thus having no vertex because they are entirely defined by their two adjacent maximal surfaces.

The above explanations about maximal surfaces and curves show that these concepts are effectively intrinsic to the object shape and get red of the constraints set by the modeling kernel and modeling process. They show also that the requirements of the symmetry analysis about the description of the piecewise boundary of the object are incompatible with the intrinsic constraints related to topology, i.e. faces are not necessarily bounded by edges and edges are not necessarily bounded by vertices.



(a) (b)

Figure 2.3: A simple illustration of an object boundary decomposition on its symmetry properties: (a) a full cylinder having an infinite number of symmetry planes, (b) two half cylinders which have only one symmetry plane passing through the axis.

2.4 Preliminary processes to the effective symmetry analysis

As mentioned above, there are some treatments required prior to obtain the volume used as input of the divide and conquer process to analyze its symmetries. The previous section has shown that the maximal surfaces and curves are key concepts to generate infinite point sets whose symmetry properties coincide effectively with that of the object. Therefore, there is a one to one mapping between these point sets and the maximal surfaces and edges. These point sets are now designated as faces, edges and vertices and the adjacencies between surfaces, curves and points can be expressed with a combinatorial approach.

Hypergraph datastructure is efficient at representing a wide range of adjacency relationships between entities an it is selected for supporting the symmetry analysis. Hypergraph datastructures provide a well suited representation of adjacency relationships between faces, edges and vertices where point sets symmetry properties can be preserved while staying connected to the topological datastructures of the initial object. Here, emphasis is put on the hypergraph transformations needed to generate the object boundary decomposition that preserves the object symmetry properties. Hypergraphs can be transformed with a large set of operators enabling face merging, edge merging, vertex deletion, etc. So, the first step consists in transforming the B-Rep model topology into hypergraph data structures. Then, there should be a process for non-manifold configuration detection and non-manifold edge and vertices splitting to produce a volume derived from the class of objects defined in the hypotheses. Finally, a process will merge the adjacent surfaces and curves to satisfy the maximal surface

and maximal curve requirements. Also, the closed boundary or loop structure of a face is needed since it can reduce the number of CSPs and it contributes to trace the propagation during the conquer phase. A loop can be considered as a closed chain of face boundary edges, which is edge adjacency information applicable to simple loops where no orientation is needed. So, loops can be extracted from hypergraphs. Special configurations will be addressed in details in chapter 4.

2.5 An overview of the symmetry analysis process

The major steps of the symmetry analysis process can be synthesized from the description of the previous principles, hypotheses and preliminary processes and they are structured as follows (see Figure 2.4).

The overall process starts from a STEP file input and translates the topology of the input object into hypergraphs (see Chapter 4). Next step is the detection of non-manifold edges and vertices and then, splitting them to build a manifold model to the range of shapes compatible with the current scope of the analysis process. To conform to symmetry properties of infinite point sets, i.e. faces, edges and vertices, and configure the piecewise decomposition of the object boundary, all adjacent surfaces of same type and same spacial location define the same point set and must be merged when the adjacency operates through edges. Adjacency through vertices requires a specific analysis of neighborhood. A similar process must be applied to the face boundaries, i.e. the intersection curves between adjacent surfaces. Whether for surfaces or curves, the corresponding point sets obtained satisfy the criterion of maximal point sets.

Having produced the boundary decomposition containing the maximal point sets, the next step takes advantage of global symmetry properties of the maximal point sets to generate CSPs attached to entities of the hypergraphs (see Chapter 5). Relying on global symmetry properties, CSPs can be structured into several categories whose meaning relates them to each dimension of infinite point set, i.e. dimension two for faces, one for edges and zero for vertices. Because symmetry properties strongly rely on the embedding of a surface in 3D space, there is a combinatorial issue to collect all the CSPs among all the interactions between the five reference surfaces. All these entities help defining CSPs sharing adjacency relationships with their neighborhood. A last category of CSPs relate to loops or, otherwise stated, disconnected point sets. The loop datastructure derived from the hypergraphs help defining the corresponding CSPs. As a result, CSPs can be assigned to entities of the object boundary: it is the divide phase of the analysis process.

Finally, following the adjacency relationships in the hypergraphs and loop structures, the conquer phase develops propagation processes (see Chapter 6). CSPs propagation processes aim at extending point sets with valid symmetry properties to cover

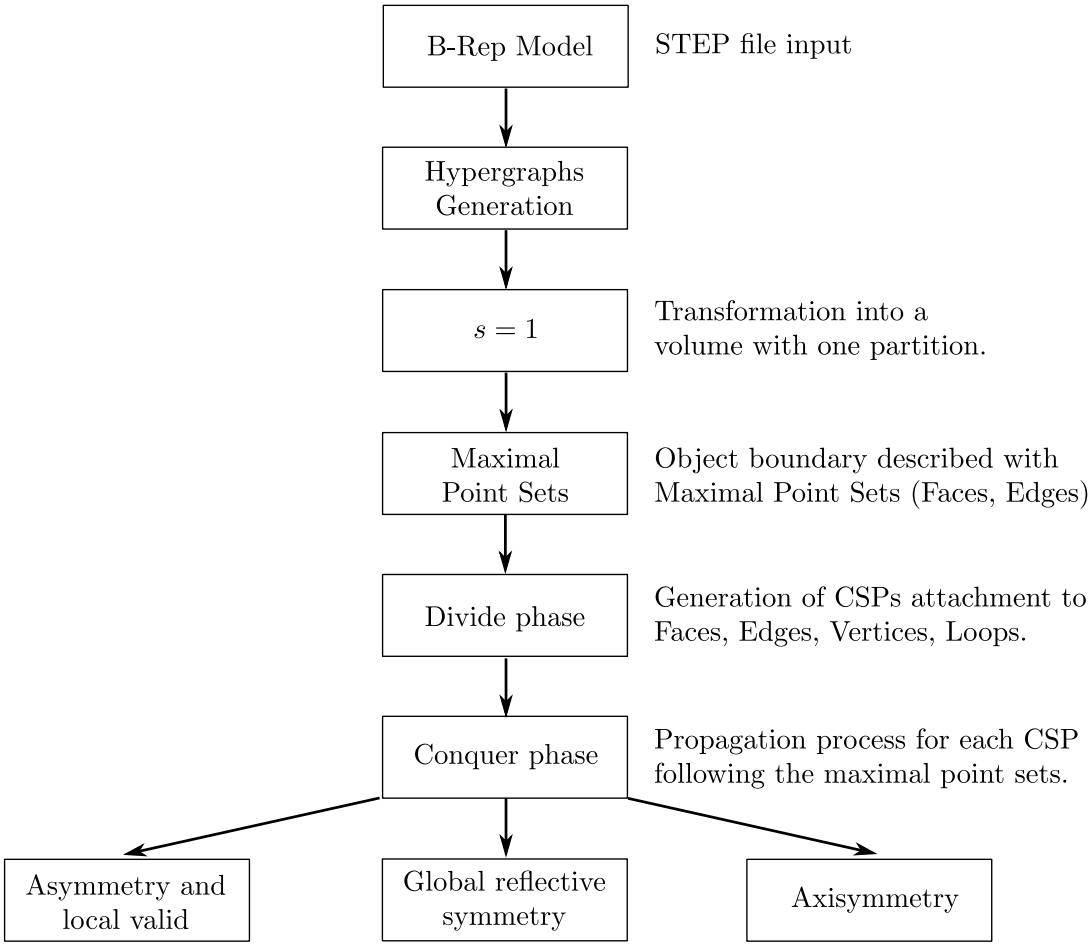


Figure 2.4: The major steps of the symmetry analysis algorithm.

the largest possible area of the object boundary. Propagation takes place at two levels:

- Extend to point sets that intersect with the CSP;
- Extend to point sets having no intersection with the CSP.

Whether the whole object surface is valid or not for a CSP dictates its status: global symmetry plane in the first case and local one for the second. The extent of the valid area for a CSP relates it to the corresponding subset of the object boundary. These results produce the answers to the targeted objectives mentioned at section 2.1.

2.6 Conclusion

This chapter has described the principle of the symmetry analysis process, which is of type divide-and-conquer. Hypotheses have been set up to define its scope in terms of shape category addressed to cover a first realistic set of shapes available in engineering applications throughout a PDP. This demonstrates the interest of the proposed approach and highlights some efficient features of the proposed approach. Except the hypotheses, some preparation phase is mandatory and must be added to the major steps of the divide-and-conquer process: transformations of non-manifold singularities, when possible, to meet the category of volumes processed and, most important, generate the maximal point sets forming the object boundary so that the divide-and-conquer process can be effectively meaningful. It is now the purpose of the two next chapters to detail this preparation phase.

Chapter 3

Basic Input Data: B-Rep CAD model and STEP format

Following the context described in the previous chapter, this chapter focuses on the analysis of the input model for the symmetry analysis process. Here, the input model of type B-Rep is described in the STEP format considered as a reference datastructure that can be found either in a modeling kernel of an industrial CAD software or as a description of an object transfered from an application software to the symmetry analysis. Then, the analysis of the STEP format helps justifying the use of surface parameters only and defining the range of models addressed compared to the models stored in STEP format by geometric modelers.

3.1 Introduction

The range of shapes used as input for the analyzed models is of type volume, i.e. a 2-manifold surface bounding the object. The input model is described through the STEP format considered as a reference format available from any industrial modeler and is able to form a reference model for a modeling kernel. A STEP file is an ISO standard format for CAD. It describes objects as B-Rep models. It is suited for neutral file exchange, data sharing and archiving. Because it is among the most widely used data format in PDPs, our work about symmetry analysis focuses on STEP files as input data.

3.2 Overview of the STEP format for volume description

Commonly, a B-Rep model is described through geometric and topological entities. Geometric ones are listed as: surfaces, curves bounding the surfaces, points bounding the curves (see Figure 3.1 for a simple example). Directly linked to the geometric entities are associated topological ones: faces, edges bounding the faces and vertices bounding the edges. The following presentation and analysis concentrates on the topological entities since they are the major ones for the symmetry analysis purpose

described throughout the next chapters. Indeed, geometric entities are kept unchanged throughout the symmetry analysis process.

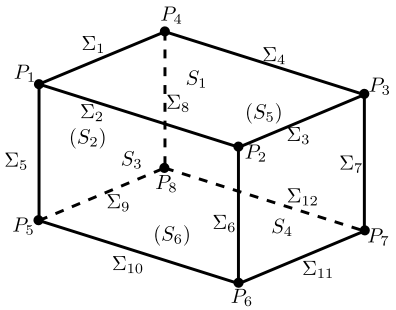


Figure 3.1: A simple B-Rep model with its associated geometric entities: S_i (surfaces), Σ_j (curves), P_k (points).

In the STEP data structure, each topological entity belongs to an independent class where its geometric data is stored in an associated geometric entity. Topological entities map each other in a hierarchical manner where, for each dimension, the highest dimensional entity describes how it is bounded by lower ones (see Figure 3.2a, b). These relations are expressed as a table structure (see Figure 3.2a, b for the object displayed at Figure 3.1) between faces and edges and edges and vertices. An example of association between topological and geometric entities is also given at Figure 3.2c.

face	edge
F_1	E_1 E_2 E_3 E_4
F_2	E_1 E_5 E_9 E_8
F_3	E_2 E_5 E_{10} E_6
F_4	E_3 E_6 E_{11} E_7
F_5	E_4 E_8 E_{12} E_7
F_6	E_9 E_{10} E_{11} E_{12}

(a)

edge	vertex
E_1	V_1 V_4
E_2	V_1 V_2
E_3	V_2 V_3
E_4	V_3 V_4
E_5	V_1 V_5
E_6	V_2 V_6
E_7	V_3 V_7
E_8	V_4 V_8
E_9	V_5 V_8
E_{10}	V_5 V_6
E_{11}	V_6 V_7
E_{12}	V_7 V_8

(b)

vertex	coordinate
V_1	(x_1, y_1, z_1)
V_2	(x_2, y_2, z_2)
V_3	(x_3, y_3, z_3)
V_4	(x_4, y_4, z_4)
V_5	(x_5, y_5, z_5)
V_6	(x_6, y_6, z_6)
V_7	(x_7, y_7, z_7)
V_8	(x_8, y_8, z_8)

(c)

Figure 3.2: Topological entities of a B-Rep data structure: F_i (faces), E_j (edges), V_k (vertices) and their 'bounded by' relations (a), (b) as tables. (c): an example of association between topological and geometrical data.

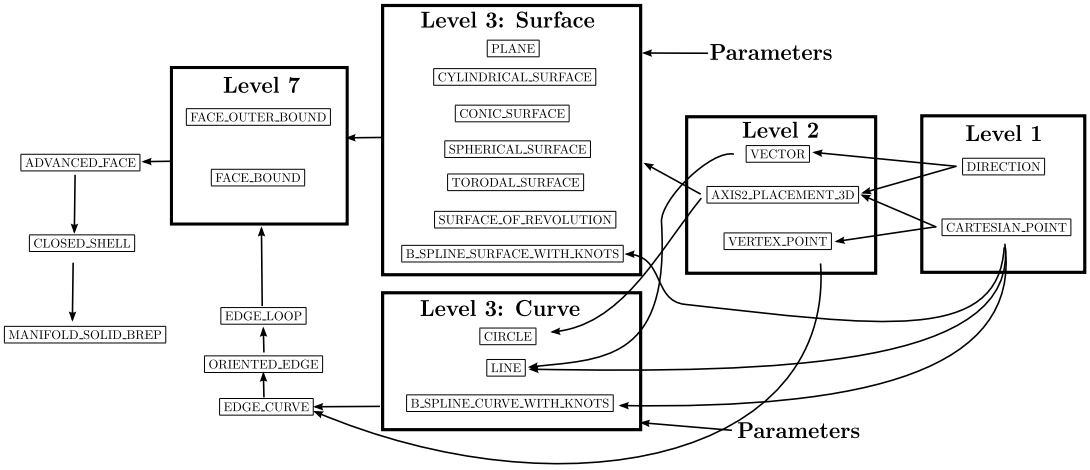


Figure 3.3: STEP format elements.

3.2.1 Structure of the major entities contributing to the definition of a B-Rep volume entity

The STEP format defines a rather sophisticated B-Rep datastructure. The entities are shown in Table 3.1, which is decomposed into a hierarchy of 12 levels. The hierarchical connections between the entities are described in Figure 3.3.

In a STEP file, the basic entities are called CARTESIAN_POINT and DIRECTION, which are part of level 1. A CARTESIAN_POINT is a set of point coordinates, such as a reference point of an axis, the coordinates of a vertex, a control point, etc. A DIRECTION is a vector with a unit length. These two basic elements are combined to form the level 2 entities of the hierarchy, e.g. an axis system as AXIS2_PLACEMENT_3D, a vertex as VERTEX_POINT that reduces to a pointer to a CARTESIAN_POINT and a vector as VECTOR. Then, curves and surfaces are described with level 1 and level 2 entities and some more parameters such as a radius, e.g. when a curve reduces to a simple curve like a circle. They are stored at level 3, which forms the set of geometric elements.

These elements are generically defined and only contain intrinsic parameters of the corresponding surface or curve. There is no boundary information unless it is mandatory to include it in the definition of a geometric element like a parametric curve. For example, a LINE is defined from a CARTESIAN_POINT plus a VECTOR and it is infinite (see Figure 3.4b L_{60}, L_{76}, L_{155} and L_{162}). A CYLINDRICAL_SURFACE is described as an infinite area too (see Figure 3.4c). Adding a vertex as boundary constraint to curves to describe them as they are in M_I (the input model), defines the entity EDGE_CURVE, which is stored at level 4. This is a first entity of type topological in the series needed to describe the topology of M_I boundary. At level 5, the ORIENTED_EDGE is created from an EDGE_CURVE. The ORIENTED_EDGE

is part of a half edge data structure (see Figure 3.4e).

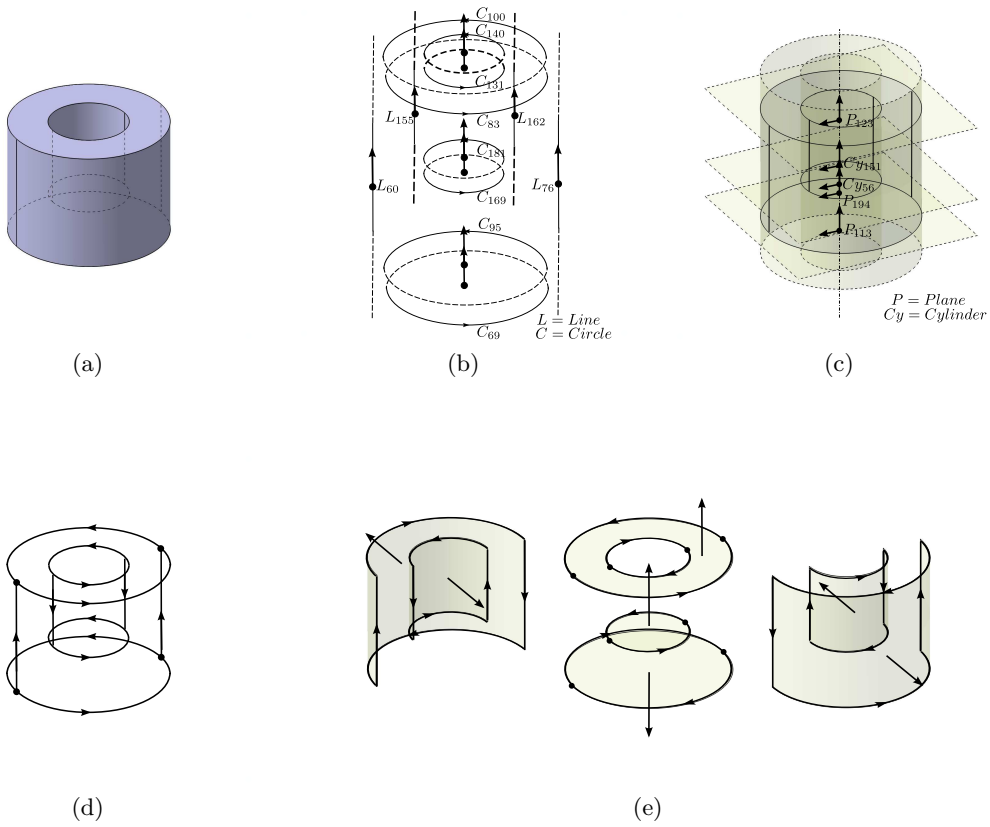


Figure 3.4: An example of STEP file: (a) is a volume; (b) are curve elements with orientation. There are 4 LINEs and 8 CIRCLES; (c) are surface elements, 2 CYLINDRICAL_SURFACEs and 3 PLANEs; (d) are EDGE_CURVEs; (e) shows the ORIENTED_EDGEs, EDGE_LOOPs, FACE_OUTER_BOUND, FACE_BOUND and ADVANCED_FACE with orientation.

The half edge datastructure is dedicated to the description of ∂M_I and its corresponding face adjacencies. Because in a manifold model, every edge is adjacent to two surfaces exactly, a half edge is constructed by splitting an edge down to a pair of half edges. This pair of half edges coincides with the original edge and they have opposite orientation to stay consistent with their adjacent faces. The half edge datastructure has good properties for describing the adjacencies in M_I and its orientation. The STEP standard is based on a half edge datastructure. In next level of STEP structure, ORIENTED_EDGEs construct an 'EDGE_LOOP'. An EDGE_LOOP is the boundary of one surface defining one face of M_I. Similar to the definition of a volume having one partition, a surface is connected: each internal point of the surface is connected with

every neighboring point in this surface. A surface is at least G^2 continuous excepted as singular points like cone apices since the surfaces addressed here are analytical ones. Because surfaces in STEP are addressed as parametric functions, a surface has only one outer boundary and can have an arbitrary number of internal boundaries. So the next level in STEP format is 'FACE_OUTER_BOUND' and 'FACE_BOUND' (face internal boundary). With the face boundary and the surface geometry defined at level 2, a surface with arbitrary boundaries can be described explicitly both from geometric and topological points of view and it is named 'ADVANCED_FACE' at level 8. Level 9 describes 'CLOSED_SHELL' as a combination of several ADVANCED_FACEs. Then, level 10 up defines the 'MANIFOLD_SOLID_BREP' from a closed shell to add naming attributes. The analysis of a volume with a single partition in STEP format ends here, which fits with the hypotheses of the current work. If the model contains more than one volume, they are stored in 'ADVANCED_BREP_SHAPE_ REPRESENTATION' at level 11.

3.2.2 Some insight about orientation information

The STEP format provides information about oriented reference entities. At first, geometric elements are oriented. The VECTOR entity in the LINE one defines the orientation of the latter. With the CIRCLE entity, its orientation is set with the right hand rule. The thumb of the right hand is defined with the DIRECTION contained in AXIS2_PLACEMENT_3D contributing to the definition of the CIRCLE. Then, curving the other fingers points at the circle orientation. At level 4, the EDGE_CURVE entity is assigned a start and an end vertices. If start to end vertices follow the orientation of the corresponding geometric curve, the EDGE_CURVE orientation is set to 'T' (true) otherwise it assigned 'F' (false). Using this information and combining it with the right hand rule, all this orientation information avoids creating the complementary part of an arc of circle. This example shows how various orientation information combines to produce valid entities.

The ORIENTED_EDGE entity is part of half edge data and belongs to an ADVANCED_EDGE entity. Its orientation inherits the one of the ADVANCED_FACE it bounds. Pointing out of the volume is defined as the positive orientation of an ADVANCED_FACE entity. If, using one vector to represent the orientation of an ORIENTED_EDGE and another vector to represent the orientation of its neighboring ADVANCED_FACE, the result of their cross product should point out of the surface (see Figure 3.5). According to this definition, if the orientation of an ORIENTED_EDGE is the same as its corresponding EDGE_CURVE, it is marked 'T', otherwise it is marked 'F'. The status of ORIENTED_EDGEs in an EDGE_LOOP are identical since their orientation derives from that of the ADVANCED_FACE bounded by this loop. At level 3, the geometric surfaces have their orientation definition as well as geometric curves. As an example, a PLANE contains an axis system, the direction

of one of them follows the plane normal and this direction is set as positive. Similarly, with CYLINDRICAL_SURFACEs and CONIC_SURFACEs, the direction pointing toward their axis is negative. In case of SPHERICAL_SURFACE, the direction pointing toward its center is negative and the opposite is positive. The direction pointing inside a TOROIDAL_SURFACE is considered as negative. The orientation definition of the ADVANCED_FACE entity follows the same principle as the ADVANCED_EDGE one: if its orientation is identical to its underlying geometric surface orientation, it is marked as 'T' otherwise it is marked as 'F'.

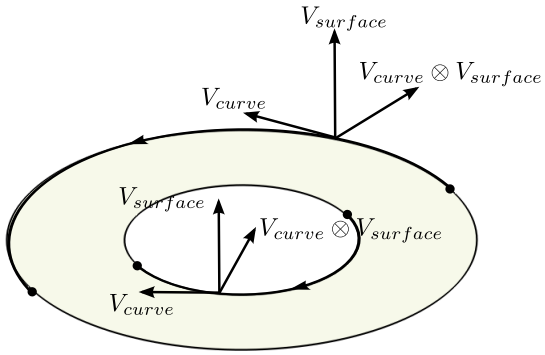


Figure 3.5: Relationship between curve and surface orientations.

STEP format does not only store geometric information of M_I, but also its topological description to define an entire B-Rep model. As already sketched through the description of the previous entities, a volume is divided into oriented faces (ADVANCED_FACEs) whose orientation is pointing outward. The boundary of these faces is described as oriented loops (EDGE_LOOPs), which combine with oriented edges. The geometric elements are independent and, somehow, the orientation of the geometric entities is consistent with that of the topological entities to describe correctly M_I but this orientation is not prescribed by the shape of M_I, it derives from the modeling process followed when generating M_I. With the boundary description of M_I, surfaces, curves and curve endpoints are available. A STEP file contains all the information needed to set up a symmetry analysis algorithm.

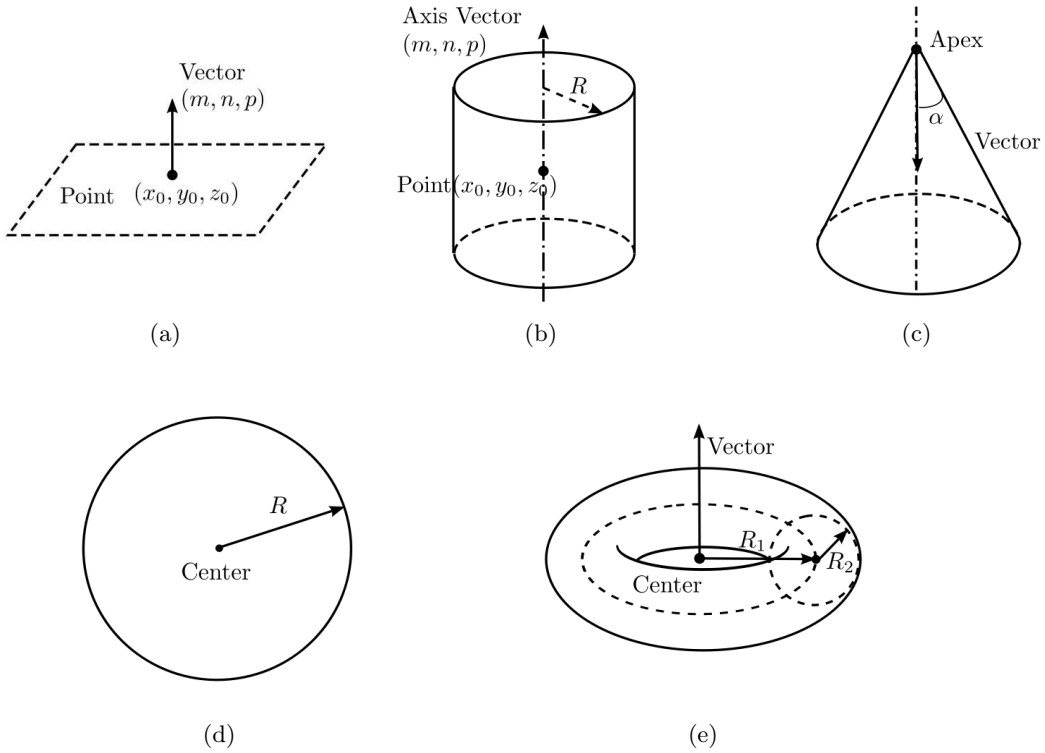


Figure 3.6: A graphic representation of the intrinsic parameters of the five reference surfaces.

LEVEL	Name	Elements	Example	Remark
LEVEL1	CARTESIAN_POINT	Coordinate	('Vertex', (0,0,0)) ;	basic points
	DIRECTION	Orientation	('Axis2P3D XDirection',(0,0,1)) ;	basic vector
LEVEL2	AXIS2_PLACEMENT_3D	CARTESIAN_POINT DIRECTION DIRECTION	('Plane Axis2P3D',#166,#167,#168) ;	Axis system
	VERTEX_POINT	CARTESIAN_POINT	(',#61) ;	
	VECTOR	DIRECTION Length	('Line Direction',#58,1.) ;	
LEVEL3	LINE	CARTESIAN_POINT VECTOR	('Line',#57,#59) ;	curve
	CIRCLE	AXIS2_PLACEMENT_3D Radius	('generated circle',#68,40.) ;	
	B_SPLINE_CURVE_WITH_KNOTS			
	PLANE	AXIS2_PLACEMENT_3D	('Plane',#169) ;	surface
	CYLINDRICAL_SURFACE	AXIS2_PLACEMENT_3D Radius	('generated cylinder',#55,40.) ;	
	CONIC_SURFACE	AXIS2_PLACEMENT_3D Radius Radian	('Cone',#55,3.5527136788E-015,0.9) ;	
	SPERICAL_SURFACE	AXIS2_PLACEMENT_3D Radius	(',#55,60.) ;	
	TORODAL_SURFACE	AXIS2_PLACEMENT_3D Radius Radius	('homeo Torus',#55,70.,20.) ;	
	SURFACE_OF_REVOLUTION	Linear AXIS2_PLACEMENT_3D	('Surface of Revolution',#95,#110) ;	
	B_SPLINE_SURFACE_WITH_KNOTS			
LEVEL4	EDGE_CURVE	VERTEX_POINT VERTEX_POINT Linear Orient	(',#62,#64,#60,.T.) ;	
LEVEL5	ORIENTED_EDGE	EDGE_CURVE Orient	(',*,*,#71,.T.) ;	half edge
LEVEL6	EDGE_LOOP	(ORIENTED_EDGE...)	(',(#84,#85)) ;	
LEVEL7	FACE_OUTER_BOUND	EDGE_LOOP Orient	(',#85,.T.) ;	external bound
	FACE_BOUND	EDGE_LOOP Orient	(',#142,.T.) ;	internal bound
LEVEL8	ADVANCED_FACE	(FACE_OUTER_BOUND FACE_BOUND...) surface Orient	('PartBody',(#127,#145),#123,.T.) ;	
LEVEL9	CLOSED_SHELL	(ADVANCED_FACE...)	('Closed Shell', (#91,#108,#118,#146,#177,#189,#199)) ;	volume
LEVEL10	MANIFOLD_SOLID_BREP	CLOSED_SHELL	('PartBody',#51) ;	
LEVEL11	ADVANCED_BREP_SHAPE_ REPRESENTATION	(MANIFOLD_SOLID_BREP...) Something	('NONE',(#138,#226),#46) ;	
LEVEL12	SHAPE_REPRESENTATION_ RELATIONSHIP	Something ADVANCED_BREP_ SHAPE_REPRESENTATION	(' ',' ',#49,#139) ;	

Table 3.1: The hierarchical structure and example of STEP format entities.

Surface	Parameter 1	Parameter 2	Parameter 3	Parameter 4
Plane	base point	normal vector		
Cylinder	axis point	axis vector	radius	
Cone	apex	axis vector	angle	
Sphere	center	radius		
Torus	Center	axis vector	radius 1	radius 2

Table 3.2: The intrinsic parameters of geometric entities used as reference surfaces.

3.3 From STEP format to B-Rep modeler datastructures

Indeed, STEP format can be regarded as very close to the datastructure defining B-Rep models in an industrial modeler. It is particularly the case with OpenCasCade [45] where the datastructure of a B-Rep model conforms with the entities of the STEP format. Consequently, the previous description can be regarded as a common denominator to configurations where a symmetry analysis is a process taking place either as part of a range of operators belonging to a modeling kernel or as a standalone process using a pre-existing B-Rep model available from a STEP file.

Based on the constitutive description of a B-Rep model, it appears that if each edge is adjacent to two faces, the effective boundaries of these two faces are described through two edge loops, each of which using independent curves to describe the edge 'shared' by these two adjacent faces. This geometric configuration comes from the fact that there is no exact solution to the representation of arbitrary intersection curves between two NURBS surfaces. This observation shows that the curves bounding the faces of M_I are not represented exactly. Rather, they are subjected to approximation processes performed during the modeling process. Consequently, the symmetry properties of these curves are not robustly contained in these entities.

Therefore, the principle of the proposed approach aims at setting up a robust process for the symmetry analysis, hence B-Rep curves are not part of the geometric data used for the analysis process. The characterization of the symmetry properties of M_I solely relies on surface parameters. These parameters being used to define some symmetry properties of the intersection curves between surfaces, the whole symmetry analysis process uses the location and intrinsic parameters of reference surfaces to improve, as much as possible, the robustness of this process.

It has to be noticed that the intrinsic parameters of the reference surfaces are effectively available whether the symmetry analysis takes place as part of an operator in a modeling kernel or as an independent process through the use of STEP files [22]. This observation validates the proposed symmetry analysis approach as a process compatible with any step of a PDP where a B-Rep CAD model is used.

To describe the five reference surfaces and based on the data stored in a STEP file, the intrinsic parameters of these surfaces are accessible. Whether a simple plane or revolution surfaces, they can be represented by several geometric parameters as illustrated in Figure 3.6 and listed in Table 3.2. They form the basis of the datastructure needed for the geometric information processed during a symmetry analysis process.

As highlighted in the previous section, the STEP format contains all the orientation information required to define the B-Rep model of a volume. During the symmetry analysis process, the shape of M_I is not modified. The boundary decomposition transformations described in the next chapter rely on hypergraphs that are non

oriented datastructures. However, the entities of these hypergraphs reference entities of M_I described through the STEP format. Consequently, model orientation is always available through these references if needed at some point of the symmetry analysis process.

From the orientation point of view, the STEP format is able to describe models with configurations extending some of the basic properties needed to model volumes, i.e. a loop bounding a face can be in contact with itself rather than restricting loops to be simple ones (see Figure 3.7).

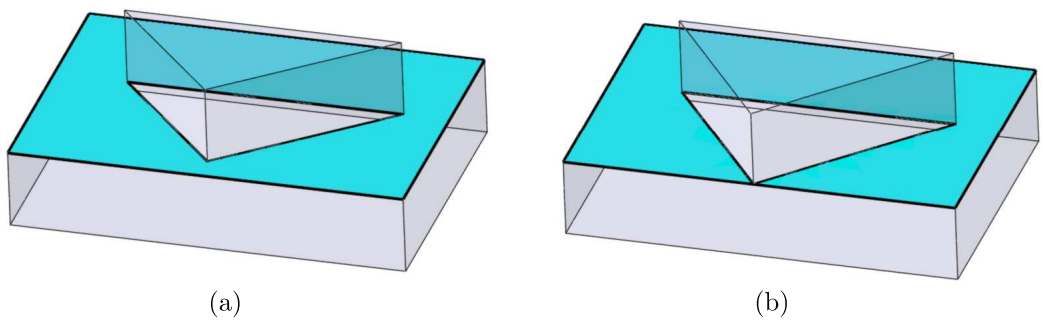


Figure 3.7: Examples of loops in B-Rep models: (a) a simple loop, (b) a loop in contact with itself.

It has also to be noticed that industrial CAD software, though they all contain volume modelers, contain extensions to the rigorous volume modeling theory and may describe objects that don't fit into the category of 2-manifold objects. Figure 3.8 gives an example of such a configuration where a vertex of the object has a non-manifold neighborhood. It appears also that this non-manifold singularity is also appearing in the STEP describing the object model. Consequently, the boundary preparation of M_I addressed in the next chapter must take into these configurations so that the objects processed in the symmetry analysis are effectively volumes bounded by a 2-manifold.

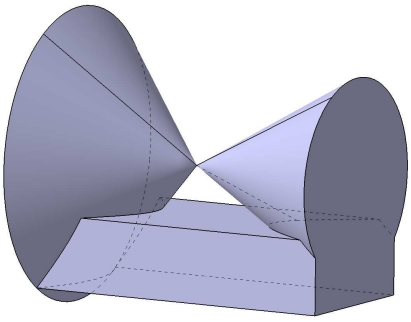


Figure 3.8: An example of object containing a non-manifold vertex that exists both in a CAD modeling kernel and in a STEP file.

3.4 Conclusion

The description and analysis of the STEP format have been useful to highlight the major entities of a B-Rep model used as input of the symmetry analysis process, M_I. The corresponding B-Rep datastructure is indeed the one used for the analysis process on top of which will be added the topological description of M_I that contains the maximal point sets and is described in next chapter using hypergraphs data models.

The analysis of the STEP format revealed also that its curve geometric description is not a robust model of face boundary description with respect to symmetry properties and justified the use of the sole surface parameters as geometric parameters to conduct the symmetry analysis.

Chapter 4

Object boundary description using a hypergraph datastructure

The aim of this chapter is to describe how to prepare an object boundary using a hypergraph datastructure. The hypergraph datastructure is the main work platform for the symmetry analysis algorithm. It can represent all the connections between topological elements while presenting the symmetry properties of the input object: Face, Edge and Vertex. It comes from the entities available in STEP format directly. With these basic operators, hypergraphs can be used to process objects with non-manifold singularities and convert them, if possible, to the shape category addressed. Also, hypergraph form the core frame work to generated the maximal faces and edges required to fit in the scope of the hypotheses. The B-Rep description of the object using hypergraphs stays a topological description. The geometry of the object is left unchanged by the hypergraph transformations.

4.1 Introduction

From the description of a STEP file structure in Chapter 3, it appears that the STEP format data structure is similar to a tree structure. The relationships between elements located at different levels of that structure are represented through branches. However, the connection between topological elements (faces, edges and vertices) is tedious to trace. Complementarily, hypergraphs aim at defining entities (faces, edges, vertices) forming point sets coinciding exactly with the real object, i.e. maximal faces and edges. Some of the properties of these maximal faces, edges no longer coincide with the properties of B-Rep faces, edges and vertices. Hypergraphs offer a general framework to define the maximal faces and edges needed for symmetry detection algorithms since they are able to describe a wide diversity of relationships between faces, edges and vertices [1], [15], [57], [65]. In 2008, Foucault et al. presented a hypergraph datastructure [19] for Finite Element meshing constraint generation. Here, hypergraphs are used to set up a datastructure dedicated to the description of the topology of the maximal faces and edges. This datastructure is placed on top of the B-Rep datastructures available from a STEP format or derived from a geometric modeler.

In Chapter 2, the shape model input to the symmetry analysis algorithm has been defined as a B-Rep model and it is noted M_I. When analyzing STEP file examples, depending on the geometric modeler, the STEP file generator and reader, non-manifold vertices may exist in B-Rep models. An algorithm is needed to detect them and split the vertices to match the 2-manifold requirements attached to volume models: the category of models targeted in the present work. Also, an algorithm generating maximal faces and maximal edges is needed. All these algorithms don't need to modify the shape of the B-Rep model, i.e. the geometric entities of M_I, they can operate purely through the concept of topological domain. As a topological representation, hypergraphs contain some efficient transformation operators for object topology description.

4.2 Hypergraph description

A hypergraph is constructed from two basic elements: Node and Hyper-Arc (Arc)(see Figure 4.1). Nodes are connected by arcs and both of them can exist independently [19]. A node can be linked to 0 or a finite number of arcs. An arc can connect 0 or a finite number of nodes, too. An arc can be split into half arcs. The number of half arcs of an arc is called the rank of the arc and noted R_A. A half arc has two categories of connections, one is attached to a node, the other one is attached to another half arc. Similarly, the number of half arcs connected to a node is called the rank of the node and noted R_N. A node can link zero or a group of half arcs only. Different connections of half arcs generate different arc names. If $R_A = 2$ and the two extremities of the arc are two different nodes, it is a regular arc, e.g. A_2 in Figure 4.1. If the two extremities connect to the same node, it is loop arc, e.g. A_5 in Figure 4.1. When an arc has more than two extremities, $R_A > 2$, and each one connect to different nodes, it is a hyper-arc, e.g. A_3 in Figure 4.1. Considering a hyper-arc, if its different extremities are attached to the same node, it is a hyper-loop.

In the context of object modeling, the description of its boundary is operated with three hypergraphs to cover all the possible connection configurations between entities (faces, edges, vertices) [15]. Each hypergraph describes adjacency relationships. Face, edge and vertex are the three topological elements defining an object boundary. The three hypergraphs represent the adjacency of face-edge, edge-vertex and face-vertex, and are noted $G21$, $G10$ and $G20$, respectively.

- $G21$ describes the face-edge adjacencies: a node represents a B-Rep face or a maximal face and an arc represents a B-Rep edge or a maximal edge. If this hypergraph describes a 2-manifold B-Rep object as produced by a geometric modeler, each B-Rep edge or maximal edge is adjacent to two surfaces and in $G21$, each arc connects exactly two nodes. The hypergraph reduces to a graph;

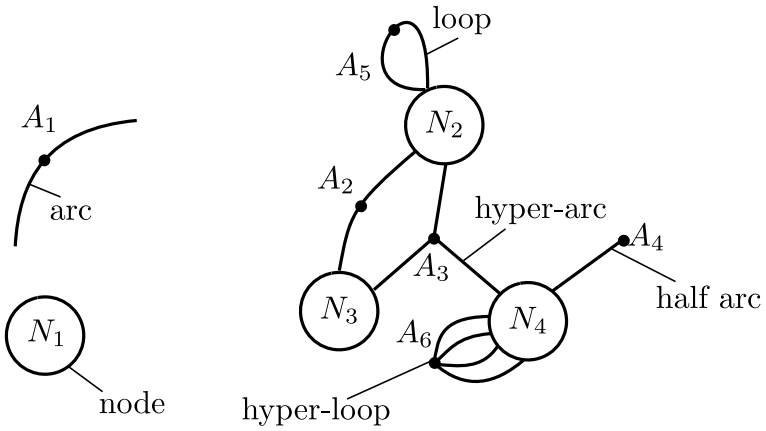


Figure 4.1: Hypergraph basic entities and hyper-arcs designations.

- $G20$ describes the face-vertex adjacencies: a node represents a B-Rep face or a maximal face and an arc represents a vertex;
- $G10$ describes the edge-vertex adjacencies: a node represents a B-Rep edge or a maximal edge and an arc represents a vertex. If this hypergraph describes a 2-manifold B-Rep object as produced by a geometric modeler, each edge has two extremities, hence each node in $G10$ connects with 2 arcs.

In the various hypergraphs, the nodes and arcs represent different entities. Consequently, the meaning of each half arc differs as exemplified in Figure 4.2 where S_i, Σ_j, P_k designates the geometric entities (surfaces, curves, points, respectively) of M_I and F_i, E_j, V_k stand for their corresponding topological entities (faces, edges, vertices, respectively). When $G21$ is generated from STEP file directly, a half arc is the half side of an edge or half edge in STEP format, which contains a half segment domain at any point on a half edge, see Figure 4.2a. In the B-Rep object boundary of M_I and in the boundary of this object using maximal faces and edges, there is no half edge in their object boundaries. M_I transformed through maximal faces and edges generation is noted M_{Max}. Hence, half arcs can only result from transient configurations during maximal faces and edges generation. In $G10$, the extremity of an edge is a half arc. A finite segment has $R_N = 2$. A half arc in $G20$ is a part of an open disk domain around a vertex.

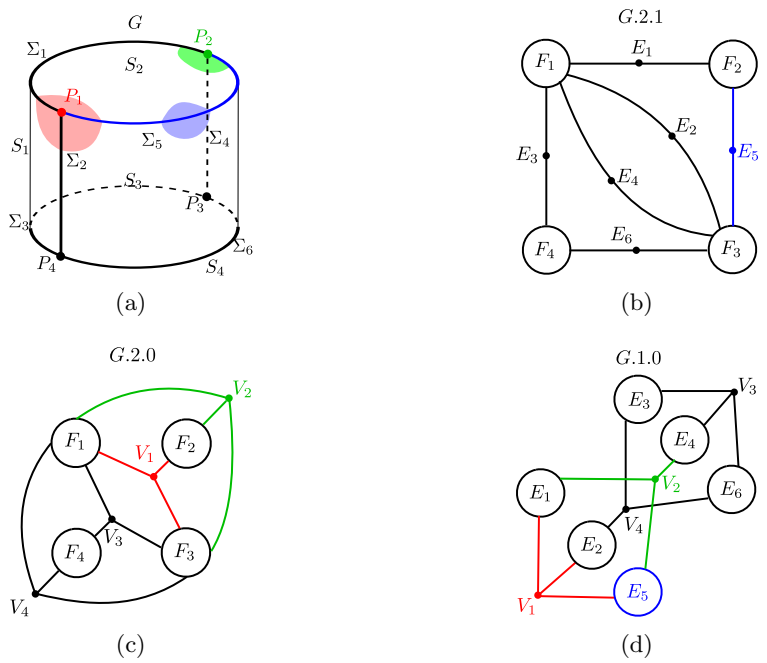


Figure 4.2: The topological meaning of a half arc in the various hypergraphs: (a) the geometric model of M_I, (b), (c), (d) the hypergraphs describing the topology of M_I.

4.3 Dual graph of a graph embedded in a surface

The decomposition of the volume boundary of M_I, ∂M_I, into surfaces, curves and points can be used to form a graph G embedded in ∂M_I. The dual graph of G is a graph G_D which has a node for each surface of G, and an arc for each curve in G joining two neighboring surfaces, for the embedding of G in ∂M_I. Each arc of G_D can be embedded in ∂M_I such that it cuts a curve of G at one point only (see Figure 4.3). The term 'dual' is used because this property is symmetric, meaning that if G_D is dual of G, then G is dual of G_D (if G is connected).

The general concept of dual graph can be applied locally at any vertex V_i of ∂M_I. If V_i is a vertex with no loop edge attached to it, faces attached to V_i appear only once around it. Then, M_I defining a volume, the neighborhood of V_i is topologically equivalent to a disk and the reduction of G to the neighborhood of V_i is a star-shaped graph. Consequently, the restriction of G_D around V_i reduces to a simple loop (see Figure 4.3). This illustrates the difference between a dual graph $G_{D|V_i}$ and hypergraph $G21$.

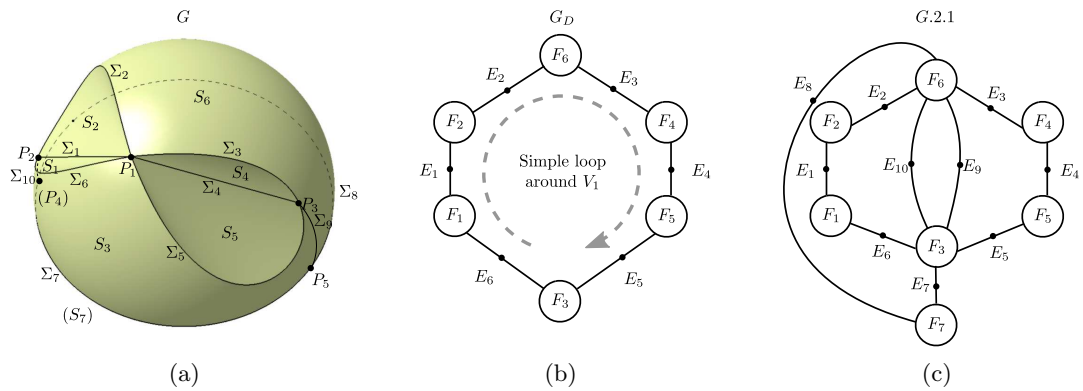


Figure 4.3: An example of dual graph G_D defined from the graph G obtained using the decomposition of a boundary ∂M_I of a B-Rep model: (a) is the original graph G embedded in ∂M_I; (b) is dual graph G_D of G around point P_1 (V_1); (c) is the hypergraph $G21$ of ∂M_I.

4.4 Hypergraph creation from a B-Rep model

In chapter 3, the data structure of STEP format has been introduced. Within this hierarchical data structure, ADVANCED_FACE at level 8, ORIENTED_EDGE at level 5 and VERTEX_POINT at level 2 are key topological entities describing a B-Rep model topology. Level 3 entities contain the geometric description of surfaces and curves. The CARTESIAN_POINT contains the coordinates of a vertex. Other levels contain either geometric information or complementary topological entities, e.g. edges without orientation, edge loops, which contribute to the definition of the key entities listed previously.

The STEP format is a typical B-Rep data structure. Depending on the model translation process from a geometric modeler to the STEP file format, depending on the internal model of this geometric modeler, the detailed model data contained in a STEP file may differ. Anyhow, a B-Rep model stored under a STEP format contains topological as well as geometric entities. A hypergraph points at the topological entities which are represented as F_{HG}, E_{HG} and V_{HG}. The geometric entities are attached to the topological ones. The connections between the topological entities are coming from the parent-child relationships contained in the STEP format. In this case, from the hypergraphs, it is possible to get all geometric parameters of the faces, edges and vertices (see Figure 4.4).

Because different users of different CAD software often produce distinct shape generation processes of the same object model, i.e. each shape is visually identical

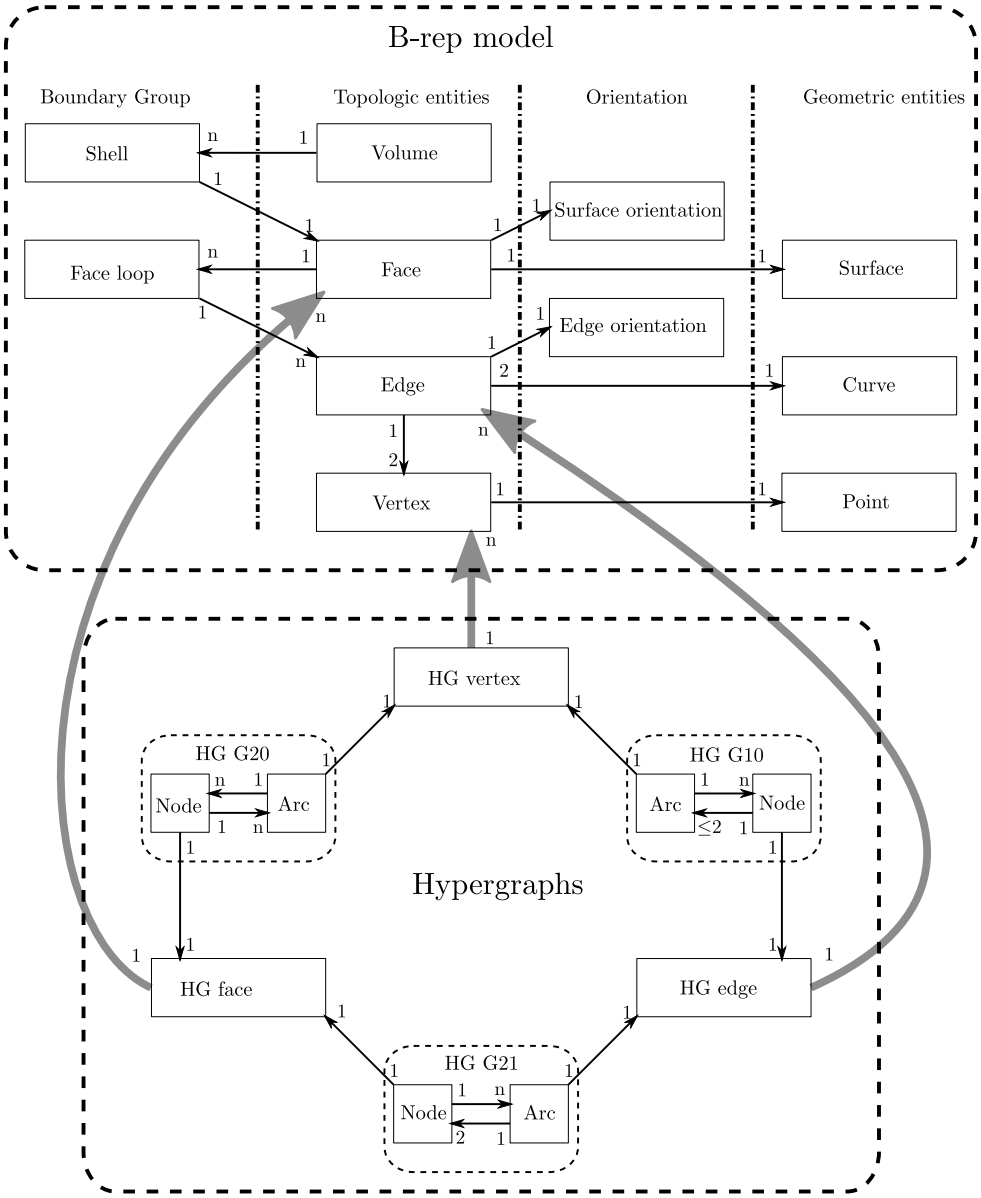


Figure 4.4: Hypergraph data structure.

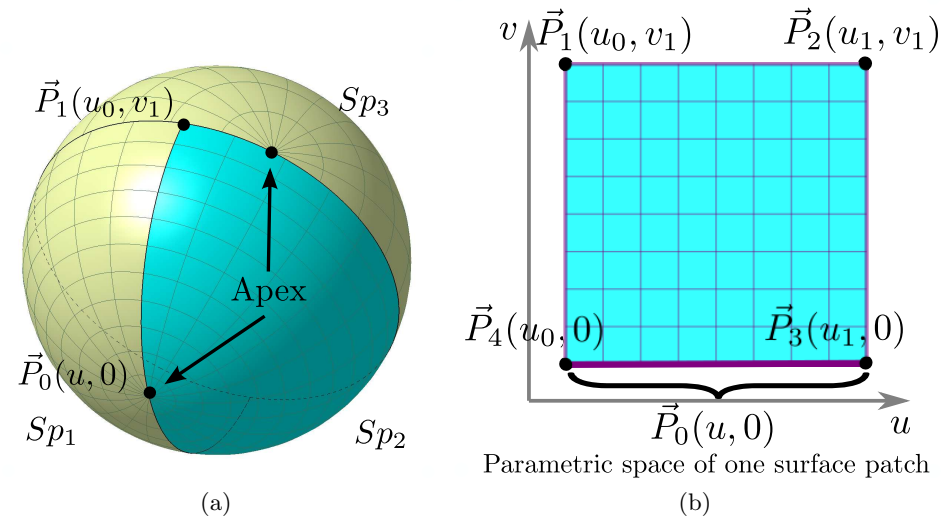


Figure 4.5: Spherical surface represented by different entities: (a) 3D representation of the sphere; (b) parametric space of a patch with a singular edge.

to the others, these models may contain different entities. More precisely, configurations with coinciding vertices or otherwise stated, non-manifold configurations around vertices, produce visually equivalent objects though they contain a different set of entities. Other configurations originate from the parametric space of surfaces, which are bi-parametric and described by square or rectangular domains in their parametric space. Some of their vertices must coincide when describing a surface of revolution: see Figure 4.5 the pole of the sphere behaving like the apex of a cone in 3D because vertices coincide together with a zero length edge. Consequently, visually similar objects may be represented by different entities of B-Rep models, some with coinciding vertices and others with zero length edges, etc. Indeed, the edge length l_E of an edge E may not be zero, as long as it is a curve shorter than the modeler tolerance, ε. Consequently, its two vertices, V_1, V_2 may not coincide exactly, rather their distance is smaller than the modeler tolerance, $\|\overrightarrow{V_1V_2}\| < \varepsilon$. These entities are usually created as a result of algorithm computations. It can be also the result of internal computations for the model translation between STEP standard and the internal model of a modeler or the opposite. Because of zero length edges and coinciding vertices, they impact the symmetry analysis of the input shape. To avoid such an influence and obtain a generic boundary description process, the zero length edges and coinciding vertices should be ignored prior to the generation of the hypergraphs forming the description of the input model M_I. The criteria supporting these transformations are:

Criterion 1 *If the edge length of E is smaller than the user-prescribed tolerance*

$l_E < \varepsilon_u$, this edge will not be collected as an HG_EDGE.

Criterion 2 *If the distance between two vertices V_1, V_2, is smaller than the user-prescribed tolerance $\|\overrightarrow{V_1V_2}\| < \varepsilon_u$, the second vertex V_2 will not be collected as a HG_VERTEX and its neighbor connections are moved to the first one V_1.*

To initialize the hypergraphs, the algorithm could be created based on criteria 1 and 2 in a first place. Then, it scans all topological faces in the B-Rep model input. For each face, it traces the boundary edges and an arc is added in $G21$ for each such edge. Using each edge of the B-Rep model, its extremities can be identified and the corresponding vertices are created in $G10$. Using the faces and vertices obtained in $G21$ and $G10$, they are inserted in $G20$ to populate it. After this process, the B-Rep model is transferred to the hypergraph datastructure where all its entities are valid within the scope of ε_u. It is a consistent topological representation since extraneous entities (edges, vertices) linked to the modeler behavior have been removed and these hypergraphs form the basic datastructure of the object boundary ∂M_I to be processed for symmetry analysis. This gives a precise definition of the content of M_I where its hypergraph description contains no edge smaller than ε_u and no vertices within a similar distance.

As a consequence of this process, loop edges can appear when vertices are merged that belong to tangent curves. The corresponding graph structure gets modified and the dual graph around a vertex V_i no longer reduces to a simple loop since a face can appear several times around V_i. It can become a set of simple loops connected through the faces appearing several times (see Figure 4.6) or exhibit dangling arcs when edges reduce to loops. The content of hypergraphs is no longer able to define the orientation of M_I but the hypergraphs are not oriented and the orientation of M_I is still available through the references to B-Rep entities accessible from the hypergraph entities. Further details about the relationship between a dual graph at a vertex and hypergraphs are formally given at section 4.6.

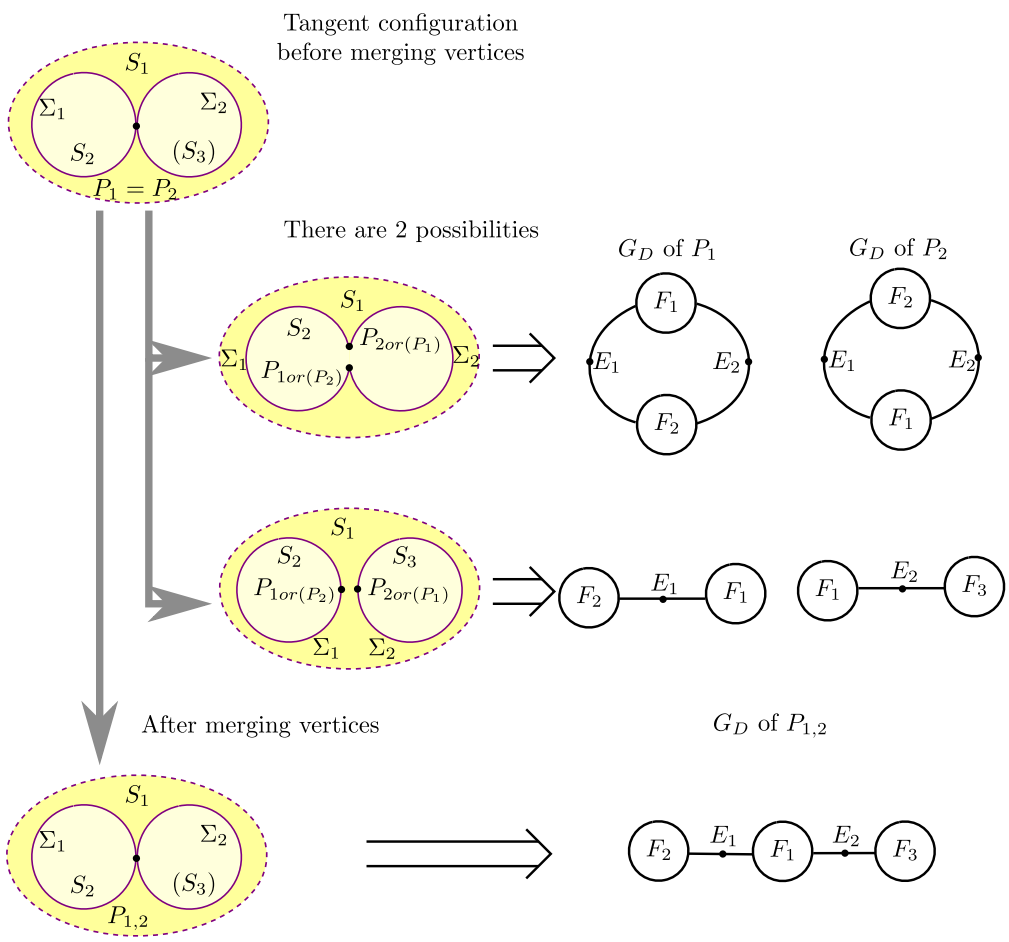


Figure 4.6: Illustration of the effect of vertex merge operations on the dual graph G_D around a vertex V_1 and V_2.

4.5 Hypergraphs transformation operators

Ignoring zero length edges and merging coincident vertices are preliminary steps of object boundary preparation. As mentioned before, the adjacent identical surfaces must be merged to generate infinite point sets where the symmetry properties are meaningful with respect to M_I. Hypergraphs, with their operators applied to the three hypergraphs describing M_I, are related to each other. The purpose of these operators is to generate M_{MAX}, the model containing maximal faces and edges and having the same shape as M_I. It is the purpose of this section to describe the hypergraphs operators individually that help breaking down the overall transformation from M_I to M_{MAX} into simpler processes. In a first place, generic or basic operators are proposed

to form the building blocks for maximal faces and edges generation.

4.5.1 Generic hypergraph transformation operators

A hypergraph datastructure connects to basic operators to edit its nodes and arcs. The most basic operators are arc creation, node creation, arc and node removal. An operation applied to an arc can be refined further down to half arc level. Adding an arc can be further decomposed into the addition of a half arc linking the candidate node or a half arc which already exists (see Figure 4.7a), or other variants of this operation are depicted also in Figure 4.7a. Figure 4.7a illustrates configurations where R_A is incremented through half arc creation. Because of the node-arc connection creation, a new node may be added to some arc neighborhood whose number of its surrounding nodes is less than its rank (see Figure 4.7b). The effective connection of a node N to a half arc increments its rank R_N. As an example, the connection of a node N_4 to an arc already connected to a node N_2 increments R_{N_4} and R_{N_2} both (see Figure 4.7b).

The arc and node removal operators are opposite to the creation operators. The arc removal removes a half arc and decrements its arc rank R_A (see Figure 4.7c). The node removal deletes the candidate node N, but keeps the surrounding half arcs (see Figure 4.7d). As a result, the ranks R_{A_i} of the half arcs surrounding N are preserved while the ranks of the nodes N_i surrounding N are decremented. The split and merge operators are a couple of opposite operations, too. With the split operator, whether it is an arc or a node, the original entity E will be split into two new ones E_1, E_2. Its surrounding connections are divided into two groups and each group connects to one of the new entities E_1, E_2. Consequently, if a hyperarc A or a node N is split, then the ranks of the new ones E_1 and E_2 satisfies R_A or $R_N = R_{E_1} + R_{E_2}$.

The merge operator uses a new entity E to replace the two original entities E_1 and E_2 and preserves all the surrounding connections and the ranks are summed up, i.e. $R_{E_1} + R_{E_2} = R_E$ (see Figure 4.8c,d). Arc contraction is a high level operator. It can be regarded as a node merging plus an arc removal operations. The selected arc A contracts and all nodes N_i connected to this arc are merged together to produce a single node N: $R_N = \sum_i R_{N_i} - R_A$. As a result, the contracted arc disappears (see Figure 4.8e). The parallel arc merging differs from the arc merging operator. It has to happen to two arcs at least, whose ranks equal 2 and connect to the same nodes. It can be considered that one arc only will be kept and the others are deleted, as shown in Figure 4.8f.

These basic operators purely focus on the graph domain, independently of any application context. As a topological data structure, the three hypergraphs describe the boundary decomposition of M_I. When these hypergraphs can be reduced to a graph, which is the case when the B-Rep datastructure of an object is loaded and really describes M_I, this graph can be used to define the genus of M_I using the Euler

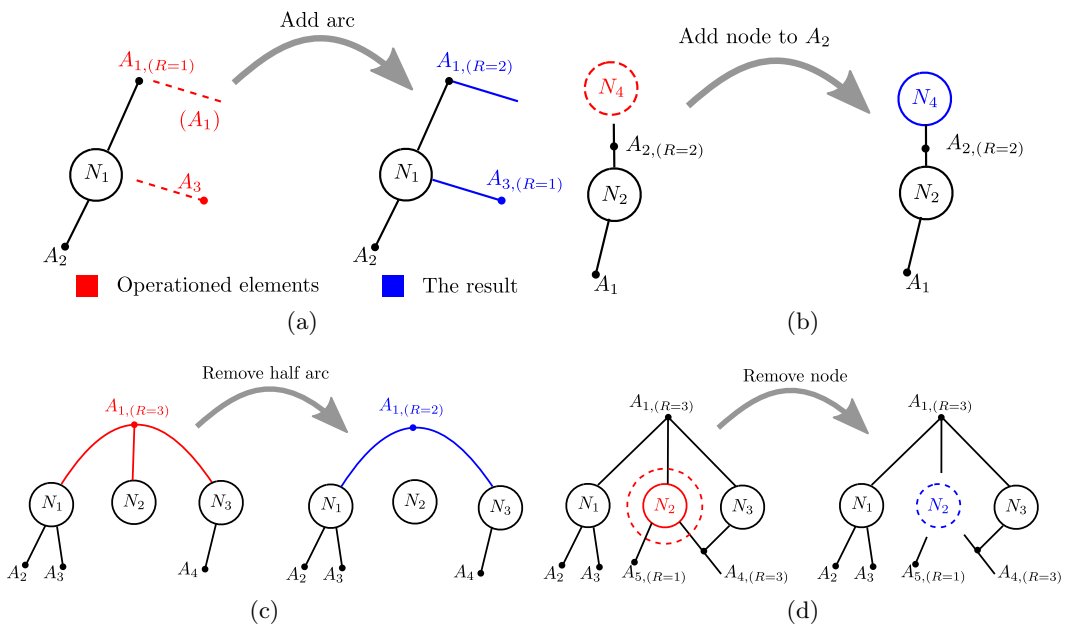


Figure 4.7: llustration of basic hypergraph operations (first subset).

theorem as follows:

$$X(M_I) = \#vertices - \#edges + \#faces. \tag{4.1}$$

Here, M_I designates a B-Rep model defining a volume as it can be defined in the STEP standard or a geometric modeler. In order to preserve the Euler characteristic $X(M_I)$, any operation in one hypergraph must trigger some operation in the other two hypergraphs. Because M_I should be a volume and a 2-manifold boundary model to conform to the current framework, there are several properties characterizing this category of models in the hypergraphs. These properties support the non-manifold configuration detection algorithm used as input to certify or transform, if needed, M_I into a real volume forming a 2-manifold model. As mentioned at section 4.4, non-manifold vertices can exist in M_I and must be removed to conform to the proposed hypotheses. In addition, such vertices are not part of real industrial components.

Property 1 $\forall R_{A_i-G21}, R_{A_i-G21} \equiv 2.$

This property expresses that each edge must be exactly adjacent to two faces, a necessary condition to define a 2-manifold. $G21$ is a binary graph. Using the definition of a half arc in $G21$, if $R_{A_i-G21} \neq 2$, it means the neighborhood of the edge A_i is not a disk. So, it is either a non-manifold edge or a boundary edge or an isolated edge.

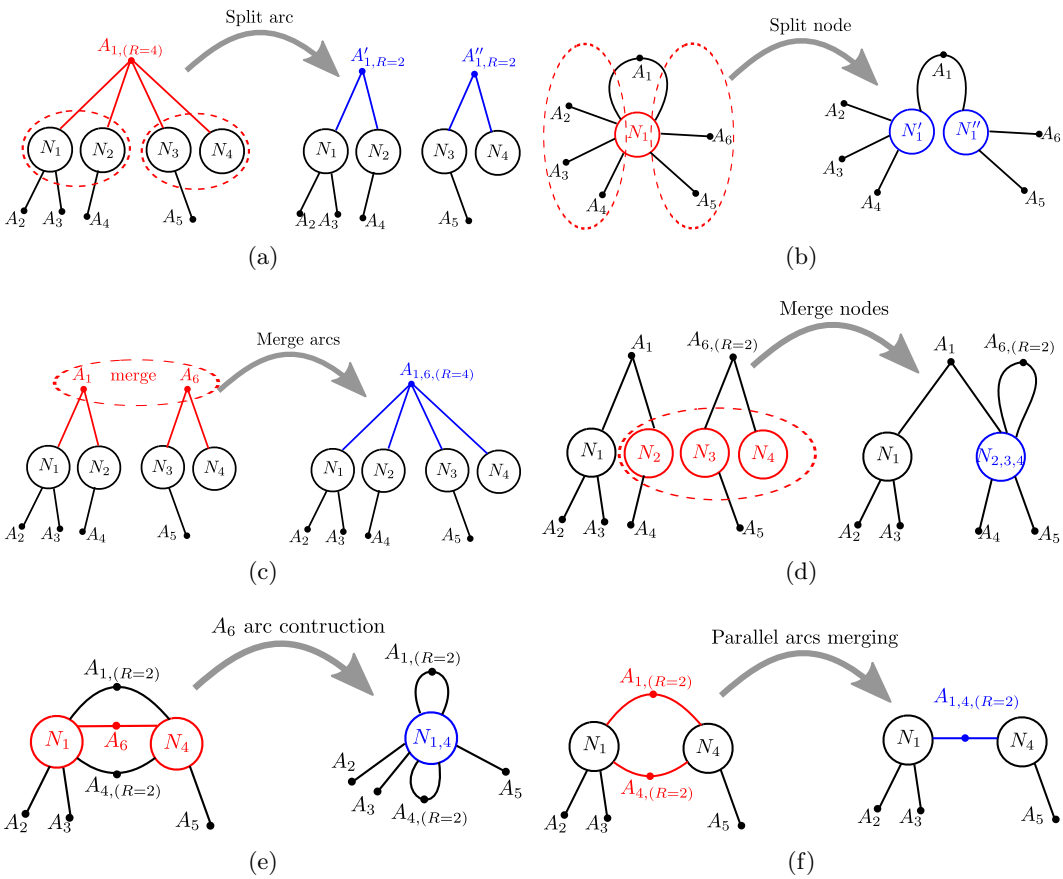


Figure 4.8: Illustration of basic hypergraph operations (second subset).

Property 2 $\forall R_{A_i-G20}, R_{A_i-G20} = R_{A_i-G10}$.

A vertex V_i is surrounded by the set of faces $[F_{\Omega V_i}]$ and the set of edges $[E_{\Omega V_i}]$. The cardinality of $[F_{\Omega V_i}]$ is R_{A_i-G20}. The cardinality of $[E_{\Omega V_i}]$ is R_{A_i-G10}. Because each edge of a 2-manifold is the intersection between two faces, the number of faces around V_i can reach up to twice the number of edges around this vertex, hence $R_{A_i-G20} \geqslant R_{A_i-G10}$ (see Figure 4.9a). Any vertex V_i lies at the intersection of edges or, at least, defines the extremities of an edge loop, so $R_{A_i-G10} > 1$. Complementarily, the cardinality of $[F_{\Omega V_i}]$ has to be bounded and that of $[E_{\Omega V_i}]$ is such that $R_{A_i-G20} \leqslant R_{A_i-G10}$. Indeed, the number of faces gets smaller than edges when faces are bounded by edge loops in arbitrary configurations of faces around a vertex (see Figure 4.9b). The result is $R_{A_i-G20} = R_{A_i-G10}$. This property expresses that each vertex has a neighborhood topologically equivalent to a disk exactly: another necessary condition to define a 2-manifold.

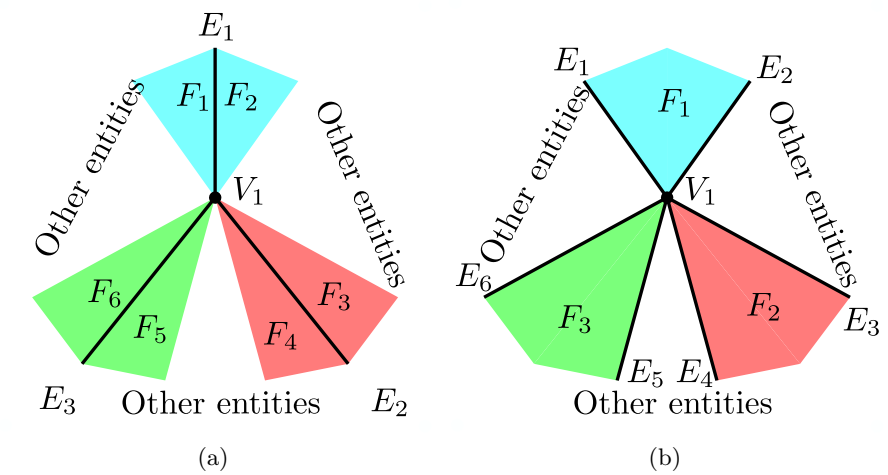


Figure 4.9: The illustration of $R_{A_i-G20} \geqslant R_{A_i-G10}$ and $R_{A_i-G20} \leqslant R_{A_i-G10}$.

To prepare a maximal 2-manifold model, M_{MAX}, suited to define all the candidate symmetry planes attached to point sets, several topological operators are needed to transform the initial boundary decomposition of the model, i.e. M_I, into maximal entities, i.e. M_{MAX}: non-manifold configuration detection, split non-manifold vertex, merge surfaces, merge edges and split manifold vertex are the corresponding operators described in the following section.

4.5.2 Non-manifold configuration detection

In a B-Rep model, depending on the geometric modeler having generated this model, even though this modeler is said a volume modeler, it is possible that this model M_I contains non-manifold entities: either vertices or edges. Based on the experience of industrial modelers, non-manifold configurations happen at vertices for two reasons: one is because the original B-Rep model contains non-manifold vertices, the other derives from vertices collecting process when geometrically coinciding vertices are merged. Under the hypotheses of the proposed approach (see Chapter 2), non-manifold entities are forbidden. Consequently, the first task is to detect non-manifold vertices and then to split them to produce a manifold model.

Hypergraphs form a typical topological representation of objects. A non-manifold vertex has a special structure across the three hypergraphs describing an object. Figure 4.10 is the illustration of non-manifold configurations detection. In Figure 4.10a, vertex V_1 is adjacent to six faces forming two disks, which is highlighted in hypergraph $G20$. From hypergraph $G21$, it can be observed that within the six faces around V_1, they connect each other into two independent loops: $[F_1, F_2, F_3]$ and $[F_4, F_5, F_6]$. Each

loop defines an 'open disk' around V_1, thus characterizing the non-manifold status of V_1. Figure 4.10b is an example of a non-manifold edge, E_7 characterized by a rank $R_A = 4$ in $G21$. Non-manifold edges can be easily detected to make sure that M_I is free of them and really a 2-manifold object.

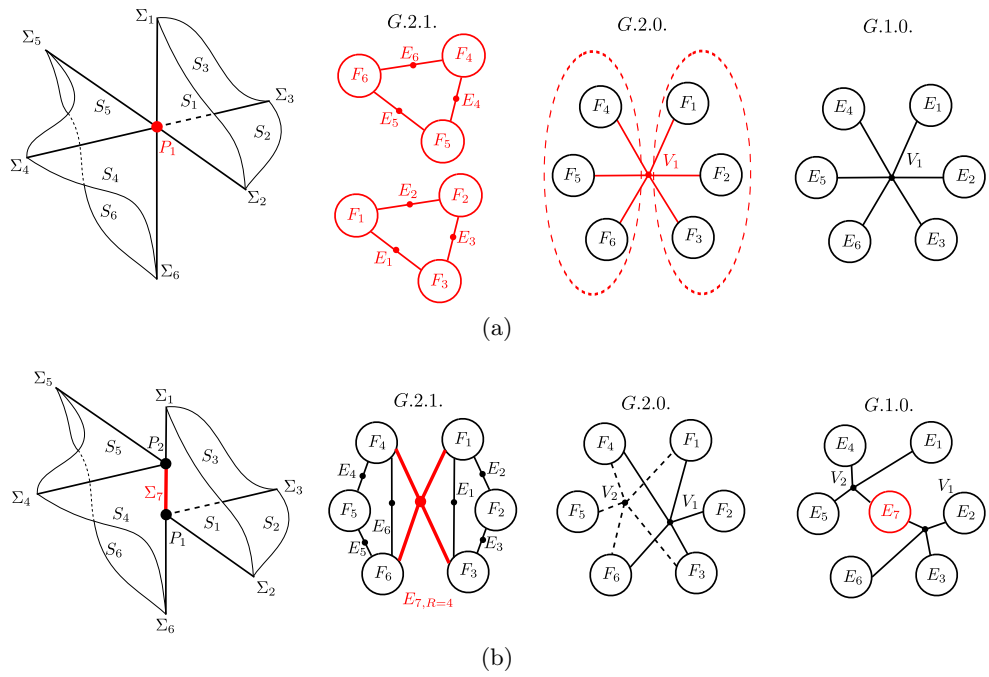


Figure 4.10: Non-manifold entity detection in hypergraphs: (a) V_1 is a non-manifold vertex characterized by two disks; (b) E_7 is a non-manifold edge with a two disks neighborhood.

The purpose of the non-manifold vertex detection is to analyze the hypergraphs neighborhood around of each vertex. The hypergraph neighborhood of a vertex V_i is based on a subgraph of $G21$ to form a dual graph at V_i, which contains all its surrounding entities (see section 4.3 for the definition of a dual graph at a vertex and section 4.6 for the equivalence between a dual graph at a vertex and a subgraph of $G21$).

The dual graph at a vertex V_i is formed from its surrounding faces and edges, respectively. In $G21$, these faces and edges are represented as nodes and arcs linking each other and each face corresponds to one node exactly. Similarly, each edge of M_I appears exactly once in $G21$ as an arc. At a vertex V_i, if all the edges connected at it differs from each other and similarly, if each face connected at it differs from each other, the corresponding subgraph of $G21$ forms the dual graph at V_i. As shown in an example at Figure 4.10a, $G21$ contains several connected components, each one

forming a simple loop. Indeed, it is the dual graph of vertex V_1 in this case. More generally, when loops bounding faces are tangent to each other and/or when loops contain singular intersection points between surfaces, faces and/or edges can appear several times around V_i, the dual graph around V_i and a subgraph of $G21$ restricted to the neighborhood of V_i are still similar but they are no longer formed of simple loops (see Figure 4.11).

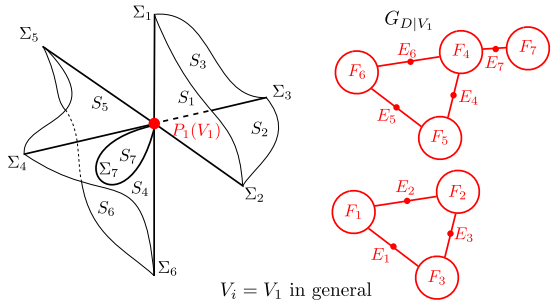


Figure 4.11: The dual graph at a non-manifold vertex V_1.

In a vertex dual graph, each connected component represents one disk around this vertex. This property is used to detect no-manifold vertices (see Figure 4.10a).

The process of dual graph creation starts from an arc in $G21$ (an edge connected to V_i) obtained from any half arc in $G10$ using V_i as input and traverses $G21$ up to one of its connected node F_k (face connected to V_i). Then, it looks in $G21$ for the next connected node along the arcs surrounding the current node. Because of Property. 1, when the traversal process is going through an arc, it always can find a next node. This traversal proceeds until it cannot find any new arc. If the last node is the start node, the corresponding subgraph is a dual loop around V_i and forms a disk or a subset of a disk. Otherwise, it is an open chain, which would indicate an open surface and must not happen. Indeed, it is an open surface if an edge in $G21$ is not connected to two faces otherwise a dangling edge in the dual graph indicates a loop edge (see Figure 4.11).

If the half arcs attached to V_i in $G10$ haven't been all visited, other loops exist around V_i. When traversing $G21$, a node can be reused to form loops connected to each other, but each arc can be traversed only once. If there are arcs left in $G10$, the traversal will start over again within the arcs left to create a new component of dual graph, until all the $G10$ arcs are traversed.

Finally, several new graphs are created. They form dual graphs G_{D_i} at all the model vertices. Because of Property 2, M_I describing a volume, there must not be any open chain existing around V_i. However, if a face F_k appears p times around V_i, $p > 1$, G_{D_i} contains $(p-1)$ loops connected at F_k rather than a simple loop around V_i. In addition, if q loop edges bound faces attached at V_i, G_{D_i} contains q dangling

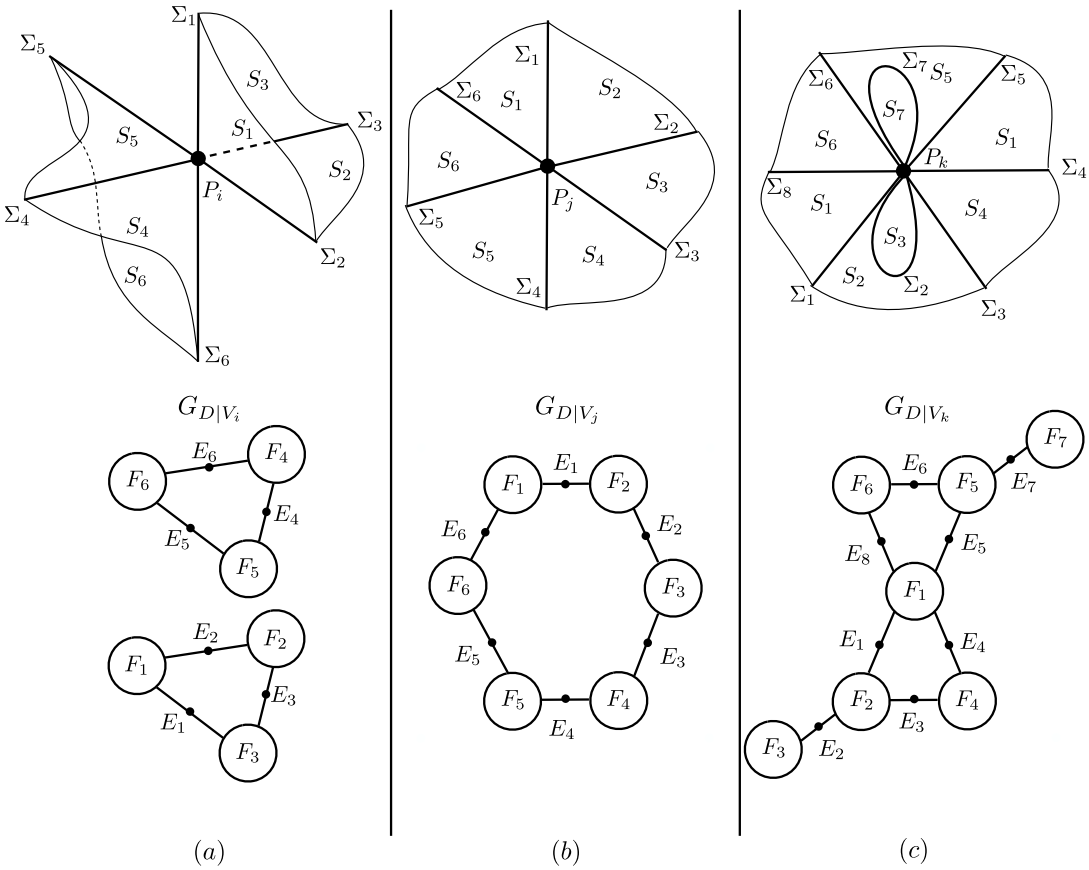


Figure 4.12: Three different dual graphs: (a) a non-manifold vertex V_i; (b)(c) at manifold vertices V_j, V_k.

edges. Anyhow, V_i is still a manifold vertex in each of these configurations as long as G_{D_i} contains only one connected component. For example Figure 4.12 illustrate three possibilities of dual graphs at vertices: (a) G_{D_i} two connected components, V_i is non-manifold, (b) G_{D_j} reduces to a simple loop, V_j is manifold, (c) G_{D_k} contains several loops and dangling edges forming a single component, V_k is manifold. The main criterion of the algorithm to detect non-manifold vertices using the dual graph components extracted from the hypergraphs can be summarized as:

Criterion 3 *For each vertex V_i:*
using its surrounding faces $[F_{\Omega V_i}]$ and edges $[E_{\Omega V_i}]$, and traversing $G21$ using $G10$ to create a dual graph G_{D_i}:
if G_{D_i} contains more than one component, V_i is a non-manifold vertex;
if G_{D_i} contains one or more dual loops, each one sharing nodes with another dual loop and/or if dangling edges are connected to these loops such that G_{D_i} contains only one

connected component, V_i is a manifold vertex.

4.5.3 Non-manifold vertex splitting

Having identified the non-manifold vertices, they are subjected to the following operator. The non-manifold vertex splitting process uses the split arc operator described at section 4.5.1. After non-manifold vertex detection, each dual graph G_{D_i} contains several connected components that divides $[F_{\Omega V_i}]$ and $[E_{\Omega V_i}]$ into the same number of components, N_c. The number of connected components minus one $(N_c - 1)$ is the number of times V_i has to be split. After the non-manifold vertex splitting process, from a topological point of view, there are several coinciding vertices but the shape of M_I has not changed. Indeed, these coinciding vertices point to the same geometric points. M_I described with the hypergraphs has been changed into a manifold model M_{MI} suited for symmetry analysis purposes. The non-manifold vertex splitting process is as follows for each dual graph G_{D_i}:

- Traversal of the first $(N_c - 1)$ components in G_{D_i} to group the nodes in $G20$ and $G10$;
- Generate the corresponding groups as reference, split arcs in $G20$;
- Generate the corresponding groups as reference, split arcs in $G10$.

4.5.4 Faces merging operator

In order to generate the maximal faces of the B-Rep model M_{MI}, a merge faces operator is needed. At first, the merge operation should only happen between adjacent faces, F_i, F_j. Because of Prop. 1, after merging these faces, their common edge will disappear (see Figures 4.13, 4.14). So, in hypergraph $G21$, the merge faces operator becomes an arc contraction operation where the new face is located in node $F_{i,j}$ (see Figure 4.13a, b).

In $G10$, the edge loss can be represented with a node removal operation. But only removing this node in $G10$ is not enough to preserve the topological consistency of the B-Rep model. Indeed, from a topological point of view, the edge removal is not only removing the common edge, but also removing the links between this edge and its extreme vertices. With the definition of a hyperarc in $G10$, after the node removal, the half arc connecting to the removed node must be removed too, reducing the rank of the corresponding hyperarc. In this case, a half arc removal is needed in $G10$. Because of Prop. 2, a hyperarc in $G20$ should be subjected to the same operation. Figures 4.13 and 4.14 illustrate the successive elementary steps of a face merging operation with

the hypergraphs transformations. F_1 and F_2 are merged and E_3 is removed. In Figure 4.13b, the related arc in $G21$ contracts. The node merging and node removal follow in $G20$ and $G10$. Then, modifying the hyperarcs in $G20$ and $G10$ is obtained using the half arc removal operator to reduce the arc rank. Figure 4.13b,c represents a complete process of faces merging operator and d, e forms another configuration of face merging operation where the two candidate faces are identical, i.e. the edge to be removed is internal to a face. This time, $F_{1,2}$ describes only one face, the node merging operation in $G20$ does not change anything. The vertex V_2 becomes an internal vertex and is deleted. The face merging process can be summarized as follows:

- E_i connects faces F_j and F_k in $G21$ and it connects V_s and V_t in $G10$;
- select the edge E_i for deletion because faces F_j and F_k need be merged;
- contract arc E_i in $G21$;
- merge nodes F_j, F_k in $G20$;
- remove node E_i in $G10$;
- remove the half arcs to reduce the rank of V_s and V_t in $G20$ and $G10$.

The topology of M_{MI} is now transformed into a new one and the maximal face generation is the first boundary transformation process applied.

4.5.5 Edge merging operator

In Figure 4.14b, E_1 and E_4 are geometrically defined as semi-circles having the same radius and centre, so they can be merged to form a maximal curve. The edge merging operator applies to a vertex V linking no more than two edges and this vertex belongs to the intersection between the faces partially bounded by E_1 and E_4 in the present example. In other words, edge merging reduces to a vertex removal operation. This operator starts in $G10$. Similarly to the face merging operator, it performs an arc contraction, too. Within $G20$, the arc related to V is deleted. Because two edges merge into one edge, in $G21$ the operation is a parallel arc merging operation. Figures 4.15 and 4.16 show the details of the edge merging operator applied successively to V_1 and V_3. After applying the edge merging to V_3, the edge $E_{1,4}$ appears as an isolated node in $G10$ and it isolates $F_{1,2}$ in $G20$ too, where the face is bounded by $E_{1,4}$ only.

The particular case where the rank of V is null, which means that V is isolated, is included in this operator since it is a configuration possibly resulting from the face merging operator but it needs a vertex removal operation, which is the content of the edge merging operator. The edge merging process summarizes as follows:

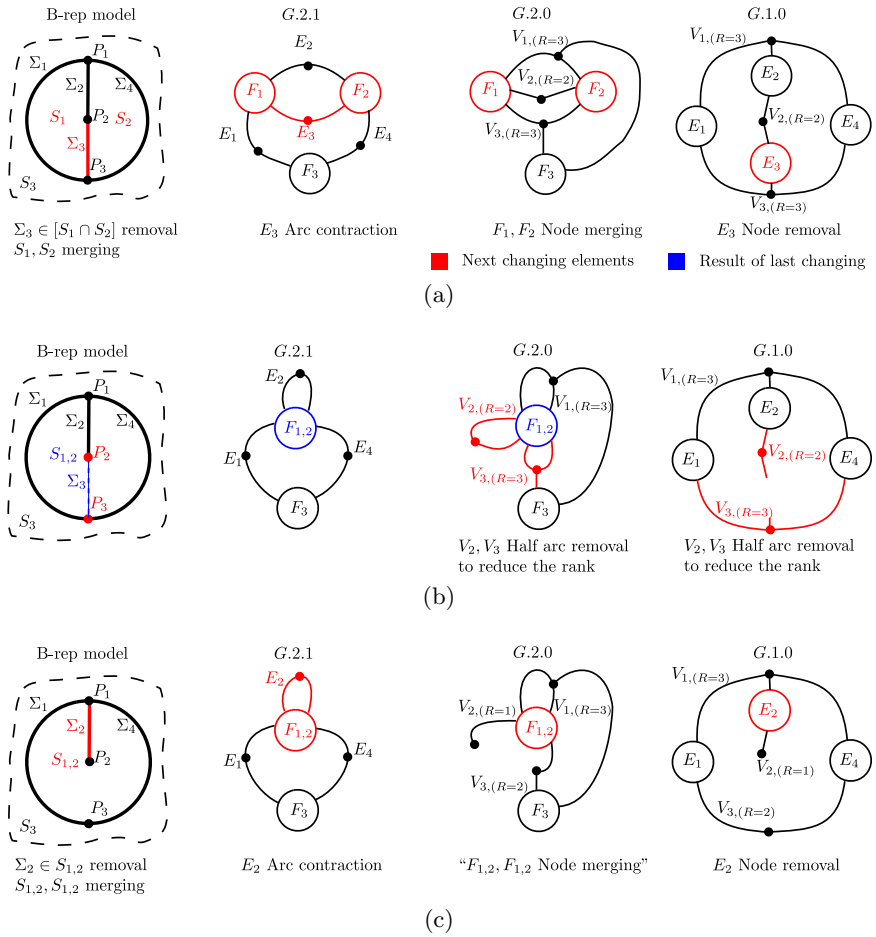


Figure 4.13: The steps of the face merging operator (first subset): (a) is the original B-Rep model M_{MI} (subset) and its hypergraphs; (b),(c) are the removal of E_3 to merge S_1 and S_2.

- After applying the face merging operator, the arc rank of V_i in $G10$ is $R_{V_i} \in \{0, 1, 2\}$;
- if $R_{V_i} = 0$, it is isolated:

  remove arc V_i in $G20$;

  remove arc V_i in $G10$;
- if $R_{V_i} = 1$, V_i lies on one edge, which is a loop edge that has coinciding extreme points:

  remove arc V_i in $G20$;

  remove arc V_i in $G10$;

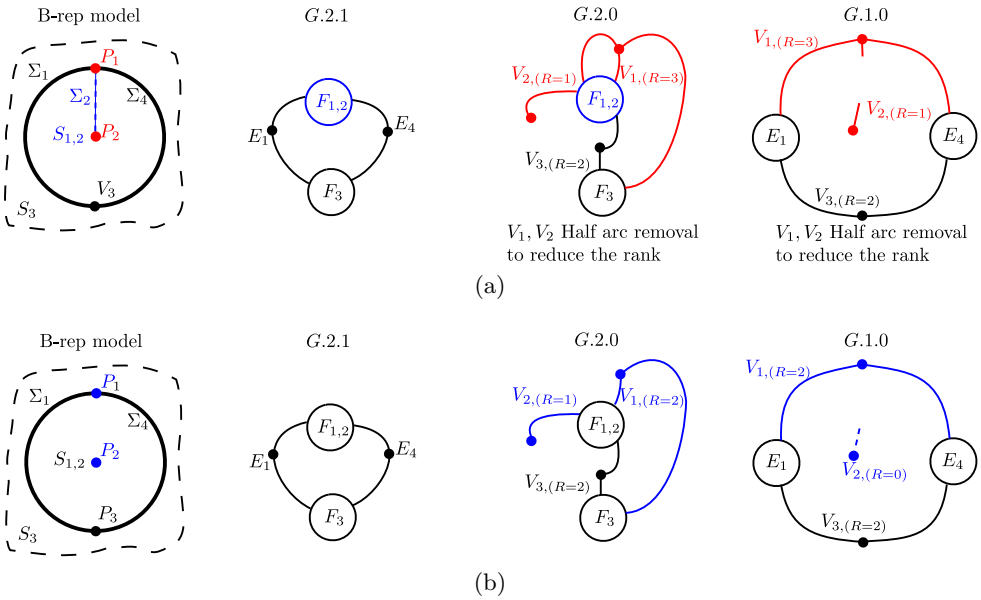


Figure 4.14: The steps of the face merging operator (second subset): (a), (b) are the removal of E_2 to "merge" $S_{1,2}$ with itself.

- if $R_{V_i} = 2$, V_i connects edges E_j and E_k that should be merged:

  contract V_i in $G10$;

  remove V_i in $G20$;

  apply the parallel arc merging operator to E_j and E_k in $G21$.

The edge merging process carries on the boundary transformation process of M_{MI} that becomes now M_{MAX} if there is no singular point among the boundary vertices of M_{MI}.

4.5.6 Manifold vertex splitting operator

This operator aims at modifying the boundary decomposition at vertices when face loops produces some singular configurations that will be described in the next section. It is applied after the face merging operator and before the edge merging one so that the latter one produces the M_{MAX} model. This operator is mandatory to avoid loosing symmetry or axisymmetry information when analyzing M_{MAX}.

Manifold vertex splitting at V_i is more complex than the non-manifold vertex splitting operator. It contains arc insertion, node insertion and node merging operators

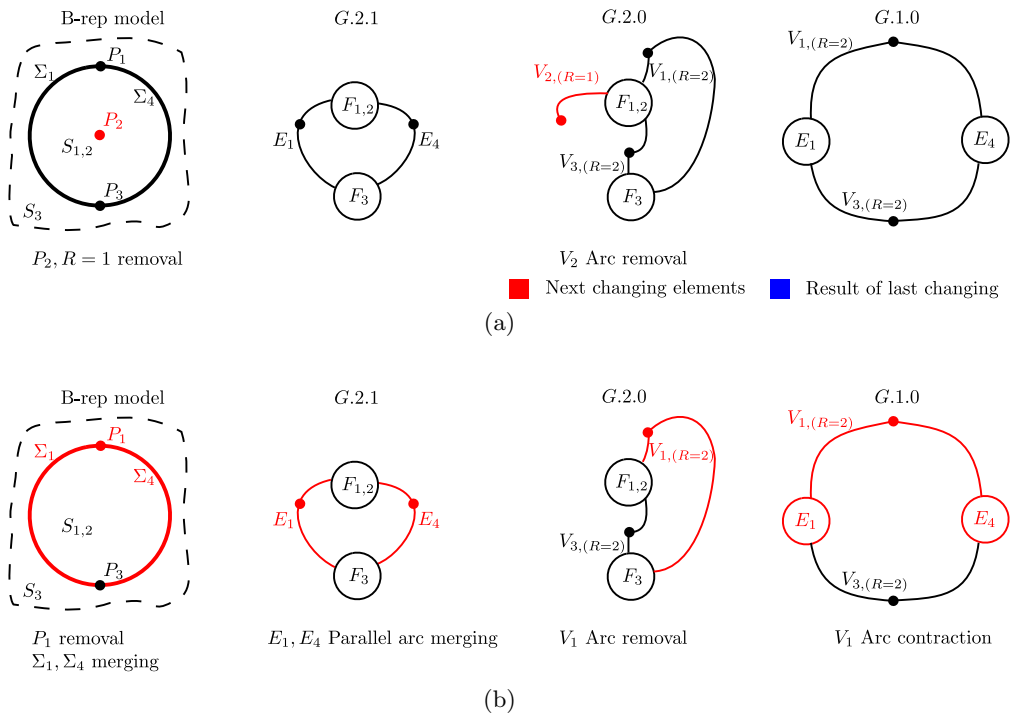


Figure 4.15: The elementary operations of the edge merging operator (first subset): (a) is the original area of an M_{MI} model and the corresponding hypergraph where V_2 is the first vertex to be removed as an example of isolated vertex; (b) and Figure 4.16a are V_1 and V_3 sequential removals.

as elementary operators because it incorporates a face merging operation (see Figure 4.17). For sake of simplicity, this operator is illustrated here with the example of Figure 4.17 where the graph around V_1 is a simple loop. The next section will state its generic criteria and describe it more formally. Globally, this operator needs the support of a dual graph for the candidate vertex V_1: G_{D_1}. More generally, this dual graph reduces to a subset of $G21$ restricted to V_i. Then, the first elementary task is to select in G_{D_1} the nodes defining the two faces to be merged, here F_3 and F_7 because they are assumed to be identical reference surfaces around V_1. After this merging operation, the nodes surrounding V_1 in G_{D_1} form two loops indicating how the surfaces and edges can be divided into two groups (see Figure 4.17b). These groups are used to update $G21$. The other hypergraphs will be divided into the similar groups of faces and edges in $G20$ and $G10$, respectively. In $G20$, the two groups of nodes (F_7, F_1, F_2, F_3) and $(F_3, F_4, F_5, F_6, F_7)$, are used as input for the arc splitting operation. Following the arc splitting operation, the merged faces F_3 and F_7 in $G21$ are now merged in $G20$. At same time, the arc V_1 is split into V_1' and V_1''. In $G10$, only the arc splitting is needed

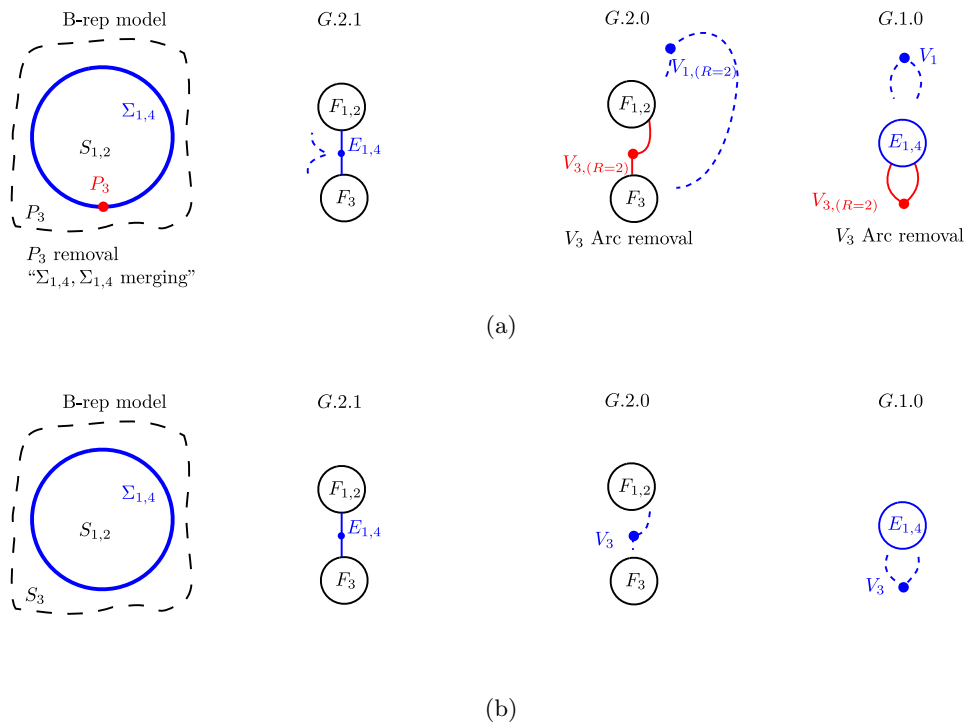


Figure 4.16: The elementary operations of the edge merging operator (second subset): Figure 4.15b and (a) are V_1 and V_3 sequential removals; (b) is the configuration after removing V_1, V_2 and V_3 to merge E_1 and E_4.

based on the unique sets of edges defined in $G21$: $(E_7, ..., E_2)$ and $(E_3, ..., E_6)$.

A more generic statement of this operator applied at V_i summarizes when the split operation appears between two nodes of a simple loop that are not shared with other loops:

- In the dual graph G_{D_i} (based on $G21$) of V_i, select the nodes F_j and F_k to be merged and generate the groups described by the resulting loops: $F_j, ..., F_{k-1}, F_k$ and $F_k, F_{k+1}, ..., F_{j-1}, F_j$;
- In $G21$, merge the nodes F_j and F_k;
- In $G20$, split the hyperarc which represents V_i with either of the two couples of sets, merge the nodes F_j and F_k;
- In $G10$, split the hyperarc with the edge sets $(E_j, ..., E_k)$ and $(E_l, ..., E_t)$ derived from $G21$, which represents V_i.

Also, G_{D_i} can have a more complex structure with loops connected to each other and possible dangling edges, then the splitting operation can take place either at a common node between two or more loops or between loops and dangling edges. Further insight about these configurations is described in the next section.

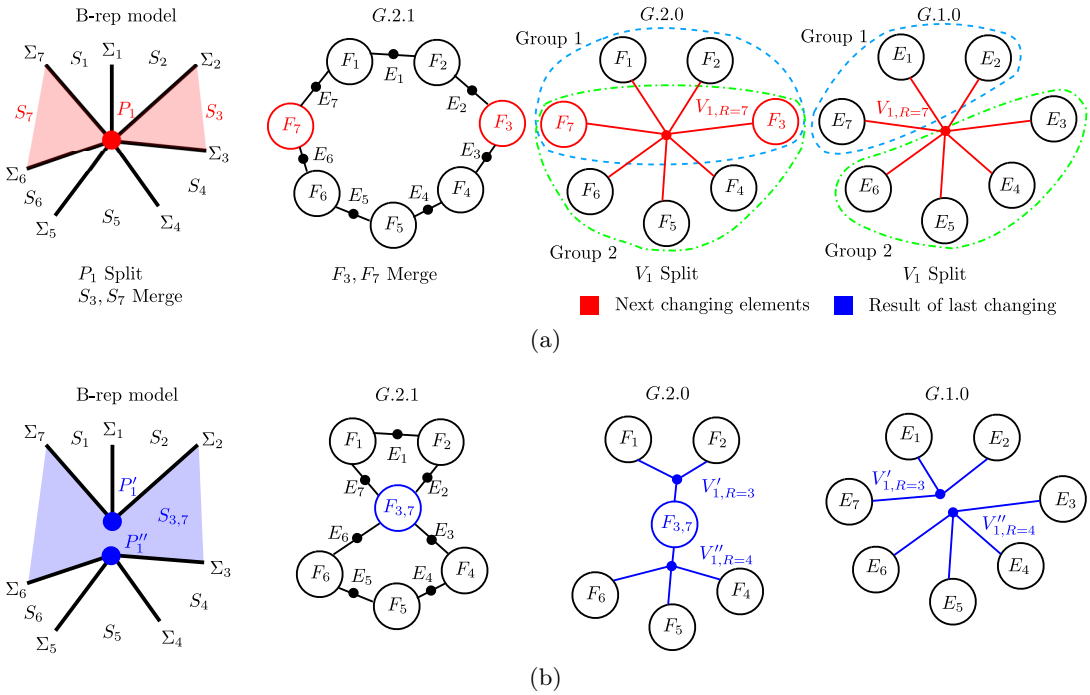


Figure 4.17: An illustrative example of a manifold vertex splitting process: (a) the initial configuration at V_1, (b) the new configuration after splitting V_1.

4.5.7 Maximal face and maximal edge merging criteria

The reason for the maximal face and edge preparation is that geometric modelers, as well as file exchange standards, produce B-Rep decompositions subjected to topological criteria (see section 2.3) that differ from those needed for symmetries and integrate also the designer's construction process, e.g. a surface of revolution or a circular boundary can be divided into pieces. All these surfaces are adjacent to each other to produce a closed and C^∞ boundary since the surface is identical on both sides of the edges subdividing the same geometric entity. But this subdivision may take place at arbitrary locations, hence loosing symmetries when the object is analyzed from its B-Rep decomposition, M_I. From the symmetry property point of view, the point sets forming ∂M_I should be as large as possible, i.e. these subdivided surfaces need be merged. This is the concept of maximal surfaces and maximal curves

corresponding to maximal faces and edges as topological entities.

From the topological point of view, in a 2-manifold B-Rep model M_{MI} defined by a set of faces $[F]$, $i \in \{1, ..., n\}$, each edge is strictly connected to two faces. Each face F_i is bounded. Without boundary constraint, considering only the reference surface intrinsic parameters and its location associated with F_i, there may exist faces adjacent to F_i whose reference surfaces are identical to that of F_i. Let F_j be one these faces, then F_i and F_j are called *homologous faces*. Because this concept refers to the reference faces of F_i and F_j, they are noted $\widetilde{F}_i$ and $\widetilde{F}_j$, respectively. These faces being homologous to each other, it is noted $\widetilde{F}_i = \widetilde{F}_j$.

Consequently, the maximal face generation process scans all the edges of M_{MI}. When detecting that a pair of adjacent faces are homologous, the hypergraph operator "Face merging" is triggered to merge them. The criterion for adjacent face merging can be simply stated:

Criterion 4 *If two adjacent faces F_i and F_j share the edge E_k and are homologous: $\widetilde{F}_i = \widetilde{F}_j$, these two faces must be merged and E_k is deleted.*

To identify the homologous faces, surface intrinsic parameters comparison criteria are needed. The method for intrinsic parameters comparison contains three parts:

- do S_i and S_j belong to the same type of reference surface?
- are the geometric parameters of S_i and S_j identical?
- are the location of S_i and S_j coincident?

The geometric parameters of surfaces have been introduced in section 3.2 to get more insight into the above conditions. When grouping the reference surfaces, the homologous faces criterion can be stated follows:

Criterion 5 *Considering two planes F_i and F_j, if their respective normals $\vec{n}_i$ and $\vec{n}_j$ are colinear and their reference point P_i lies in F_j and reciprocally for P_j, these planes coincide.*

Criterion 6 *Two revolution surfaces (cylinders, cones, tori) F_i and F_j, are homologous if their axes coincide. The radii of cylinders, the angle of cones and the two radii of tori must be identical.*

Criterion 7 *Two spheres F_i and F_j, are homologous if their centers coincide and their radii are identical.*

Regarding orientation, at any point of the common edge between F_i and F_j, their tangent planes coincide. Hence, their respective normals $\vec{n}_i$, $\vec{n}_j$ can produce two configurations only $\vec{n}_i \cdot \vec{n}_j = \pm 1$. As defined by the existing orientation of M_{MI}, these normals define its interior/exterior. Having $\vec{n}_i \cdot \vec{n}_j = -1$ would prescribe interior, resp. exterior, partitions on two opposite sides at the same point (see Figure 4.18), which is not possible. $\vec{n}_i \cdot \vec{n}_j = 1$ derives from the orientation of M_{MI}, which is the only solution left, hence no operation needs to take place about orientation.

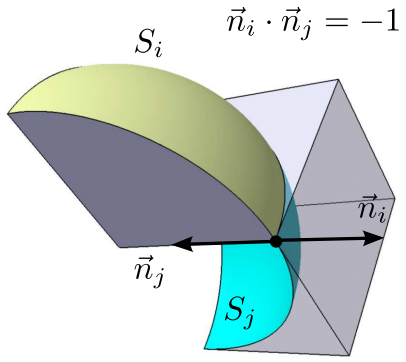


Figure 4.18: Opposite orientations of adjacent faces cannot happen when they are homologous to each other.

When finishing the face merging process, the set of maximal faces $[F_{Mi}]$, $i \in \{1, ..., m\}$, $m \leqslant n$, has been generated because all the edges of M_{MI} have been visited and the merging process is strictly increasing the area of surfaces. Consequently, at every edge left, its adjacent faces have different intrinsic parameters, i.e. these edges characterize a curvature discontinuity in ∂M_{MAX}.

The next step is the maximal edge generation. Between two adjacent maximal faces F_{Mi} and F_{Mj}, the intersection curve contains a set of points on the open interval defined by the two extreme points of this curve. This property derives from the property of a B-Rep NURBS modeler where every point P_k of each edge E_k of M_{MI}, bounded by vertices V_r, V_s, is the image of a unique intersection point between F_{Mi} and F_{Mj}; except at V_r and V_s, i.e.:

$$\begin{aligned} w &\xrightarrow{f} (u_i, v_i) \\ \vec{P}_k = \vec{P}_k(w) &= \vec{P}_i(u_i, v_i) \text{ unique} \\ w &\xrightarrow{g} (u_j, v_j) \\ \vec{P}_k(w) &= \vec{P}_j(u_j, v_j) \text{ unique} \end{aligned}$$

where f, g are bijections. Vertices V_r, V_s can be the locations of singular points such

that:

$$\exists w_1, w_2 \ / \quad \vec{P}_k(w_1) = \vec{P}_i = \vec{P}_j = \vec{P}_k(w_2),$$

see Figure 4.19.

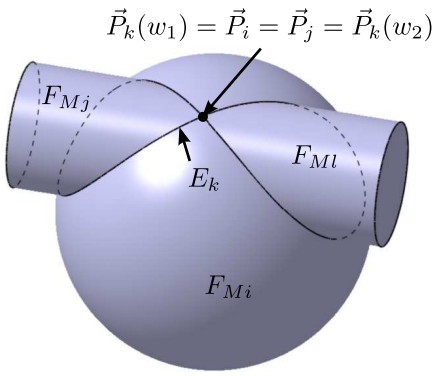


Figure 4.19: A configuration of intersection curve with a singular point $\vec{P}_k$. $\vec{P}_k$ can be the location of two vertices V_r, V_s. The parametric representation of E_k contains two coinciding points $\vec{P}_k(w_1)$ and $\vec{P}_k(w_2)$.

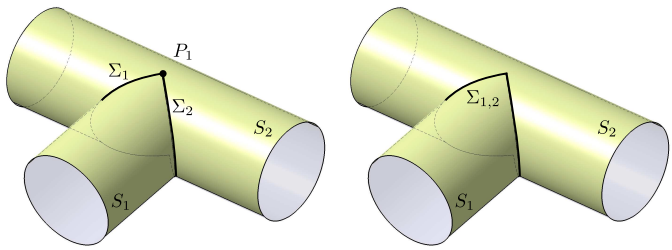
After the maximal face generation process, if a vertex is only connected to two edges, E_k, E_l, surely it lies on the intersection between faces F_{Mi} and F_{Mj}. E_k and E_l are continuous and their common vertex V_r is assumed to be a regular intersection point. Consequently, V_r must be deleted. The criterion to merge adjacent edges in this configuration states:

Criterion 8 *After surface merging, if the rank of a hyperarc A_i in $G10$ equals or less than 2, the corresponding vertex V_i must be removed and the two nodes representing its adjacent edges E_k, E_l must be merged.*

Because V_i may not be a regular intersection point between F_{Mi} and F_{Mj} when $\widetilde{F}_{Mi}$ and $\widetilde{F}_{Mj}$ are tangent to each other at V_i (see Figure 4.20a). Indeed, V_i satisfies criterion 8 and E_k, E_l form a C^0 curve but V_i is singular and this curve is not C^1 (see Figure 4.20b), which may influence the symmetry analysis of M_{MI}. A configuration of tangent reference surfaces produces singular intersection points and two alternatives can be proposed to process adjacent edges with such vertices:

- Add a geometric criterion to merge edges when C^1 continuity is ensured at their common vertex, i.e. V_1 in Figure 4.20b would not be removed;
- Leave Crit. 8 as it is while ensuring the proper generation of candidate symmetry planes.

Indeed, both of them can be used and will be discussed in chapter 5. In the present chapter, Crit. 8 is left as it is, which produces the largest possible edges.



(a) (b)

Figure 4.20: A continuous edge with a sharp point after the maximal edge generation: (a), P_1 connects to two curves, (b) from a topological point of view, $\Sigma_{1,2}$ forms a single edge $E_{1,2}$.

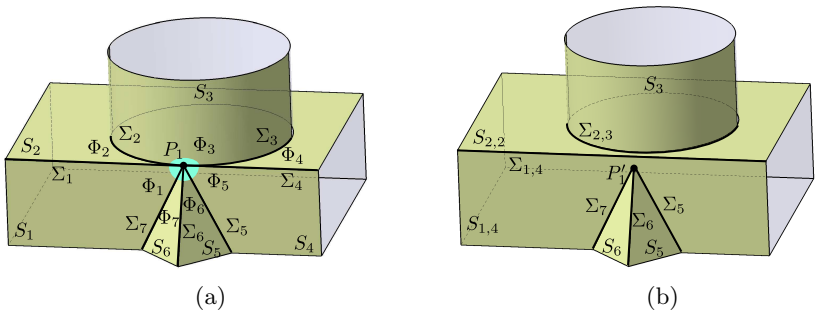
4.6 Vertex splitting at regular and non-regular points

In order to generate the maximal boundary of M_{MI} containing the maximal point sets for symmetry analysis, Crit. 4 and Crit. 8 are necessary conditions but not sufficient. When a set of faces $[F_{\Omega V_i}]$ shares the same vertex V_i, such as in a tangent surfaces configuration, V_i can connect with more than two edges (see Fig 4.22a). According to the curves defining the edges attached at V_i, maximal edges could be derived from these edges but their continuity is somehow broken by V_i. Within $[F_{\Omega V_i}]$ couples of surfaces $(\widetilde{F}_1, \widetilde{F}_4)$ and $(\widetilde{F}_3, \widetilde{F}_5)$ are homologous, but because they only intersect at a vertex V_1, Crit. 4 cannot be applied.

Figure 4.21a is an illustration of another configuration and is also analyzed as a preliminary example. The curves $(\Sigma_2 \cup \Sigma_3)$ and segments $(\Sigma_1 \cup \Sigma_4)$ are tangent to each other at P_1. Their common vertex V_1 breaks their continuity. As of the rest of the boundary decomposition, local symmetry properties are not correctly represented and this model representation is still not describing a maximal boundary decomposition. In this example, the cylinder surface S_3 is not axisymmetric because of vertex V_1. In order to follow the maximal edge hypothesis, Σ_2 and Σ_3 should be merged into one curve forming an edge without a vertex, Σ_1 and Σ_4 should be merged also as one segment. This requires a manifold vertex splitting operation. Vertex V_1 will be split into three vertices and the faces $(\widetilde{F}_1, \widetilde{F}_4)$ can be merged as well as $\widetilde{F}_3$ can benefit of axisymmetry property. Then, the edge merging operation can be applied. In the end, the topological structure conforms to Figure 4.21b where the structure of $\widetilde{F}_2$ has been modified from one bounding loop to two loops.

The two above examples show that singular intersection points, faces with bounded

by tangent loops and homologous faces connected to each other through a vertex characterize the criteria of the manifold vertex splitting operation.



(a) (b)

Figure 4.21: An example configuration where curves bounding surfaces are tangent to each other. The continuity of the corresponding edges is broken at P_1: (a) the initial B-Rep model M_{MI}, (b) the maximal boundary model M_{Max}.

The interesting thing is that not all the similar cases produce a manifold vertex split operation. For example, in Figure 4.22, the local structure looks similar to Figure 4.21a. Indeed, when analyzing the faces surrounding point P_1, the cylindrical faces F_{M3} and F_{M5} are homologous. P_1 connects with S_2 twice, i.e. on both sides of P_1. Obviously, F_{M2} is homologous to itself. Also, the planar surfaces F_{M1} and F_{M4} are homologous. Now, the difference between Figure 4.21a and 4.22a is that the pairs of homologous surfaces in Figure 4.22a are crossing each other, i.e. (S_3, S_5) 'crosses' (S_1, S_4), (S_3, S_5) 'crosses' (S_2, S_2) when considering their adjacencies around P_1. If V_1 is split and some pairs of homologous surfaces are merged, the continuity properties of other surface pair(s) cannot be achieved. Currently, there is no criterion available to select a pair of homologous faces to be merged or to be broken (see Figure 4.22b and c). Because the vertex splitting operation is not unique in the present configuration and there is no criterion appearing to justify the choice of one solution, no vertex split should take place. Consequently, the point P_1 is called a *non-regular vertex* and if the point could be split it is called a *regular vertex*. To distinguish the regular and non-regular configurations at vertices, the definition is as follows:

Definition 3 *Considering a vertex V_α of M_{MI}, V_α is surrounded by n ordered sectors Φ_i and n ordered edges η_i, $i \in [1, n]$, $n \geqslant 3$, based on the adjacency relationships between the edges attached at V_α. A sector of a face F_{Mj} is formed by two adjacent edges of F_{Mj} whose common vertex is V_α. Based on the input model M_{MI}, there exists an order of sectors available from the real B-Rep datastructure of M_{MI} that defines the disk around V_α as part of the B-Rep model, i.e. its orientation is consistent with that of the B-Rep. Because the sector and edge numberings differ for each one, a dual graph based on Φ_i and η_i, restricted to V_α, can be generated that forms a simple loop around V_α.*

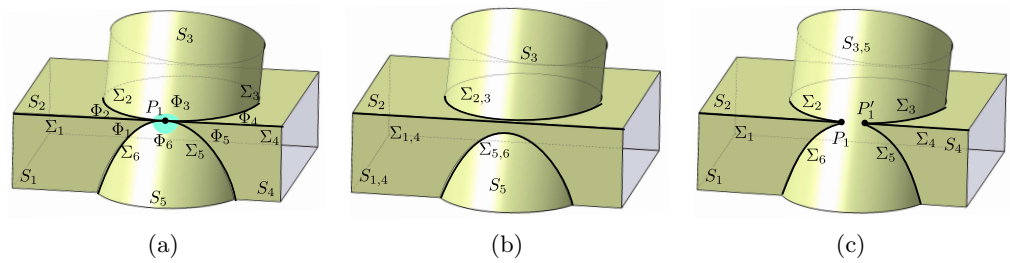
Now, let us consider that there exists at least a couple of homologous sectors: $\exists(\Phi_i, \Phi_j) \quad /\widetilde{\Phi}_i = \widetilde{\Phi}_j$, $(i, j \leqslant n)$, $i < j$.

If there exists at least another couple of homologous sectors: $\exists(\Phi_k, \Phi_q) \quad /\widetilde{\Phi}_k = \widetilde{\Phi}_q$, $(k, q \leqslant n)$, (Φ_i, Φ_j) *and* (Φ_k, Φ_q) *are said to cross each other if* $i < k < j$ *and* $j < q \leqslant n$ *or* $1 \leqslant q < i$. *This configuration is qualified as* ***non-regular*** *configuration at vertex* V_α *and* V_α *cannot be split for* (Φ_i, Φ_j) *and* (Φ_k, Φ_q).

If there exists one or more homologous sectors only: $\exists(\Phi_k, \ldots, \Phi_q) \quad /\widetilde{\Phi}_k = \ldots = \widetilde{\Phi}_q$, $k, q, \ldots > i$ *and* $k, q, \ldots < j$, $(\Phi_k, \ldots, \Phi_q)$ *are not crossing* (Φ_i, Φ_j) *and form a* ***regular*** *configuration at* V_α. V_α *can be split to assign its copy to the sectors* Φ_t, $t \in \{(i+1), \ldots, (j-1)\}$

Otherwise, the configuration at V_α *is said* ***neutral*** *because the vertex split is not meaningful for it.*

It has to be noted that each sector Φ_i *maps to one face* F_{Ml} *of* M_{MI} *but* F_{Ml} *may generate several sectors* Φ_i, Φ_j, ...



(a) (b) (c)

Figure 4.22: An illustration of non-regular vertex P_1: (a) the original B-Rep model M_{MI}, (b) and (c) are two possible maximal configurations.

Figure 4.22 is an illustration of non-regular configuration. In this figure, surrounding point P_1, $\widetilde{\Phi}_3 = \widetilde{\Phi}_6$, $\widetilde{\Phi}_1 = \widetilde{\Phi}_5$ and $\widetilde{\Phi}_2 = \widetilde{\Phi}_4$. (Φ_3, Φ_6) is crossing (Φ_1, Φ_5) as well as (Φ_2, Φ_4). Hence, this is a non-regular configuration at P_1.

Back to the example in the Figure 4.21. $\widetilde{\Phi}_1 = \widetilde{\Phi}_5$ and $\widetilde{\Phi}_2 = \widetilde{\Phi}_4$, but both of them are not crossing each other. Hence, this example is a regular configuration at vertex P_1.

Given this regular configuration at point P_1, the face merging process is applied as follows. Its surrounding faces can be detected easily. In Figure 4.24b, the corresponding sectors are $(\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6, \Phi_7)$. Because sectors Φ_2 and Φ_4 are homologous as well as Φ_1 and Φ_5 and these couples are not crossing each other, vertex V_1 can be split and face F_{M1} can be merged with F_{M4}. Also, the splitting operation removes the tangent loop in F_{M2}.

Based on the previous qualitative analysis, the purpose is now to formalize the

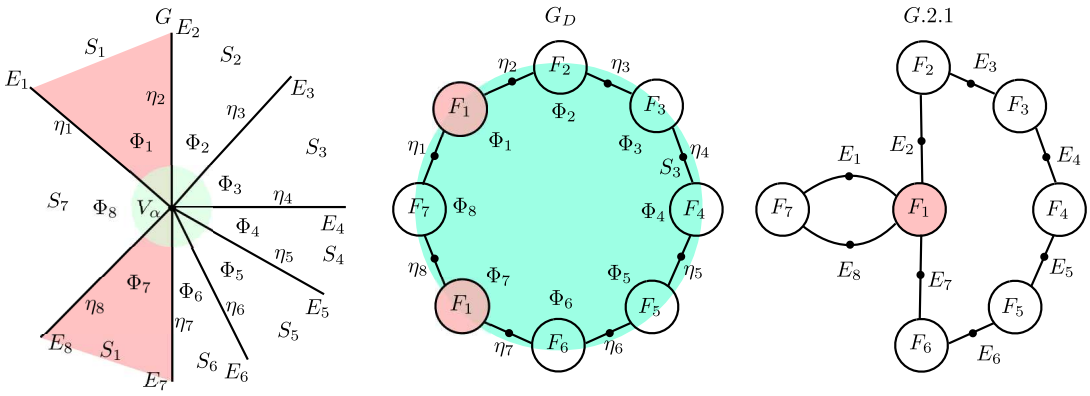


Figure 4.23: Dual graph based on sectors Φ_i at V_α and corresponding transformations.

characterization of the configurations and set the corresponding operators. Starting from the distinct sectors mentioned in definition 3 that form a simple loop, the purpose is to connect the transformations of this loop through the homologous sectors to the maximal faces of M_{MI}. To this end, let us consider any two homologous sectors $\tilde{\Phi}_i$, $\tilde{\Phi}_j$, $i \neq j$. Because they are homologous to each other, this property can be expressed in the simple loop by merging Φ_i and Φ_j to produce $\Phi_{i,j}$. Consequently, this simple loop becomes two tangent simple loops at the node $\Phi_{i,j}$ if $|i-j| > 1$ (see Figure 4.23): $\Phi_{i,j}$, Φ_{i+1}, $\ldots$, Φ_{j-1} and $\Phi_{i,j}$, Φ_{j+1}, $\ldots$, Φ_n, Φ_1, $\ldots$, Φ_{i-1}. Indeed, $|i-j| > 1$ holds because Φ_i and Φ_j cannot be adjacent otherwise they would belong to the same maximal face, which contradicts the fact that every sector belongs to a maximal face. Now, let us consider the following consequences:

a If Φ_i and Φ_j belongs to the same maximal face F_{Mk}, this merging operation sets a one to one mapping between $\Phi_{i,j}$ and F_{Mk};

b If two of the edges η_t attached to Φ_i and Φ_j are mapped to the same edge E_p of M_{MI}, then these edges η_r and η_s may be merged to set a one to one mapping with E_p. If η_r and η_s connect to the same sector of index such that $i+1 = j-1$, i.e. there is only one sector between Φ_i and Φ_j, then η_r and η_s can be merged and form a dangling arc to connect $\Phi_{i,j}$ to Φ_{i+1}. Otherwise, if the sectors Φ_k, Φ_l connecting to η_r and η_s are mapped to non homologous faces F_{Mp}, F_{Mq}, these edges cannot be merged since merging would produce a hyperarc, which contradicts the fact that the neighborhood of V_α is manifold.

Assuming that all homologous sectors $\tilde{\Phi}_i$, $\tilde{\Phi}_j$ producing a one to one mapping between $\Phi_{i,j}$ and F_{Mk} are effectively merged and that all edges η_i that can be merged

are effectively merged, the resulting graph structure has a one to one mapping between each of its nodes and a face F_{Mk} and a one to one mapping between each of its arcs and an edge E_p. Consequently, the resulting graph (Φ, η) is isomorphic to a subset of hypergraph $G21$ restricted to the faces and edges connected at V_α: $G21_{|V_\alpha}$. Because of this isomorphism, the subsequent analysis is performed on $G21_{|V_\alpha}$. Based on the B-Rep datastructure of M_{MI}, it can be assumed that the simple loops tangent to each other and the possible dangling arcs are available to structure $G21_{|V_\alpha}$.

Now, repeating the face merging process for any set of m homologous faces F_{Mp}, F_{Mq}, ..., in $G21_{|V_\alpha}$ produces new loops and two configurations may occur:

- All the homologous faces belong to the same simple loop, then the merge operation creates m new simple loops only from this initial loop;
- Homologous faces belong to different simple loops, then the merge operation creates m new loops among which there can be simple loops and at least one connection between existing simple loops.

Applying a vertex split operator to this graph structure transforms it as follows: the split has to be associated with a face F_{Mk} of $G21_{|V_\alpha}$ and is submitted to the configurations regarding its surrounding b edges E_i:

- If there exist h edges E_j, E_l, ... whose ranks in $G10$, $R_{j-G10} = R_{l-G10} = \ldots = 1$, i.e. these edges are loop edges defined by a single vertex, they may generate up to h split operations of F_{Mk};
- Either $h = 0$ and b is even or $h \geqslant 0$ and $b - h$ is even, then E_i generates either $d = b/2 - 1$ or $d = (b + h)/2 - 1$ split operations, respectively.

Then, the effect of the split is the d-times duplication of F_{Mk} whose rank is R_{Nk}: $F_{Mk} \rightarrow (F^1_{Mk}, \ldots, F^d_{Mk})$ and the connection of each F^j_{Mk} with a subset of E_i to terminate each elementary split. Then, each termination of elementary split with an increment of the number of components in $G21_{|V_\alpha}$ characterizes a `regular` configuration and each elementary split without this increment is not terminating and characterizes a `non-regular` configuration. A completed elementary split creates a new component in $G21_{|V_\alpha}$ because it is assigned to the new copy of F_{Mk}, showing that the neighborhood of this new vertex can be independent of that of V_α, which is the purpose of the splitting mechanism. Similarly, a non terminating elementary split reflects a configuration where no independent neighborhood can be found in V_α that can be assigned to a new vertex. Hence, the vertex split criterion can be stated as follows:

Criterion 9 *Having merged all the sets of homologous faces connected at vertex V_α, the hypergraph $G21_{|V_\alpha}$ restricted to the neighborhood of V_α is transformed with new*

connections. The type and number of edges E_i connected at V_α generate d candidate elementary vertex split at nodes F_{Mk}. The ith elementary vertex split characterizes a `regular` *configuration of F_{Mk} at V_α if it increments by one the number of connected components in $G21_{|V_\alpha}$. As a result, the homologous faces merged to generate this configuration can be effectively used to generate a independent neighborhood for the new vertex generated by the elementary split.*

If an edge E_i is a loop edge: $R_{i-G10} = 1$ and if it belongs to a dangling edge or, more generally, a branch of $G21_{|V_\alpha}$, any copy of F_{Mk}: F^j_{Mk} connects to E_i, is assigned a rank $R_{N^j_k} = 1$ and **creates a new component** in $G21_{|V_\alpha}$, which really terminates this elementary split. If E_i is a loop edge and belongs to a loop of $G21_{|V_\alpha}$, a copy F^j_{Mk} connects to it and the new content of $G21_{|V_\alpha}$ is scanned to look for an increment in the number of components. If its number of components has not been modified, this elementary split is not performed and characterizes a non-regular configuration. Repeating this process $h-1$ times if $b = 0$ or h times if $b > 0$ leaves $b-h$ edges to connect to copies of F_{Mk}.

Based on the current content of $G21_{|V_\alpha}$, each copy $F^{(h+1)}_{Mk}$ or F^h_{Mk}, ... of F_{Mk} left is connected to two edges of E_i in accordance with the loop structure associated with $G21_{|V_\alpha}$. If this copy and its new connections in $G21_{|V_\alpha}$ create a new component, this elementary split terminates and characterizes a `regular` configuration. Otherwise, the configuration is `non-regular`. If the configuration is non-regular, some of the homologous face merge are no longer necessary because they are not meaningful. Therefore, it is necessary to restore these loops to obtain a consistent face decomposition. To this end, let us first consider that all the nodes candidate to the split operation have been processed, extracting components from the initial $G21_{|V_\alpha}$. Then, the component left in the final version of $G21_{|V_\alpha}$ can contain nodes that were connected to simple loops in the initial $G21_{|V_\alpha}$. Consequently, the purpose of the transformation of the current version of $G21_{|V_\alpha}$ consists in splitting all the nodes connecting multiple loops to produce simple loops as large as possible but preserving the connections between loops that were existing in the initial version of $G21_{|V_\alpha}$.

Once all the regular and non-regular configurations have been identified through the splitting process and the non-regular configurations have led to split face nodes resulting from unnecessary face merge operations, it forms a subset $G21_0$ of the transformed $G21_{|V_\alpha}$. Then, the graph components generated for each regular configuration needs to be connected again to $G21_0$ to obtain the correct version of $G21_{|V_\alpha}$ to be inserted in $G21$ (see Figure 4.24 as an example of this process).

Now, coming back to examples, focusing on vertex V_1 in the configuration depicted at Figure 4.21, Figure 4.24a, b gives the hypergraph $G21_{|V_1}$ as well as $G20_{|V_1}$ providing the set of faces surrounding V_1 and $G10_{|V_1}$ describing the set of edges surrounding V_1, as starting point of the face merging and vertex split processes. Figure 4.26a

illustrates the underlying connection between $G21_{|V_1}$ and the initial dual graph based on sectors. Figure 4.24d illustrates the result of the face merge step in $G21$ and $G20$ where F_{M1} and F_{M4} are homologous to each other and become $F_{M1,4}$. $G10$ represents the configuration when $F_{M1,4}$ is being processed for splitting. The set of edges E_i connected to $F_{M1,4}$ does not contain loop edges ($h = 0$) resulting in $d = 1$ elementary splits. Figure 4.24c, illustrates the result of this elementary vertex split producing V_1'. Figure 4.26b contains the new structure of $G21$ showing the component created with V_1'. Figure 4.26b contains also the transformation attached to the next candidate face in $G21$, i.e. F_{M2}. F_{M2} is a pre-existing node in $G20_{|V_1}$ producing tangent loops. There $h = 0$ and $d = 1$, $G20$ and $G10$ show the results of the corresponding elementary split: V_1''. Figure 4.26b is the graphical representation of the split generating V_1''. Finally, Figure 4.26b summarizes the final configuration after performing all the vertex splits and subsequently the edge merge operation resulting in $E_{M1,4}$ and $E_{M2,3}$. Then, Figure 4.25d represents the new local content of $G21$, $G20$, $G10$ to be inserted in these hypergraphs. It can be noticed that F_{M2} and F_{M3} are now disconnected from $F_{M1,4}$ in $G20$ but will be connected to other surrounding faces. $E_{M2,3}$ becomes an edge without vertex and $E_{M1,4}$ will be connected to extreme vertices of E_1 and E_4.

Comparing Figure 4.26a and b shows that the orientation of M_{MI} contained in the initial dual graph at V_1 cannot be conveyed in the hypergraph $G21$ resulting from the face merging process. However, this orientation is not mandatory for performing the face merge and vertex splits, which is a property of this process even though it is not formally demonstrated here.

Because F_2 is homologous with itself. The next work is to "merge" them and in this example the vertex V_1 of Figure 4.21 can be split again. The last step is to merge the edges using criterion 8, because the previous steps may create some vertices connected only to two edges (see Figure 4.25d). All the operations in the hypergraphs follow the hypergraph basic editing operations described at section 4.5.1.

Another example configuration is illustrated in Figure 4.28a with a set of eight faces around V_1. In this case, the homologous faces form a larger set $\widetilde{F}_1 = \widetilde{F}_3 = \widetilde{F}_5 = \widetilde{F}_7$ (see Figure 4.28b). Within the set of faces, there is no other homologous faces. Regarding the criterion 9, after merging these four faces (see Figure 4.28d), the face $F_{M1,3,5,7}$ is the only candidate face for vertex split with $h = 0$ and $d = 3$. In the present configuration, each elementary split in $G21_{|V_1}$ produces a new component, thus characterizing a regular configuration. Figure 4.28c and d respectively, show the resulting vertex configuration and the effect of the splits in $G20$ and $G10$.

Finally, the configuration of Figure 4.22 is used to illustrate how $G21_{|V_1}$ is transformed into a configuration where $F_{M3,5}$ and $F_{M1,4}$ are candidate nodes for the vertex split operator but this operation cannot terminate, highlighting the non-regular configuration (see Figure 4.27).

Through the previous sections, all the processes and criteria have been clarified to

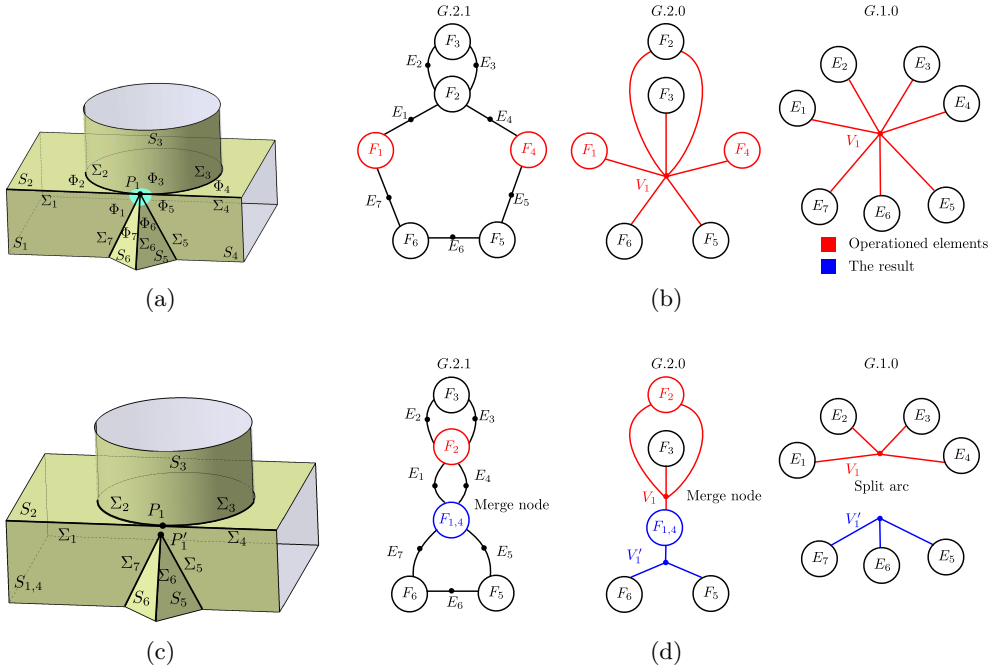


Figure 4.24: The successive steps to process homologous faces at a regular vertex through merging operations and vertex split. The final edge merging operation is also illustrated (first subset).

generate a boundary decomposition from the initial B-Rep model M_I that is suited for its symmetry analysis. After some processing depending on the conformity of M_I with respect to a 2-manifold model, M_I can be transformed into M_{MI} with the non-manifold vertex split. Then, the boundary decomposition of M_{MI} is modified to meet the symmetry analysis requirements. To this end, the maximal faces are generated with the face merge operator to remove edges between homologous and adjacent surfaces. Maximal faces are further extended with the merging process of faces adjacent to each other through a vertex using a manifold vertex split. The resulting face neighborhoods are extended around their common vertex. Finally, maximal edges are produced through a vertex removal operation between adjacent edges. The resulting model is M_{MAX} that has a boundary decomposition intrinsic to the object shape and operational for symmetry analysis. The overall process enumerated previously is summarized in Figure 4.29.

It has to be reminded that the boundary description M_{MAX} contained in the hypergraphs has no effect on the geometric description of the object. All the transformations performed are strictly topological and have no influence on the surfaces, curves and points describing the object. Further, the datastructure set up estab-

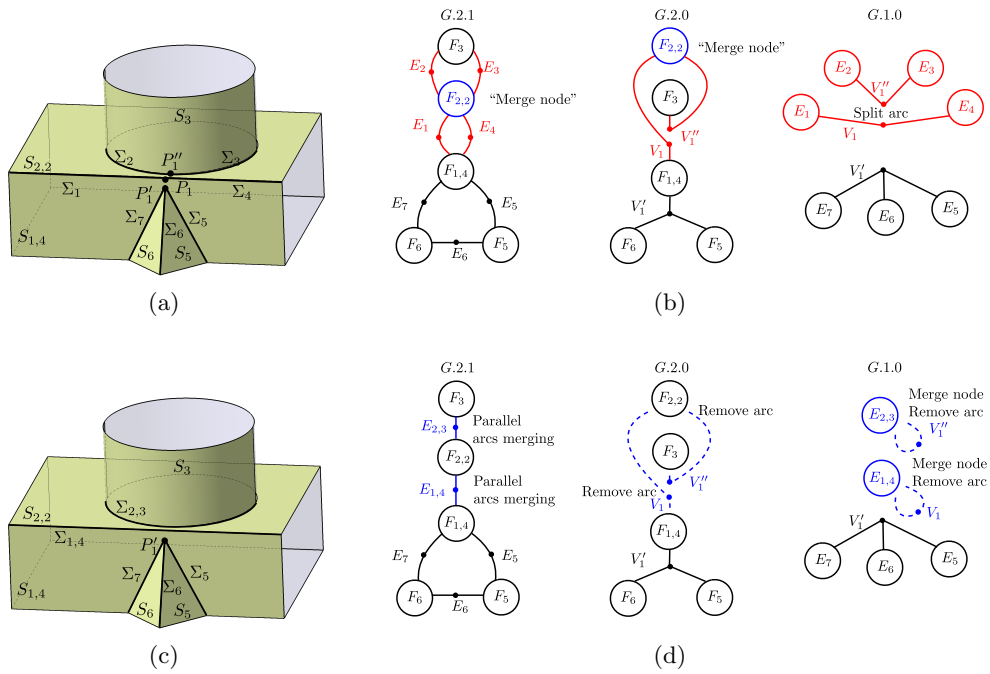


Figure 4.25: The successive steps to process homologous faces at a regular vertex through merging operations and vertex split. The final edge merging operation is also illustrated (second subset).

lishes a connection between the B-Rep CAD topological description of the object and the hypergraphs structuring this object boundary in accordance with the symmetry requirements. Consequently, the hypergraphs content combined with the geometric data describing the object can be seen as a view of the object devoted to its symmetry analysis. In the framework of a PDP, it is effectively a view of the object similarly to the approach proposed by Foucault et al. [18] devoted to mesh generation constraints for finite elements simulations.

Finally, for the sake of simplicity, maximal faces and edges respectively noted F_{Mi} and E_{Mj} in this chapter will be noted as F_i and E_j in the following chapters since all the subsequent treatments for symmetry analysis will be performed on an object decomposition using the maximal faces and edges.

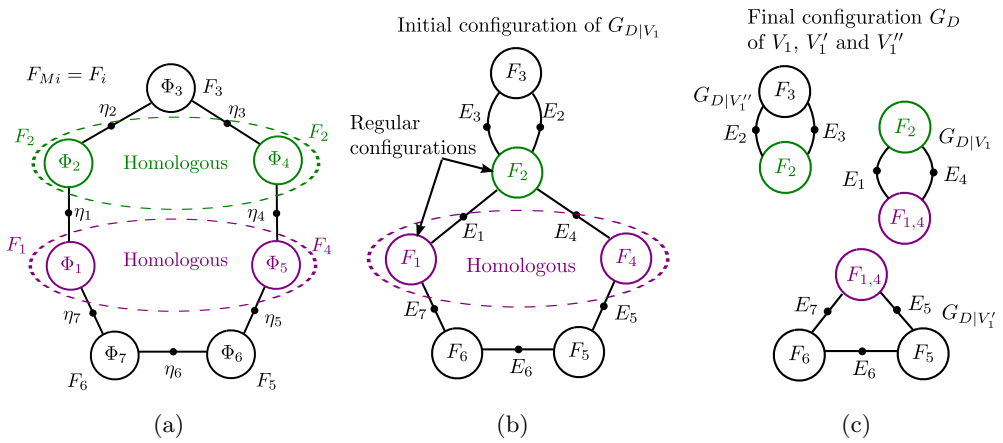


Figure 4.26: The dual graph from Figure 4.24b in $G21_{|V_1}$ with its surrounding sectors: (a) initial dual graph showing that the merge faces produce regular configurations, (b) resulting graph after the face merging processes.

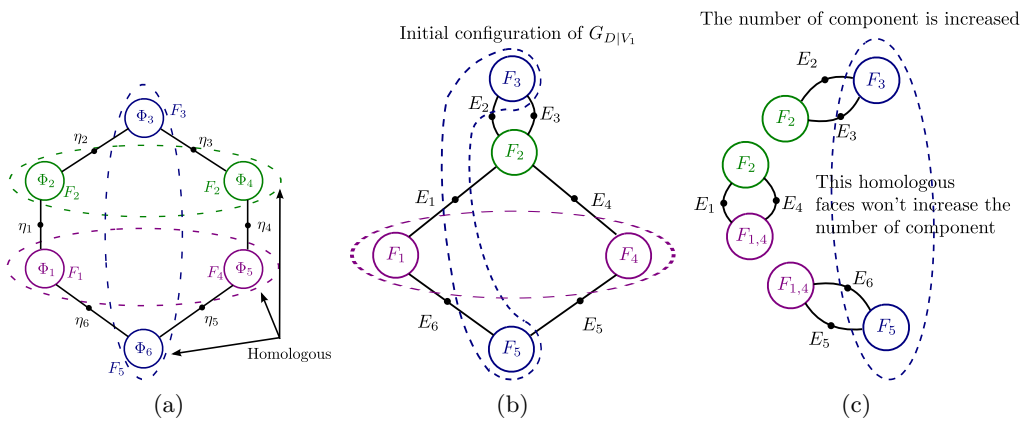


Figure 4.27: Graph transformations of $G21_{|V_1}$ based on Figure 4.22: (a) the status of homologous faces; (b) the homologous configurations of faces showing that the split vertex operator; (c) one homologous faces won't split the graph.

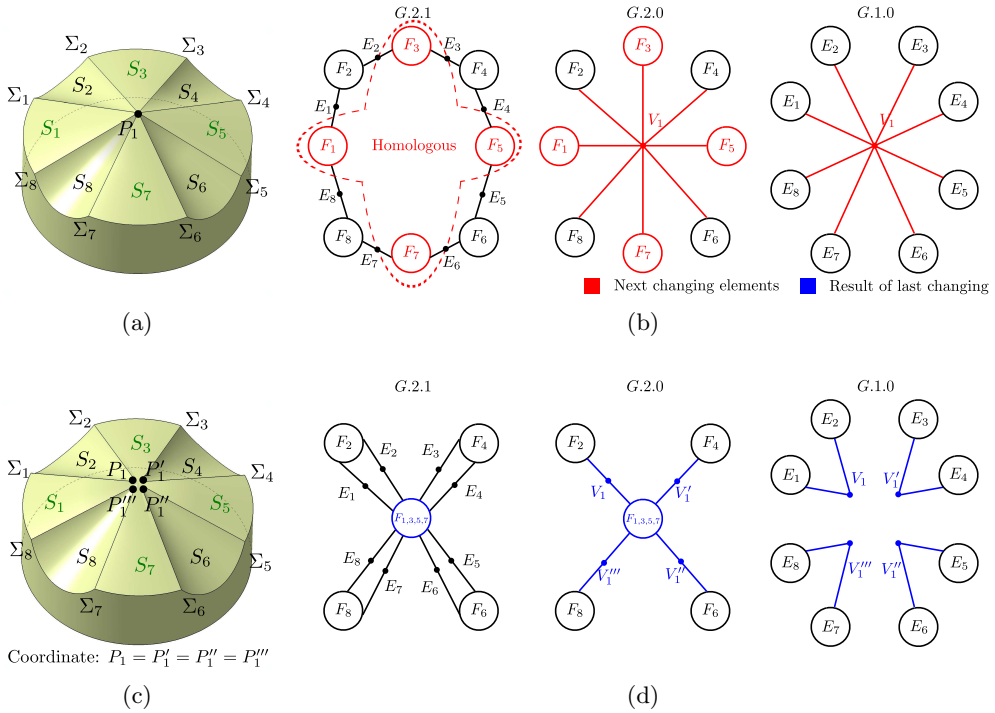


Figure 4.28: A special case of non-regular vertex: (a) The B-Rep model; (b) local hypergraph around the regular vertex V_1; (c) the result after splitting; (d) the result of local hypergraph.

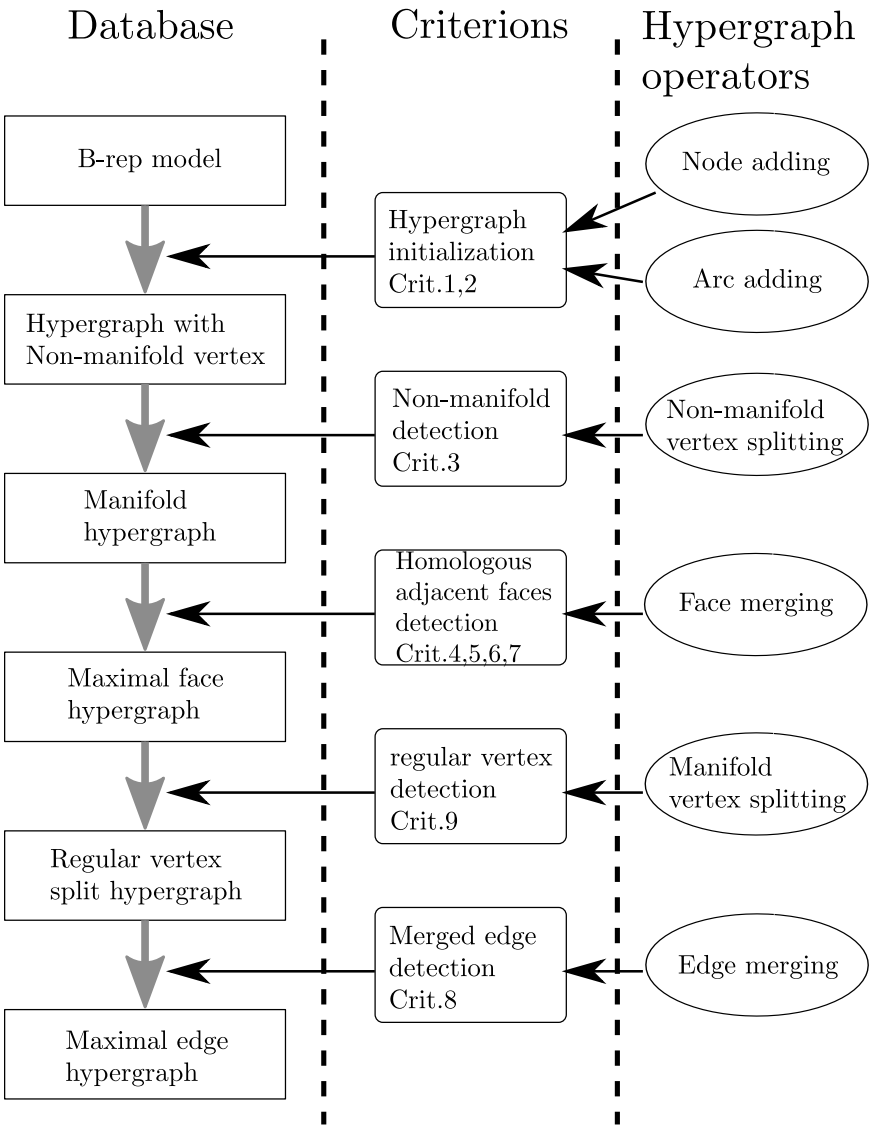


Figure 4.29: The process flow of creating hypergraphs describing a maximal boundary decomposition from an initial standard B-Rep model M_I. The resulting model is M_{Max}, which is used for subsequent symmetry analysis.

4.7 Conclusion

Starting form a B-Rep model conforming to the STEP format to ensure a large compatibility with industrial modelers, this chapter has described how the transformation of the boundary ∂M_I of the input model M_I can be achieved to meet the requirements for the symmetry analysis of M_I.

The symmetry analysis relying on a global approach using infinite point sets, the surfaces and curves forming ∂M_I are subjected to modeler constraints related to topological properties needed to characterize a volume. Consequently, faces, edges and vertices needed for this purpose differ from the requirements of the symmetry analysis where the topological description of faces, edges and vertices have cross influences on the boundaries of the infinite point sets covering M_I, i.e. prescribing edges bounded by two vertices generate bounds on closed curves that can remove their axisymmetry properties.

Similarly, a modeling process as conducted by a designer during a PDP impacts the object boundary with a face, edge and vertex decomposition that does not necessarily meets the decomposition needed for its symmetry analysis, i.e. splitting an edge subdivides a curve whose symmetry planes attached to its pieces may differ from that of the entire curve.

Hypergraphs have been the basic tools for topology description that have been used to describe an object boundary that is intrinsic and meets the symmetry analysis requirements. The particular structure of edges defined with one or no vertex at all are among these requirements to preserve the symmetry properties of the object and justify the use of hypergraphs.

Then, the proposed operators aimed at transforming an object into a 2-manifold if it has no impact on its shape, as perceived by a designer and, most important, they are devoted to the generation of the object boundary decomposition into the largest possible areas of infinite point sets spanning each dimension of entities describing the object boundary, i.e. faces, edges, vertices. This means that each operator and its associated criteria:

- Produce a larger point set than each of its initial operands. Merging faces fits into this category to extend the associated surfaces embedded in the object boundary and merging edges is similar for curves;
- Remove a point set of lower dimension to effectively modify its boundary. Merging faces can be associated to the edge removal or to the co-called manifold vertex split that merges also faces while removing their common vertex.

All these operators generate the M_{Max} model of the analyzed shape, which forms the input for the divide and conquer process used to analyze the symmetry properties

of this shape. The next chapter addresses the 'divide' phase of the symmetry analysis with the study of the candidate symmetry planes that can be assigned to each face, edge or vertex of the object boundary and their assignment process.

Chapter 5

Symmetry planes of point sets defined from surfaces and curves

The algorithm of reflective symmetry analysis is based on the boundary decomposition model with maximal point sets, i.e. maximal surfaces and curves, and the corresponding data structure is now available in the hypergraphs of M_{Max}. They form the elements of ∂M_{Max} and their symmetry properties are analyzed and categorized. Here, a symmetry plane valid over the area of a maximal point set is named a *Candidate Symmetry Plane*(CSP). These semi-global areas are the smallest areas used to define symmetry planes. Several categories of CSPs are introduced depending on their attachment to vertices, edges or faces of the hypergraphs for point sets intersecting CSPs and loops forming point sets without intersection with CSPs. This forms the divide phase.

Loops are also analyzed for each reference surface and categorized to structure ∂M_{Max} and contribute to the symmetry analysis.

5.1 Introduction

Maximal faces and edges together with vertices define the point sets forming ∂M_{Max}. Each reference surface $\tilde{S}$ attached to a maximal face F_i possesses its own symmetry properties. Each of the edges E_j bounding F_i has also its own symmetry properties. These symmetry properties interact with each other to produce only a subset of symmetry planes valid for both. It is the purpose of this chapter to take into account these interactions to obtain the CSPs simultaneously valid for E_j and F_i, thus characterizing the extent of validity of each CSP.

Whatever, the ∂M_{Max} entity attached to a CSP Π_k, it is characterized by a set of intrinsic parameters and location of each of the faces F_i involved in the definition of Π_k. This avoids any reference to the intersection curve between adjacent faces F_i and F_l since this curve is not represented exactly in CAD models. Referring to their geometric model would necessitate taking into account the modeling accuracy and the approximations of a modeler. The locations of extrema would become much more difficult to determine and interpret. Referring to the surface parameters is not influenced by the modeler accuracy and approximations, they directly reflect the user's input parameters and are regarded as control parameters to modify the shape of M_{Max}.

In order to analyze the symmetries of M_{Max} and, later on, modify it to remove some asymmetries, it is important that the CSPs are assigned to the least possible number of ∂M_{Max} entities so they establish a meaningful link between each CSP and the intrinsic parameters and location of the faces F_i involved in its definition. This link will be helpful to define a reverse process to modify F_i parameters so that either existence or new position of the corresponding CSP increases the extent of M_{Max} symmetries.

Because symmetry properties of surfaces is not always characterized by a finite number of symmetry planes, it is also important here to distinguish geometric configurations producing finite of infinite numbers of CSPs. In the latter case, an infinite number of CPSs forms a category designated as *Candidate Symmetry Axis* (CSA).

Finally, edges E_j bounding F_i form necessarily one loop at least. If F_i is bounded by more than one loop, edges of these loops interact to form also CSPs and their structure needs also be analyzed to assign the correct CSPs forming the divide phase of the symmetry analysis of M_{Max}.

The above observations define the framework of the following analysis assigning CSPs and CSAs to the entities of ∂M_{Max}.

5.2 Reflective symmetry properties of the five reference surfaces

Here, we consider the so-called reference surfaces $\tilde{F}$ and state some of their properties related to reflective symmetry as a basis for their global properties that help defining their symmetry planes. It is the basis of the algorithm defining the symmetry planes of M_{Max}, as defined at chapter 6.

As it is part of a B-Rep model, a reference surface $\tilde{F}$ is simply connected and its area is finite. Regarding symmetry, the proposed approach addresses rectilinear symmetry axes, symmetry planes and symmetry with respect to a point. The reflective symmetry definition stated at chapter 1 is now restated in the scope of the present work where points belong to reference surfaces.

Definition 4 *: Self reflective symmetry of a face F_i*

Given an arbitrary point $X \in I\!R^3$ of a face F_i generated from a reference surface $\tilde{F}$ and a symmetry plane Π, there exists a unique point X_F such that X_F belongs to F_i and satisfies:

$$\overrightarrow{MX_F} = -\overrightarrow{MX} = -\|\overrightarrow{MX_F}\| \cdot \vec{n}$$

where F_i denotes a bounded maximal face, M is the normal projection of X on Π , $\vec{n}$ is the normal to Π at M. X_F is said to be the image of F_i, i.e. the symmetric point of X in F_i. Conversely, if any point X of F_i has a symmetric point X_F lying in F_i through Π, then Π is a symmetry plane for F_i.

The definition of reflective symmetry at a point applies also to space curves when F_i is substituted by a loop Γ, i.e. an arbitrary 3D curve either open or closed.

According to the framework of the proposed approach, the five reference surfaces $\tilde{F}$ form the basis of the faces F_i used as basic surface patches combined to form a B-Rep model and F contain infinite point sets. Each type of reference surface has its own symmetry properties. These symmetry properties provide the basic symmetry constraints and they provide the only symmetry possibilities if the model reduces to a unique reference surface defining a unique face F_1. These elementary symmetry properties form the starting point of the algorithm's conquer phase to analyze the global symmetry properties of the model. It is the first level of semi-global symmetry. The reference surface symmetry properties are intrinsic properties, they connect with the intrinsic parameters of this surface only.

For example, a plane P is an infinite surface. Whatever its boundary, when it is symmetric, the symmetry plane of the corresponding bounded surface has to follow the properties stated before, i.e. the symmetry plane Π is orthogonal to P. The cylinder and the cone behave similarly as the plane with the additional axisymmetry property. With an unbounded surface of revolution, its axis is important since it characterizes the fact that there is infinity of symmetry planes containing this axis. A bounded surface of revolution stays axisymmetric if it is limited by circular boundaries. Considering a cylinder Cy, if its boundary curve is symmetric with respect to a plane Π_1 orthogonal to Cy axis, Π_1 is still a symmetry plane. Based on the previous observations, the symmetry properties of the five reference surfaces $\tilde{F}$ can be stated as follows (see Figure 6.2):

- **Plane**: In $I\!R^3$, a plane F_p can be defined in different ways. All these definitions can be reduced to a point lying on F_p and a vector orthogonal to F_p, which defines its normal. With an unbounded plane F_p, any plane Π orthogonal to F_p is a symmetry plane for F_p (see Figure 6.2a): there is an infinite number of symmetry planes. Now, when F_p is bounded by a closed loop Γ, whatever its location in F_p, the number of symmetry planes becomes either finite or is still infinite if Γ is a circle. In the latter case, all the planes share a common property since they contain the center of the circle but if a symmetry plane Π exists with respect to Γ, Π has to be orthogonal with F_p. This shows that the effect of the loops or edges E_j reduce the number of symmetry planes, usually to a finite number, but these symmetry planes stay orthogonal to F_p, which is a geometric constraint deriving from the planar surface;

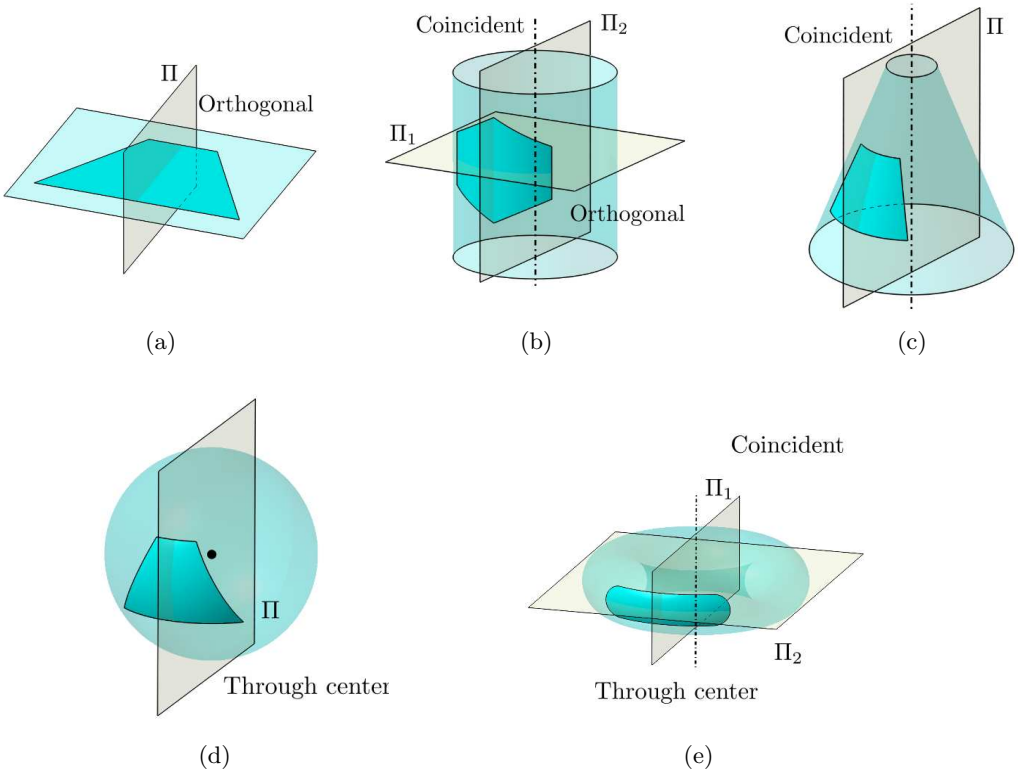


Figure 5.1: Symmetry properties of the five reference surfaces.

- **Cylinder**: To describe a cylinder F_{cy}, its intrinsic parameters reduce to a point, a vector to define its axis A and a radius. A acts as a symmetry constraint. As a surface of revolution, a cylinder is axisymmetric, i.e. any plane Π_2 passing through A is a symmetry plane. Considering an infinite cylinder, any plane Π_1, orthogonal to A is a symmetry plane (see Figure 6.2b). Inserting loops to limit a cylindrical area and form one domain reduces the symmetry planes to one plane Π_1 at most and up to infinity for Π_2 if there are two circular loops only. Otherwise, symmetry planes like Π_2 form a finite set;

- **Cone**: A cone F_{co} intrinsic parameters contain an axis, an apex location defining a reference point for the axis and a cone angle. The vector orientation defining the axis can be considered as the cone orientation to reduce the cone to one nappe. The symmetry constraint of an unbounded cone reduces to planes Π, necessarily containing its axis and exhibiting axisymmetry (see Figure 6.2c). Adding one or two loop(s) to generate a finite area preserves the axisymmetry if it(they) is(are) circular, otherwise the symmetry planes Π left are in finite number;

- **Sphere**: A sphere F_s intrinsic parameters are the location of its center and its radius. A full sphere is symmetric with respect to any plane Π passing through its center. Every vector originated from its center defines a symmetry axis (see Figure 6.2d). Reducing the area of F_s but still forming a surface bounded by one or two parallel circular loops reduces the symmetry axes to only one, orthogonal to the plane defined by the circular loop(s). Any other number and configuration of loops will result in a finite number of symmetry planes, all containing the center of F_s;

- **Torus**: A torus F_t originates from revolving a circle whose center does not lie on its revolution axis A. Torus intrinsic parameters are its small radius, average radius, axis of revolution and center O. Because it is a surface of revolution, F_t benefits of axisymmetry and all the symmetry planes Π_1 contain its axis. In this case, F_t is also symmetric with respect to Π_2, a plane orthogonal to A. If the surface of F_t is reduced by a circular loop contained in a plane Π orthogonal to A, only the axisymmetry of F_t around A is preserved. Now, if the surface of F_t is limited by two circular loops and if the planes $P_1$and P_2 containing each a loop are symmetrically set with respect to O, the axisymmetry and the symmetry plane Π_2 of F_t are preserved, otherwise if P_1 and P_2 are not symmetrically located around O, only the axisymmetry of F_t exists. If F_t is reduced with one arbitrary loop only that divides F_t in two areas, symmetry properties decrease to a finite number of planes Π_1. If there are two or more arbitrarily shaped loops bounding F_t, the symmetry properties of F_t are expressed by a finite number of symmetry planes Π_1 around A and, possibly the plane Π_2. These are the symmetry constraints set by a torus.

Definition 5 *: Reflective symmetry of two faces with respect to a plane* Π

Let two faces be F_1 and F_2 then, for every point $X_{F_1} \in F_1$, $X_{F_1} \subset I\!R^3$ and a symmetry plane Π, there exists a unique point X_{F_2} such that $X_{F_2} \in F_2$ and a point $M \in \Pi$ satisfying:

$$\overrightarrow{MX_{F_2}} = -\overrightarrow{MX_{F_1}} = -\|\overrightarrow{MX_{F_2}}\| \cdot \vec{n} \tag{5.1}$$

where F_1 and F_2 denotes two maximal faces each bounded by one loop at least, M is the normal projection of X_{F_1} on Π , $\vec{n}$ is the normal to M going through X_{F_1}. X_{F_2} is said to be the image of X_{F_1} on F_2, i.e. the symmetric point of X_{F_1} in F_2. Because this property holds for the infinite point set defined by F_1, F_2 is said the image of F_1. Conversely, if any point $X_{F_1} \in F_1$ has a symmetric point X_{F_2} lying in F_2 through Π, then Π is a symmetry plane for F_1 and F_2. Similarly, this definition applies also to space curves or loops, too.

With the five reference surfaces, the reflective symmetry between any two faces derives from their intrinsic parameters and locations in $I\!R^3$. To this end, they must be of same type, have the same intrinsic parameters (same radii, etc.). Secondly, their locations must be symmetric with respect to a plane Π. To formalize the configuration of symmetric location, a symmetric axes criterion is used:

Criterion 10 ***Reflective symmetry between axes with respect to a plane*** Π***:*** *Given a symmetry plane* Π *and two oriented axes* A_1 *and* A_2, Π *is regarded as reference entity, let* V_1 *be an arbitrary point of* A_1, $V_1 \in A_1$, *then,* V_2 *is a point symmetric of* V_1 *with respect to* Π *and* $V_2 \in A_2$. *Considering a second point* V_1', $V_1' \in A_1$, *its symmetric point with respect to* Π *is* V_2'. *If* V_2' *lies on* A_2, $V_2' \in A_2$, *then* $\overrightarrow{V_1V_2} \cdot \vec{n} = -\overrightarrow{V_1'V_2'} \cdot \vec{n}$, $\overrightarrow{V_1V_2} \cdot \overrightarrow{V_1'V_2'} > 0$ *where* $\vec{n}$ *is the normal to* Π, *which shows that* A_1 *is symmetric to* A_2 *with respect to* Π.

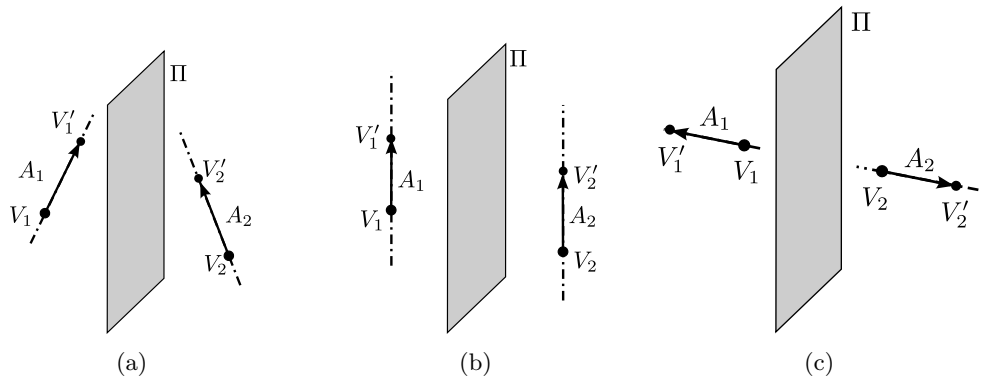


Figure 5.2: Examples of symmetric configurations for axes.

The definition of reflective symmetry for axes is described from a point set standpoint. With the axes symmetry criterion, the reflective symmetry criterion between two surfaces can be stated as follows:

- **Planes**: Any point V_1 on an infinite plane F_{p1} combined with the plane normal $\vec{n}_1$ forms an oriented axis. Considering a plane F_{p2}, if there exists a point V_2 of F_{p2} symmetric of V_1 with respect to a symmetry plane Π, a normal $\vec{n}_2$ of F_{p2} at V_2 defines also a oriented axis. Then, F_{p1} is said symmetric to F_{p2} with respect to Π if either $\vec{n}_1$ is symmetric to $\vec{n}_2$ or $\vec{n}_1$ is symmetric to $-\vec{n}_2$ since $\vec{n}_2$ and $-\vec{n}_2$ define the same point sets. These conditions always hold, which means that two symmetry planes can be associated to any couple (F_{p1}, F_{p2});

- **Cylinders**: Given two infinite cylinders F_{c1} and F_{c2} having the same radii. Given an arbitrary point V_1 on the axis of F_{c1}, V_1 and this cylinder axis define

an oriented axis $\vec{A}_{c1}$. Let V_2 be a point symmetric of V_1 with respect to a plane Π such that V_2 belongs to the axis of F_{c2}. V_2 and this last axis form an oriented axis $\vec{A}_{c2}$. Then, F_{c1} is said to be symmetric to F_{c2} with respect to Π if either $\vec{A}_{c1}$ is symmetric to $\vec{A}_{c2}$ or $\vec{A}_{c1}$ is symmetric to $-\vec{A}_{c2}$, i.e. similarly to a plane $\vec{A}_{c2}$ and $-\vec{A}_{c2}$ define the same point sets and characterize two symmetry planes;

- **Cones**: Considering two infinite cones F_{co1} and F_{co2} with one nappe having the same angle and V_1 and V_2 as apices respectively. These apices together with their respective cone axes uniquely define oriented axes $\vec{A}_{co1}$ and $\vec{A}_{co2}$ respectively. If $\vec{A}_{co1}$ is symmetric to $\vec{A}_{co2}$ with respect to a plane Π, then F_{co1} and F_{co2} are symmetric with respect to the plane Π;

- **Spheres**: Two spheres F_{s1} and F_{s2} of identical radii and center points V_1 and V_2 respectively are symmetric with respect to a plane Π if V_1 and V_2 are symmetric with respect to this plane. V_1 and V_2 are sufficient to define axes for F_{s1} and F_{s2} and there always exists Π for this couple of surfaces;

- **Tori**: The condition for two tori F_{t1} and F_{t2} is close to that of cylinders and planes. Let V_1 and V_2 be the center of F_{t1} and F_{t2}, respectively. These centers, together with their respective torus axes define oriented axes $\vec{A}_{t1}$ and $\vec{A}_{t2}$ respectively. The two tori F_{t1} and F_{t2} are symmetric with respect to the plane Π either if $\vec{A}_{t1}$ is symmetric to $\vec{A}_{t2}$ or $\vec{A}_{t1}$ is symmetric to $-\vec{A}_{t2}$ since these axes configurations define the same point sets and a unique symmetry plane.

The surface self reflective symmetry condition and two surfaces reflective symmetry conditions are the basic symmetry properties of surfaces seen as infinite point sets that form the basic symmetry criteria of the symmetry analysis algorithm. These symmetry properties reflect configurations where the point sets are either connected, in the first case, or possibly disconnected, in the second one. It is an illustration of the influence of a topological property of the point sets: the number of connected components of these sets.

5.3 Symmetry constraints originated from face boundary loops

The five reference surfaces are divided into two groups: one group is of type infinite surfaces, which contains planes, cylinders and cones; the other group is of type finite surfaces and contains spheres and tori. The faces F_i forming M_{Max} are bounded, their boundary is defined by a set of closed loops. The effect of boundary loops over a face differs according to the category they belong to. Classically, two categories are considered for bi-parametric surfaces (see Figure 5.3a):

- The first one is limiting the surface area and acts as an external boundary. This category applies to unbounded surfaces (planes, cylinders, cones) or to finite surfaces (tori) when they need to be split apart into pieces;
- The second one expresses material removal from the surface to form holes and it helps distinguishing internal and external boundaries.

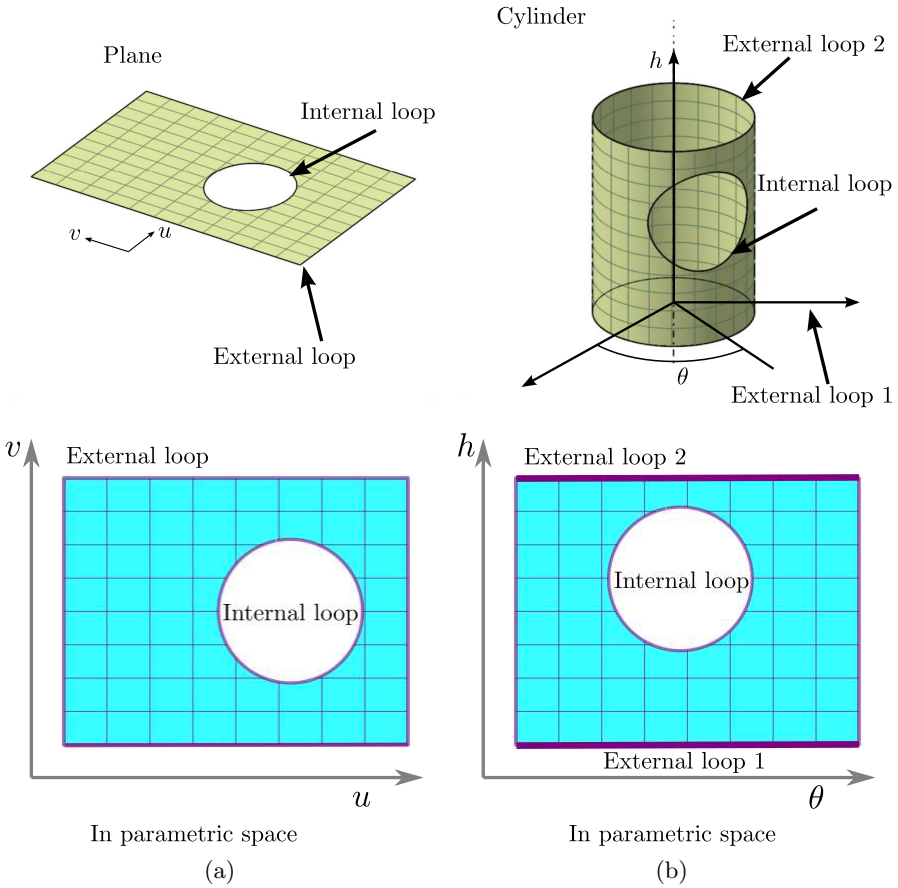


Figure 5.3: Loop categories in case of parametric surfaces: (a) loop categories in the parametric space of a plane and its corresponding surface, (b) loop categories in the parametric space of a cylinder and its corresponding surface.

When applicable for the same type of surfaces, these two categories of loops complement each other. In fact, bi-parametric surfaces can describe cylindrical, conical, spherical, toroidal areas but their loop boundary may not match the loops obtained after the generation of maximal curves and faces. Taking the example of a cylinder, its

parameterization is classically based on directions along its generatrix and circular section. B-Spline, NURBS and analytical (models using trigonometric functions) models produce necessarily a rectangular domain where the circular section uses a reference point to locate and stitch extremities of the parametric domain (see Figure 5.3b). Consequently, the loops obtained after the generation of the maximal curves and surfaces differ from that of B-Spline and analytical models because they don't refer to any parameterization and any reference point. The loops derived from the hypergraphs are regarded as more intrinsic to the cylinder than those of the B-Spline and analytical models. Rather than reasoning in the parametric space of surfaces, to stick to the loops resulting from the maximal curves and surfaces generation it becomes necessary to take into account the embedding of each surface in $\mathbb{R}^3$ to characterize the loops bounding the surfaces. To this end, the purpose of this sub-section is a proposal for the characterization of the loops bounding the reference surfaces $\tilde{F}$.

The external loops and internal loops are defined as:

Definition 6 ***External loop:*** *An external loop is a piecewise C^0 curve without self intersection. This type of loop is defined in $\mathbb{R}^3$ where a reference surface is embedded, not in its parametric space as usual. Depending on the surface type, an external loop can be characterized by:*

- *the largest bounding box when it lies on a planar surface, see Figure 5.5a;*
- *the largest integral angle*[1] *covered on the section of a cylinder around its axis. If the largest angle equals 2π, there must be two external loops, see Figure 5.5b and c;*
- *the largest integral angle covered on the section of a cone around its axis. If the maximal integral angle reaches 2π, the corresponding loop is an external one and there can be two such ones, see Figure 5.5d, e and f;*
- *On a torus, if a loop reaches the integral angle of 2π around its axis, this loop is not dividing the torus into two domains. In this case, there must be two such loops to define a partition of the torus. Similarly, if a loop reaches 2π along its average circle, two such loops must exist to define a partition of the torus.*

Obviously, loops contributing either to the definition of a finite surface or a partition in a finite surface are classified as external loops.

Definition 7 ***Internal loop:*** *An internal loop is a piecewise C^0 curve without self intersection. This type of loop is also defined in $\mathbb{R}^3$ where a reference surface is embedded. Depending on the surface type, an internal loop is:*

[1] the notion of integral angle will described in more details in section 5.7

- *a loop within a bounding box smaller than the largest one on a plane;*
- *a loop with a sector angle smaller than the largest angle or smaller than* 2π *on a cylinder;*
- *This applies similarly on a cone and on a torus.*

As a common denominator of internal loops, they are all trivial cycles from a topological point of view (see Figure 5.4a).

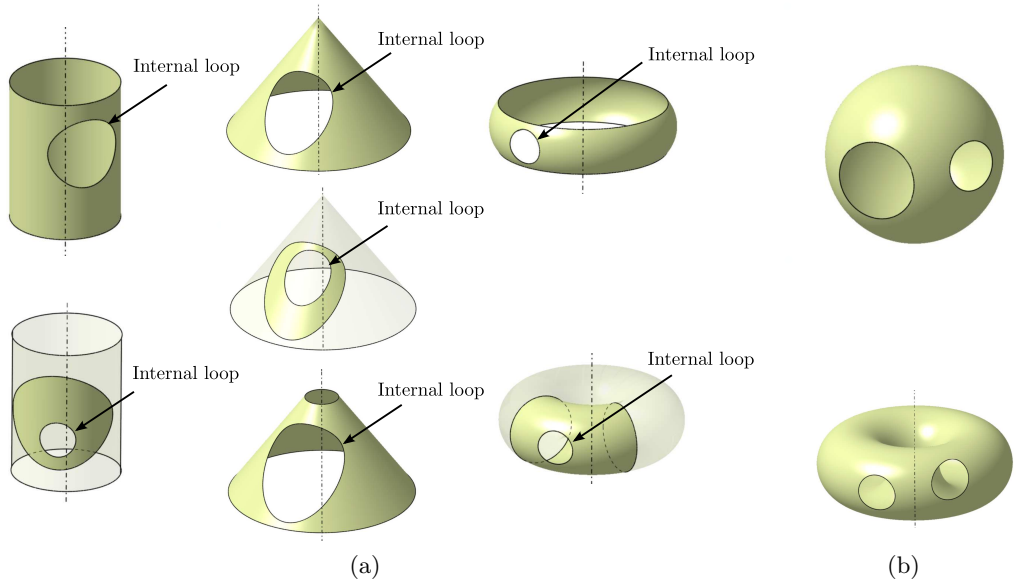


Figure 5.4: Loop categories: (a) examples of internal loops on cylinder, cone and torus, the marked loop is internal one and the rests are external, (b) loop configurations over spheres and tori.

Indeed, the external loop of a cone differs from that of a cylinder because the external loop of a cylinder is a one-cycle from a topological point of view whereas the external loop of a cone is a trivial one. However, the external loop of a cone has a geometric feature since it contains the cone apex, a geometric singularity characterizing the cone embedding in $I\!R^3$. Therefore, loops taking into account this feature get the status of external loop.

The case of the sphere is particular and has not been processed above. When there is only one loop on the sphere that divides the surface into two new ones, there is no possible classification of the loop. This observation applies also to the torus when it is cut along trivial cycles only. However, when there is more than one loop over these surfaces, two configurations can occur (see Figure 5.4b):

- There exists one loop containing all the others. It means that one partition of the surface contains all the others apart from one. In this case, this loop is the 'external' one and the others are 'internal' ones. There cannot be any other loop containing several loops;

- There is no loop containing other loops. Here, there is no classification possible.

As a conclusion regarding the above analysis, processing configurations of spheres or tori containing only one loop is considered as a configuration without status since there is no strong property to form a homogeneous scheme.

An infinite surface has to have at least an external loop Γ^{ex} to reduce its infinite area to a finite one. Within the finite surface area, there can be numerous loops which are internal loops: Γ^{in}. For a face F, $\partial F = [\Gamma_F]$. Here, $[\Gamma_F]$ designates the set of loops of F:

$$[\Gamma_F] = [\Gamma_F^{ex}] \cup [\Gamma_F^{in}] \cup [\Gamma_F^{uk}]$$

because the torus and the sphere are finite surfaces, they can exhibit configurations where they have loops without status. Across different surface types, the external loop number and property differ in accordance to the surface embedding (see Table 5.1 and Figure 5.5).

Surface type	Plane	Cylinder	Cone	Sphere	Torus
Surface extent	Infinite	Infinite	Infinite	Finite	Finite
1st ext-loop	1	1	1	1 or unknown	1 or unknown
2nd ext-loop	no	1	0 or 1	no	0 or 1
Figure 5.5	(a)	(b)(c)	(d)(e)(f)		

Table 5.1: The number of external loops for different surfaces

As mentioned in section 6.2, a surface has intrinsic symmetry constraints, but before discussing its symmetry, an infinite surface has to be bounded to become a subset of a B-Rep model. Consequently, a surface boundary sets also important symmetry constraints. When adding a surface boundary, surface symmetry properties can be summarized as follows, in accordance with the face type of F, i.e. taking into account the embedding of the surface. Consequently, the following sections analyze the symmetry constraints set by face boundary curves on the symmetry properties of faces and highlight the various configurations of symmetry planes encountered.

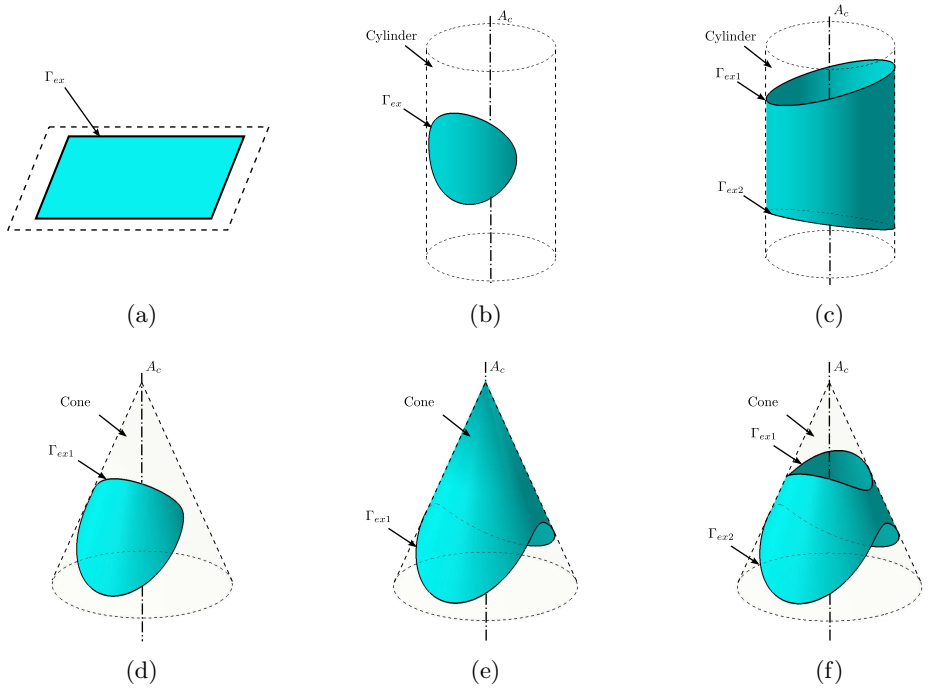


Figure 5.5: Illustrations of external loops for: (a) planes, (b) cylinders, (c) cones.

5.3.1 Symmetry constraints derived from loops over planar faces

Any symmetry plane Π of a planar face P is orthogonal to P. P being bounded, its boundary ∂P must be closed, i.e. it is at least C^0 continuous. ∂P can be formed of an arbitrary number n_L of loops $\Gamma_i, i \in \{1, \cdots, n_L\}$, each loop defining an area S_i, i.e. $\partial S_i = \Gamma_i$ (see Figure 5.6a). As a finite, connected, planar graph without any edge intersection, there exist only two types of loops: one is the external loop (see Figure 5.6, Γ_1), the others are internal loops (see Figure 5.6, Γ_2 and Γ_3).

For any ∂P, the external loop, say Γ_1, is unique and defines a finite planar area. The internal loops can be in arbitrary number and lie inside the external loop. An internal loop, Γ_i, $i \in \{2, \ldots, n_L\}$, cannot contain any other loop, $\Gamma_i \cap \Gamma_j = I$, $i \neq j$ where I is either a set of isolated points if the loops touch each other or the empty set. If I happens to be something else, P would not be connected.

Every loop Γ_i is described by n_E maximal edges $[E]$. Each edge has its own reflective symmetry information defined as *orthogonal symmetry*[2] $[\Pi_O]$ or *bisector symmetry plane* $[\Pi_{BS}]$, and two edges connected at a vertex have symmetry informa-

[2]This category of CSP as well as the following ones are explained in detail and analyzed in the following sections.

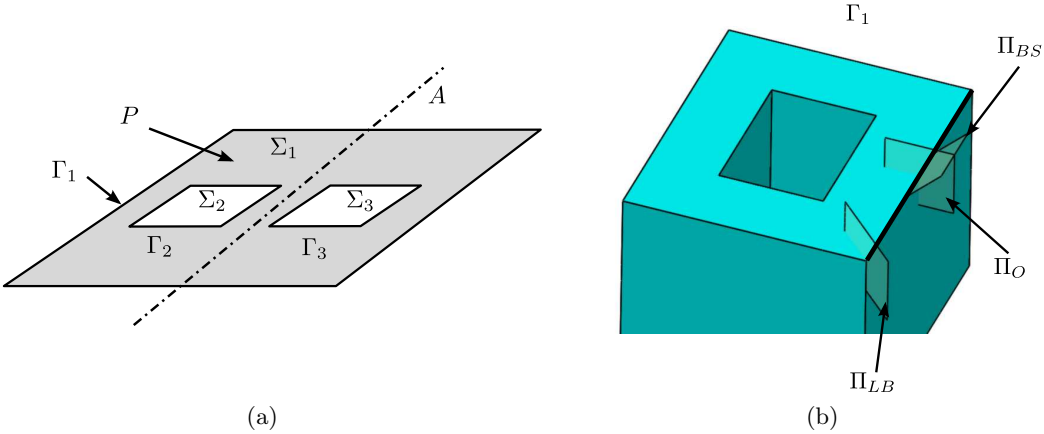


Figure 5.6: An example of planar face P: (a) bounded by three loops Γ_i and the trace of the symmetry plane A on P, (b) with an illustration of symmetry plane categories and their interaction with Γ_1.

tion defined as *loop bisector symmetry* $[\Pi_{LB}]$ (see Figure 5.6b).

Considering an internal loop Γ_j, if a reflective symmetry plane Π_{Γ_j} exits for it, it has to coincide with one or two symmetry planes attached to the external loop, i.e. they can be of type orthogonal or loop bisector symmetry planes, to extend the validity of the symmetry properties over P. Otherwise, the edges intersecting Π and the vertices lying on Π don't contain the same symmetry properties. If Π_{Γ_j} exists, this loop symmetry plane is a subset of the CSP collection of type orthogonal and loop bisector, $[\Pi_{\Gamma_j}] \subset ([\Pi_O] \cup [\Pi_{LB}])$. Because $[\Pi_O]$ derives from an edge and $[\Pi_{LB}]$ from a vertex adjacent to two edges, there is no current need to analyze the symmetry property of two disconnected edges: this is left to the propagation phase.

5.3.2 Symmetry constraints derived from loops over cylindrical faces

Any symmetry plane Π of a cylindrical face Cy is either (see section 5.2):

a) containing the cylinder axis A_c;

b) or orthogonal to it.

Existence of these symmetry planes is subjected to the symmetry properties of the boundary ∂Cy of Cy. Similarly to P, ∂Cy must be closed and can be formed of an arbitrary number of loops, n_L, Γ_i, $i \in \{1, \cdots, n_L\}$. The components of ∂Cy lie on Cy and can be parameterized in cylindrical coordinates, i.e. $\Gamma_i = \Gamma_i(t_i) = \Gamma_i(\theta(t_i), z(t_i))$.

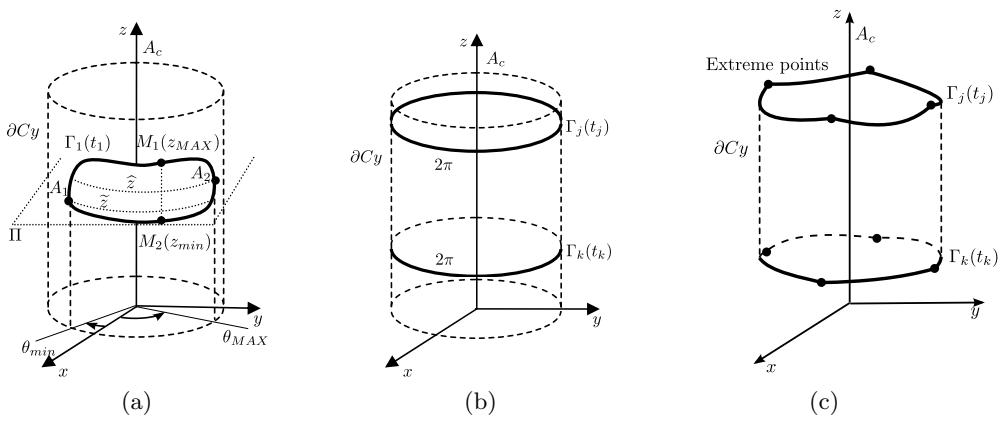


Figure 5.7: Parameters for symmetry configurations on a cylinder: a) boundary $\Gamma_1(t_1)$ with an orthogonal symmetry plane Π and $arc(A_1, A_2) < 2\pi$; b), c) two boundaries $\Gamma_j(t_j)$ and $\Gamma_k(t_k)$ with $arc = 2\pi$; c) with extreme points.

A cylinder can have one or two external loops (see section 5.3). An external loop provides symmetry constraints because it must be self symmetric when a symmetry plane exists for it.

Firstly, let us consider case b) above with ∂Cy containing one external loop. Let Π be the symmetry plane of the straight line segments $[M_1, M_2]$ (see Figure 5.7a) coinciding with a generatrix of Cy and bounded by ∂Cy: $M_1 = \Gamma_1(\theta, z_{MAX})$, $M_2 = \Gamma_2(\theta, z_{min})$. Then, Π will be a symmetry plane orthogonal to A_c for Cy if Π holds for any point of ∂Cy. There can be at most one such symmetry plane Π orthogonal to A_c. The corresponding symmetry plane can derive from an edge as *orthogonal symmetry* Π_O or a vertex as *loop bisector symmetry* Π_{LB} of Γ_i.

Then, regarding case a), the area of Cy is finite when ∂Cy contains one loop $\Gamma_1(t_1)$ if this loop projects along z into a curvilinear arc A_1A_2 parameterized with an angle smaller than 2π, i.e. the definition domain of $\Gamma_1(t_1)$ according to θ: $A_1 = \Gamma_1(\theta_{min}(t_{1min}), \widetilde{z})$, $A_2 = \Gamma_1(\theta_{MAX}(t_{1MAX}), \widehat{z})$ and $arc(A_1, A_2) < 2\pi$, $\forall(\widetilde{z}, \widehat{z}) \in [z_{min}, z_{MAX}]$ (see Figure 5.7a). θ is the radial angle of Cy and the integral angle describing ∂Cy is: $\theta \in [\theta_{min}, \theta_{MAX}]$ and $|\theta_{MAX} - \theta_{min}| < 2\pi$. Let Π be a plane containing A_c, Π is generated from two points $N_1 = \Gamma_1(\theta_\Pi, z_1)$, $N_2 = \Gamma_1(\theta_\Pi, z_1)$ of ∂Cy. If Π is a symmetry plane for any point of ∂Cy, then Π is a symmetry plane for Cy. There is no more than one symmetry plane containing A_c. Also, Π can derive from an edge as *orthogonal symmetry* Π_O or a vertex as *loop bisector symmetry* Π_{LB}.

Now, if ∂Cy contains more than one component, then either:

1) two external loops $\Gamma_j(t_j)$, $\Gamma_k(t_k)$ are parameterized with an integral angle $|\theta_{MAX} -$

$\theta_{min}| = 2\pi$;

2) one external loop and all the loops $\Gamma_i(t_i)$ are parameterized with an integral angle $|\theta_{MAX} - \theta_{min}| < 2\pi$.

If ∂Cy falls into case 1, then if $\Gamma_j(t_j)$ and $\Gamma_k(t_k)$ are defined as circles $C(\theta)$ (see Figure 5.7b), i.e. $\Gamma_j(\theta(t_1), z_1) = C_1(\theta)$, $\Gamma_k(\theta(t_2), z_2) = C_2(\theta)$, $\theta \in [0, 2\pi]$ and are the only loops of ∂Cy; Cy contains an infinite number of symmetry planes passing through A_c, i.e. Cy is axi-symmetric. These two loops generate also a symmetry plane orthogonal to A_c, that does not contain any vertex or any edge of Γ_j, Γ_k; it is designated as *loop symmetry* CSP: Π_{LS} (see section 5.4).

If $\Gamma_j(t_j)$ and $\Gamma_k(t_k)$ are not circles (see Figure 5.7c) but loops such that, $\Gamma_j(t_j) = \Gamma_j(\theta_i, z_i)$ are either:

a) extreme points lying on a circle $C_1(\theta_i)$: $\Gamma_j(\theta_i, z_1) = C_1(\theta_i)$ and $\frac{\partial \Gamma_j}{\partial z}(\theta_i, z_1) = 0$;

b) or/and points where the geometric continuity of $\Gamma_j(t_j)$ is either G^0 or G^1 at most. The points of G^0 or G^1 continuity are originated from the intersection curves between the faces adjacent to Cy, as a result of the object modeling process, and are the extreme points of the maximal edges.

Considering only key points b and assuming a finite number of values of $\theta_i \in \{\theta_1, \theta_2, \ldots, \theta_n\}$, $\theta_i \in [0, 2\pi]$, and similarly for $\Gamma_k(t_k)$ with $z = z_2$ and $[\theta_i]$, the same set as above. Then, the number of symmetry planes over Cy and attached to these key point reduces to a finite number with n as upper bound. It reflects configurations where key points of $\Gamma_j(t_j)$ and $\Gamma_k(t_k)$ are located on circles and may generate a set of CSPs. These CSPs are all attached to vertices as *loop bisector symmetry* $[\Pi_{LB}]$ CSPs containing A_c. If the key points listed above satisfy all the above conditions but are not located on circles, n is no longer the upper bound.

More precisely, if n is an even number, the resulting symmetry planes containing A_c are pairwise coinciding because pairs of symmetry planes (Π_i, Π_l) are referenced by angles θ_i and there exist angles $\theta_l = \theta_i + \pi$ to define Π_l, which means that Π_i and Π_l coincide in 3D space. Hence, the number of distinct planes in this case reduces to $n/2$ as upper bound. If n is an odd number, all symmetry planes differ from each other in 3D space: the upper bound stays n. In what follows, the distinction between configurations with odd or even numbers of symmetry planes is not detailed for sake of conciseness unless stated for more accuracy.

Now, $\Gamma_j(t_j)$ and $\Gamma_k(t_k)$ must be symmetric along θ, $\theta_i = 2\pi/i$ so that the previous key points are really the location of symmetry planes, i.e. $[\Pi_{LB}]$, $[\Pi_O]$ CSPs. This writes: $\Gamma_j(\theta + i\theta_i, z) = \Gamma_j(-\theta + i\theta_i, z)$, $\Gamma_k(\theta + i\theta_i, z) = \Gamma_k(-\theta + i\theta_i, z)$, $\forall i \in [1, n]$,

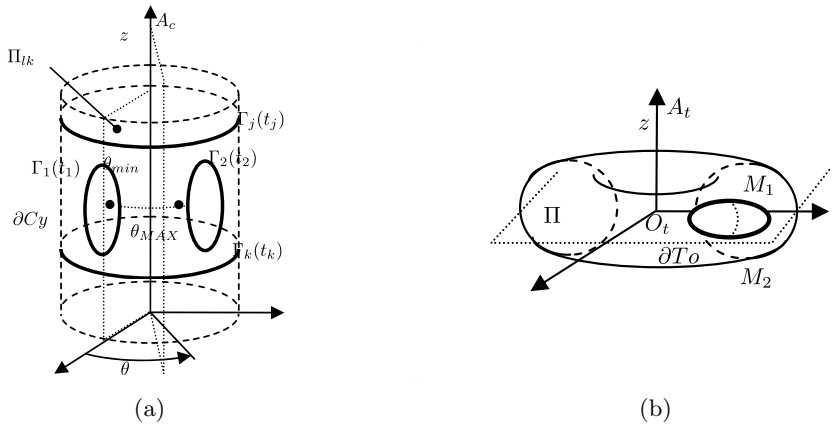


Figure 5.8: (a) Parameters for symmetry configurations on a cylinder with internal loops, (b) Parameters on a torus.

$\theta \in [0, \pi]$. In addition to the previous condition, the key points being extreme values along z, there must exist other key values where $\frac{\partial \Gamma_j}{\partial z}(\widetilde{\theta}_i, z_3) = 0$ and/or $\Gamma_j(\widetilde{\theta}_i, z_3)$, $\widetilde{\theta}_i \in \{\widetilde{\theta}_1, \widetilde{\theta}_2, \ldots, \widetilde{\theta}_n\}$, $\widetilde{\theta}_i \in [0, 2\pi]$ between the key values θ_i, i.e. if $\Gamma_j(\theta_i, z_1)$ are maxima, then $\Gamma_j(\widetilde{\theta}_i, z_3)$ are minima and conversely. Between two consecutive key points (θ_i, θ_{i+1}), the arcs of Γ_j and Γ_k can be arcs of circles: extreme points reduce here to middle points of the corresponding arcs. If $\widetilde{\theta}_i = (\theta_i + \theta_{i+1})/2$, $i \in [1, (n-1)]$ and $\Gamma_j(\theta, z) = \Gamma_j(-\theta, z)$, $\theta \in [0, \widetilde{\theta}_i]$, the interval $\theta \in [\theta_i, \theta_{i+1}]$ is symmetric with respect to $\widetilde{\theta}_i$ and if the same property applies to $\Gamma_k(\theta, z)$, the number of symmetry planes for Cy reaches the upper bound $2n$ if n is odd or n otherwise. All the corresponding CSPs thus added are now attached to edges and express *orthogonal symmetry*, $[\Pi_O]$, and contain A_c.

If $\Gamma_j(t_j)$ and $\Gamma_k(t_k)$ don't share the same sequence of key values, i.e. θ_i, $i \in [1, n]$ and $\widehat{\theta}_i$, $i \in [1, m]$ respectively, the maximum number of symmetry planes varies according with the greatest common divisor of n and m, i.e. $gcd(m, n)$. Anyhow, if $gcd(m, n) = 1$, the resulting number of symmetry planes is at most one.

If ∂Cy contains more loops than the two external ones $\Gamma_j(t_j)$ and $\Gamma_k(t_k)$ (see Figure 5.8a), it means there are internal loops. The number of symmetry planes deriving from Γ_j and Γ_k may be further reduced depending on the symmetry properties of the internal loops Γ_i, $i \neq j$, $i \neq k$. If these components are uniformly spread over θ and form a repetitive pattern, symmetry planes may exist in between two successive loops, generated by the arcs connecting the components. Such a set of CSPs is related to the symmetric location of these internal loops and fall into the category of *loop symmetry* CSPs $[\Pi_{LS}]$. The corresponding symmetry planes Π_j in 3D space amount, at most, to the number of internal loops n_I, $n_I = (n_L - 2)$, when n_I is odd or $n_I/2$ when n_I is even. If the loops Γ_i, $i \neq j$, $i \neq k$ are themselves symmetric, another set of symmetry planes Π_k exists in addition to Π_j, these ones are necessarily attached

either to a vertex or an edge, hence they are either $[\Pi_{LB}]$ or $[\Pi_O]$ CSPs, respectively. The amount of such symmetry planes equals to either n_I or $n_I/2$ if n_I is odd or even, respectively. Finally, the total number of symmetry planes generated by the internal components can reach either $2n_I$ or n_I whether n_I is odd or even, respectively.

The total number of symmetry planes in Cy can be stated as: $n_T = min(gcd(m,n), 2n_I)$ in the most general configuration or other variants of this minimum when $gcd(m,n)$ or $2n_I$ are either odd or even, respectively. This analysis shows that the overall number and type of CSPs of a cylindrical face is set, in a first place, by its external loops and then, is further reduced by the properties of its internal loops. If $[\Pi_\Gamma]$ defines the set of CSPs for Cy, then $[\Pi_\Gamma] \in ([\Pi_O] \cup [\Pi_{LB}] \cup [\Pi_{LS}])$.

If ∂Cy falls into case 2, then Cy has at most one symmetry plane Π containing A_c since all the arcs $\widehat{A_1A_2}$ whose extremities lie on ∂Cy, are smaller than 2π and orthogonal to A_c. Indeed, $[\Pi_\Gamma] \in ([\Pi_O] \cup [\Pi_{LB}] \cup [\Pi_{LS}])$. In particular, $[\Pi_\Gamma] \in (\Pi_O \cup \Pi_{LB})$ where Π_O and Π_{LB} are attached to the external loop of Cy.

The above analysis does not distinguish all the configurations since it intents to highlight some categories of symmetry planes and highlight their interactions, their connections with faces, edges, vertices and loops as well as the interest of loop status. Some of these observations can be also applied to planar and other faces but are not stated to avoid unnecessary repetitions.

5.3.3 Symmetry constraints derived from loops over conical faces

Any symmetry plane Π of a conical reference surface Co necessarily contains the cone axis A_c. Existence of symmetry planes is subjected to the symmetry properties of the boundary ∂Co of Co. As defined with an oriented axis, Co contains only one sheet, possibly including the cone apex. Similarly to P and Cy, ∂Co must be closed and can be formed of an arbitrary number n_L of loops $\Gamma_i \in \{1, \dots n_L\}$. Similarly to Cy, the loops of ∂Co can be parameterized in cylindrical coordinates, i.e. $\Gamma_i = \Gamma_i(t_i) = \Gamma_i(\theta(t_i), z(t_i))$.

Compared to Cy, the area defined by ∂Co is always finite even if ∂Co reduces to only one component, i.e. there is no such condition as in case 1 of Cy. Indeed, all loops over Co are topologically identical to each other and are cycles that can be reduced to a point, i.e. trivial cycles. However, if ∂Co contains one external loop $\Gamma_1(\theta, z)$ parameterized with an integral angle $|\theta_{MAX} - \theta_{min}| = 2\pi$, it can be handled using the properties stated at case 1 of Cy and applied to $\Gamma_1(\theta, z)$ alone to characterize the maximum number of symmetry planes. This similarity holds because the cone apex is a point with a geometric singularity that can be used to distinguish loops cycling around it from those that are not. In case of such external loops, ∂Co must bound a domain containing the cone apex, which needs to be checked prior to process the loops $\Gamma_1(t_1)$. If ∂Co contains two loops parameterized with an integral angle

$|\theta_{MAX} - \theta_{min}| = 2\pi$, this configuration falls into case 1 of Cy. ∂Co cannot contain more than two external loops.

If ∂Co contains more than one loop, then cases considered above for Cy apply and processing ∂Co for symmetry planes related to its external boundary as well as its internal components is similar to the description given for Cy. If one of the loops satisfies $|\theta_{MAX} - \theta_{min}| = 2\pi$, ∂Co conforms to case 1 of Cy, otherwise cases fall into case 2.

5.3.4 Symmetry constraints derived from loops over toroidal faces

Any symmetry plane Π of a toroidal reference surface To is either containing the torus axis A_t or orthogonal to it and passing through the center O_t. Processing To is rather similar to the properties of Cy since To contains categories of cycles that are topologically similar to those existing on a cylinder. The symmetry definition particularly applies to the two categories of arcs forming To.

Let us first consider the arcs forming the small radius of To, a symmetric configuration holds if at least one arc exists between two points M_1, M_2 of ∂To such that Π, the symmetry plane orthogonal to A_t, is a symmetry plane for this segment (see Figure 5.8b). Then, Π will be a symmetry plane orthogonal to A_t for To if Π holds for any point of ∂To. There can be at most one such symmetry plane for To.

The loops of ∂To are parameterized with the intrinsic torus coordinates, i.e. $\Gamma_i = \Gamma_i(t_i) = \Gamma_i(\theta(t_i), \alpha(t_i))$ where θ is the angle around A_t (see Figure 5.8b).

Similarly to Cy, two configurations exist that influence the distribution of symmetry planes. If ∂To contains more than one loop, then either:

1) two of these loops $\Gamma_j(t_j)$, $\Gamma_k(t_k)$ are parameterized with an angle $|\theta_{MAX} - \theta_{min}| = 2\pi$. There cannot be more than two components parameterized that way;

2) two of these loops $\Gamma_j(t_j)$, $\Gamma_k(t_k)$ are parameterized with an angle $|\alpha_{MAX} - \alpha_{min}| = 2\pi$. No more than two components can be parameterized that way;

3) all the components $\Gamma_i(\theta, \alpha)$ are parameterized with angles $|\theta_{MAX} - \theta_{min}| < 2\pi$ and $|\alpha_{MAX} - \alpha_{min}| < 2\pi$.

Then, the detailed study of these configurations follows the approach of cylinders where the z parameter is substituted by α to represent case 1. It is not replicated here but can be easily transposed from that of Cy.

Considering the arcs forming the average radius of To, parameterized with θ, these arcs are equivalent to the generatrices of Cy and the arcs forming the small radius of To, parameterized with α are equivalent to the circles defining the section of Cy.

Somewhat similar to loops over Cy, loops $\Gamma_j(t_j)$, $\Gamma_k(t_k)$ falling into case 2 above define a finite area over To. At the difference of Cy where such loops would exhibit symmetry properties as described at section 5.3.2, this configuration cannot produce as many symmetry planes as Cy since the generatrices of Cy are now replaced by arcs. Indeed, one symmetry plane exists at most, which is orthogonal to A_t and contains O_t (see section 6.2 for the possible symmetry planes of a torus).

5.3.5 Symmetry constraints derived from loops over spherical faces

Any symmetry plane Π of a spherical face Sp necessarily contains the center of the sphere O_s. If ∂S_p is the boundary of S_p and $\Gamma_i \in \{1, \ldots, n_L\}$ its loops, all the areas bounded by Γ_i are finite and so are their complement, i.e. $Area(S_p) - Area(S_p|_{\Gamma_i})$. This property shows that the configurations 1 and 2 set for cylinders and cones are no longer applicable here since the loop status over S_p does not refer to the concept of external/internal. Differing from cylinders and cones, a sphere has no particular direction for symmetry planes. Hence, symmetry planes over S_p rely more strongly on the symmetry properties of ∂S_p, which can be characterized from symmetry and location properties of faces F_a adjacent to S_p. These properties are detailed in section 5.4 where the influence of references surfaces will be detailed.

5.3.6 Synthesis about loops and loop symmetry CSP

From the analysis of symmetry constraints related to face boundaries (see section 5.3), the importance of loops and the diversity of CSPs $\Pi_O, \Pi_{LB}, \Pi_{LS}$ becomes clear in a surface symmetry analysis process. Whether a face F has an external loop or not, the number of loops and their relative position over F, all these informations are key elements influencing the symmetry properties of F and give bounds on the maximum number of symmetry planes depending on loop configurations.

Each loop Γ_i of F has its own symmetry plane set, namely self symmetry planes: $[\Pi^{slf}_{\Gamma_i}]$. These symmetry planes cross Γ_i, which indicates that they can be either of type orthogonal $[\Pi_O]$ or loop bisector $[\Pi_{LB}]$ only. Now, when considering different loops $\Gamma_i, \Gamma_j \in F$, their common self symmetry planes may come from the intersection between $[\Pi^{slf}_{\Gamma_i}]$ and $[\Pi^{slf}_{\Gamma_j}]$. Because loops over F must be either disconnected or touch each other at isolated points only to stay consistent with B-Rep model fundamentals, $[\Pi^{slf}_{\Gamma_i}] \cap [\Pi^{slf}_{\Gamma_j}]$ may not produce all the symmetry planes set by the constraints $\Gamma_i \cup \Gamma_j$.

The relative position of Γ_i and Γ_j, as two disconnected infinite point sets, may also satisfy the symmetry condition 5.1 where F_1 and F_2 are substituted by Γ_i and Γ_j. Consequently, the corresponding symmetry plane Π is called *loop symmetry* candidate symmetry plane and noted Π_{LS}. Indeed, Π_{LS} being derived from loops Γ_i and Γ_j, it can be noted more precisely $\Pi_{LS,\Gamma_{i,j}}$. Regarding their relative position in M_{MI},

Γ_i and Γ_j cannot touch each other more than once, otherwise they define a face F_k, homologous to F, lying in between the faces F_i and F_j that could be associated to Γ_i and Γ_j respectively. Each contact point being surrounded by more than three faces F_i, F_j, F_k and F, and no more than two loops in contact can be attached to two faces (see section 5.4.2), e.g. $F_i = F_j = F_{ij}$, then F_{ij} crosses F_k ($F_k = F$). Hence, other loops (faces) touching F_i or F_j or F_k are not homologous with these ones, therefore these faces are not crossing F and F_{ij} and the manifold vertex splitting operator will duplicate these contact points but the edge merging one will remove them, leaving no more than one contact point representing the crossing configuration in M_{MAX}. The CSP attached to that point is necessarily of type Π_{LB} not Π_{LS}. All the contact points associated to non crossing configurations will be removed and represented by Π_{LS} not Π_{LB}.

This analysis shows that Π_{LS} are strictly attached to loop entities and, hence, disconnected from the hypergraphs Gij, which contain vertices and maximal edges only. This justifies the existence of loop datastructures derived from the hypergraphs to complement the faces, edges and vertices.

Let $[\Gamma^{ex}]$, $[\Gamma^{in}]$ and $[\Gamma^{uk}]$ be the sets of loops representing the *external loops*, *internal loops* and *loops of unknown status* of F, respectively. The loops Γ_i, Γ_j and Γ_k belong to either set $[\Gamma^{ex}]$ or $[\Gamma^{in}]$ or $[\Gamma^{uk}]$ without influencing the existence of Π_{LS}. However, Π_{LS} cannot be defined by loops Γ_i and Γ_j belonging to either set: $\Gamma_i \in [\Gamma^{ex}]$, $\Gamma_j \in [\Gamma^{in}]$, for example. Consequently to the above analysis of connected/disconnected loops, one pair of loops only has a unique Π_{LS} plane. Because $\Pi^{LS}_{\Gamma_{(i,j)}}$ is not crossing any other loop, $\Pi^{LS}_{\Gamma_{(i,j)}} \notin [\Pi^{slf}_{\Gamma_i} \cap \Pi^{slf}_{\Gamma_j}]$. Therefore, the set of symmetry planes of Γ_i and Γ_j satisfy:

$$[\Pi_{\Gamma_{(i,j)}}] = ([\Pi^{slf}_{\Gamma_i}] \cap [\Pi^{slf}_{\Gamma_j}]) \cup \Pi^{LS}_{\Gamma_{(i,j)}}. \tag{5.2}$$

In case a face F is bounded by external loops $[\Gamma^{ex}]$ as well as internal ones $[\Gamma^{in}]$, because $[\Gamma^{ex}]$ can be said as the mandatory limitation of F, i.e. the loops that avoid having F unbounded, and they are distinct from $[\Gamma^{in}]$, whatever the symmetry properties of $[\Gamma^{in}]$, if $[\Gamma^{ex}]$ is not symmetric, F is not symmetric. As a result, $[\Pi_{\partial F}] \subset [\Pi_{\Gamma^{ex}}]$. Now, considering $[\Gamma^{in}]$, the symmetry planes of F must be contained in $[\Pi_{\Gamma^{in}}]$: $[\Pi_{\partial F}] \subset [\Pi_{\Gamma^{in}}]$. Finally, we have:

$$[\Pi_{\partial F}] = [\Pi_{\Gamma^{ex}}] \cap [\Pi_{\Gamma^{in}}]. \tag{5.3}$$

To unify eq. (5.3), when Γ^{ex} does not exist, $[\Pi_{\Gamma^{ex}}]$ means all possible symmetry planes of F in $I\!R^3$. Similarly, if Γ^{in} does not exist, $[\Pi_{\Gamma^{in}}]$ means all possible symmetry planes of F in $I\!R^3$, too. In these cases, symmetry plane sets are infinite. When F contains loops of unknown status, it means that $[\Pi_{\Gamma^{ex}}]$ does not exist. Indeed, $[\Gamma^{uk}]$

behaves like $[\Pi_{\Gamma^{in}}]$ and can be included in it.

As an example, regarding cylinders, their maximal number of external loops is two. Then, with the eq. (5.2), $[\Pi_{\Gamma^{ex}}]$ is expressed as:

$$[\Pi_{\Gamma^{ex}}] = [\Pi_{\Gamma^{ex}_{(1,2)}}] = ([\Pi^{slf}_{\Gamma^{ex}_1}] \cap [\Pi^{slf}_{\Gamma^{ex}_2}]) \cup \Pi^{LS}_{\Gamma^{ex}_{(1,2)}}. \tag{5.4}$$

If the number of internal loops is smaller than two, eq. (5.4) has to be filtered with the set of symmetry planes for the internal loop but there is no Π_{LS} plane involved in $[\Pi_{\Gamma^{in}}]$. However, if the number of internal loops is greater than or equal to two, there is no criterion for filtering the symmetry planes. $[\Pi_{\Gamma^{in}}]$ can be bounded by the union of the symmetry properties of each pair of internal loops. As a result, eq. (5.4) changes to:

$$[\Pi_{\Gamma^{in}}] \subset \bigcup_{\substack{i=1\ldots n\\ j=1\ldots n\\ i<j}} [\Pi_{\Gamma^{in}_{(i,j)}}]. \tag{5.5}$$

Finally, for a surface having two external loops and several internal loops, the symmetry planes combine as follows:

$$[\Pi_{\partial F}] \subset ([\Pi_{\Gamma^{ex}}] \cap \left(\bigcup_{\substack{i=1\ldots n\\ j=1\ldots n\\ i<j}} [\Pi_{\Gamma^{in}_{(i,j)}}] \right)), \tag{5.6}$$

$$[\Pi_{\partial F}] \subset [\left(([\Pi^{slf}_{\Gamma^{ex}_1}] \cap [\Pi^{slf}_{\Gamma^{ex}_2}]) \cup \Pi^{LS}_{\Gamma^{ex}_{(1,2)}} \right) \cap \left(\bigcup_{\substack{i=1\ldots n\\ j=1\ldots n\\ i<j}} ([\Pi^{slf}_{\Gamma^{in}_i}] \cap [\Pi^{slf}_{\Gamma^{in}_j}]) \cup \Pi^{LS}_{\Gamma^{in}_{(i,j)}} \right)]. \tag{5.7}$$

The boundary loops of F are strong symmetry constraints. There are two different boundary loop configurations: external ones Γ^{ex} and internal ones Γ^{in}. Reference surfaces having at least one external loop, such as planes, cylinders and cones, the CSPs of F are constrained by the external loops first. If the number of internal loops is less than two, symmetry planes are defined by the intersection between external and internal symmetry plane groups only, i.e. there is no Π_{LS} plane. Otherwise, the symmetry planes are defined by the intersection between external loops symmetry planes and the union of internal loops symmetry planes including the Π_{LS} planes belonging to both sets. If F has no external loop, the internal loops are the only constraints.

Right now, the symmetry planes associated with loops stand for symmetry information attached to a face boundary as well as neighboring faces adjacent to edges and vertices of the loop. As a result, it incorporates symmetry properties of these faces:

Π_O and Π_{LB} symmetry planes. For example in Figure 5.9, a boundary loop of the cylinder is a planar curve and it is an ellipse. The ellipse has two symmetry planes orthogonal to the plane containing this curve. Back to the symmetry properties of a cylinder, valid symmetry planes either pass through the axis or are orthogonal to it. In this case, the symmetry plane of the ellipse coinciding with its minor axis is invalid. Consequently, the combination of symmetry constraints of faces with loop structures reduces the number of CSPs compared to independent processing of loops and surfaces.

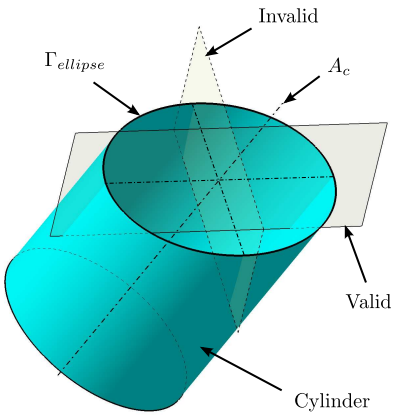


Figure 5.9: Combination of face and its boundary loop symmetry properties.

5.4 Self symmetry planes of a boundary loop

Based on the discussion in section 5.3, the symmetry plane set of ∂F is characterized by the analysis the boundary loops. There are two different symmetry plane sets: self symmetry planes $[\Pi^{slf}]$, relying on O-CSPs and LB-CSPs, and a new category of CSP: $[\Pi_{LS}]$. The methods for detecting the different symmetry planes differ. This section concentrates on $[\Pi^{slf}]$ detection.

With the reflective symmetry definition and self symmetry plane description, if a loop Γ is symmetric with respect to a plane Π, there has to be entities, i.e. edges or vertices, belonging to Γ and intersecting with Π where CSPs exist and are of type either O-CSP or LB-CSP. It means that if the symmetry properties of all the entities belonging to Γ are collected, $[\Gamma]$, the effective symmetry planes of F exist in this set. This principle conforms to the divide phase stated at the beginning of this chapter. Hence, the purpose is to analyze all the entities of a loop to generate a CSP set effectively containing the real symmetry planes of this loop.

Regarding the boundary model resulting from chapter 4, hypergraphs are created from the input B-Rep model to describe the object with entities intrinsic to the object

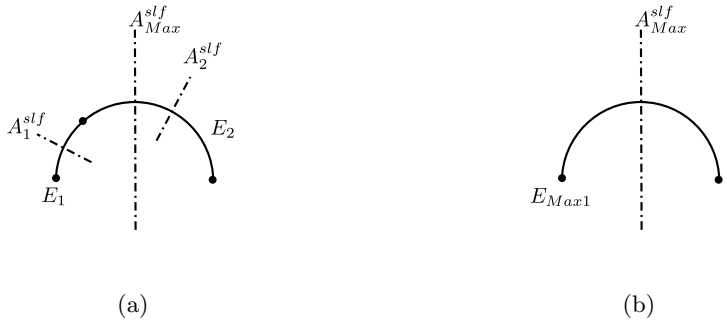


Figure 5.10: Reasons for referring to the maximal edge concept. (a) without maximal edge, each edge has its own symmetry property but not the real one, (b) the symmetry property of maximal edge can lead to real symmetry.

symmetry properties. There, boundary loops contain edges and, often, vertices. After maximal boundary generation, faces and edges are maximal point sets. In this case, the start and end vertices of maximal edges are the locations where the surface continuity of the shape is changing, i.e. vertices indicate changes in reference surfaces. As mentioned in section 2.3, if a vertex is located on an edge at a C^∞ point, it removes the intrinsic symmetry properties of this edge: in a plane, the self symmetry axes A_i^{slf} of the edges E_1, E_2, don't describe the correct symmetry properties of the maximal edge, A^{slf}_{Max} (see Figure 5.10). To obtain the symmetry properties of a sequence of adjacent edges, such a sequence has to be processed specifically and the self symmetry of the reflective symmetry of each edge are useless, see Figure 5.10a. Maximal edges are therefore necessary.

On a complementary basis, after the maximal edge generation, because of the continuity between two adjacent maximal edges differ, if these edges were merged, a symmetry plane Π of this maximal edge combination would not be attached to their common vertex since it is of type Π_{LB}. As a result, on the one hand, this new edge is regarded as one entity but, on the other hand, this edge contains a discontinuity generated by three surfaces at least, so Π cannot be identified as a symmetry plane. Consequently, combinations of edges have to coincide with their reference faces to reflect their symmetry properties: maximal edges defined at chapter 4 are sufficient. This analysis shows that maximal edges convey intrinsic symmetry properties of an object and justify the maximal edge generation process.

5.4.1 Symmetries defined through a Candidate Symmetry Axis (CSA)

In some cases, the intersection curve bounding a face can be a circle (for a plane, a cone or a sphere) or a set of circles having centers lying on a straight line L orthogonal

to their planes. In these configurations, the number of symmetry plane is infinite because the bounded faces involved are rotational symmetric. Indeed, L coincides with the reference face axis (for cylinders, cones, tori) or center (for spheres) and any plane containing L is a reflective symmetry plane and there is an infinite number of such planes. Such a configuration is denominated axisymmetry.

Among the five reference surfaces, four are surfaces of revolution. Consequently, when a plane is orthogonal to an axis of revolution of either of the four revolving surfaces or, alternatively, if two axes of revolving surfaces coincide, if the intersection curves producing a reference face form a set of full circles, these faces are axisymmetric. Each of their corresponding reflective symmetry planes is a *Candidate Symmetry Plane*. In this case, the concept of CSP is turned into the concept of Candidate Symmetry Axis (CSA): a means to characterize the fact that an infinite number of CSPs is attached to these axisymmetric faces.

Axisymmetry is a strong property contributing to the symmetry analysis algorithm. In other symmetry detection algorithms [36], [35], [40], [44], [61], [62], axisymmetry is impossible to detect rigorously with point set models or mesh based ones. They are restricted to axisymmetry approximation only.

5.4.2 O-CSP of a maximal edge through the analysis its two adjacent faces

Let us consider the maximal edge E, the intersection curve between two adjacent reference faces F and F_a; F and F_a are maximal faces. F and F_a are simple analytic surfaces but E has no simple analytic properties, in general. F and F_a interact commutatively with each other, i.e. considering the intersection of F with F_a, or the opposite, produces the same edge E. Hence, F can be arbitrarily taken as target surface for the current analysis. F and F_a define the smallest possible interaction producing an intersection curve E. In a general configuration of a B-Rep model, intersection curves between faces are bounded by other surrounding faces. The effect of these faces will be addressed later at section 5.4.4.

Symmetry properties of E take place at locations defined with curvature and torsion extreme and reflective symmetry planes are contained in the Frenet reference plane defined by the normal and bi-normal vectors at these extreme. As a result, these symmetry planes can be said 'orthogonal' to E, hence the designation of O-CSP, which the main focus of this section.

The symmetry properties of the maximal curve E can be either addressed as a stand alone entity or characterized by the symmetry properties of its adjacent faces F and F_a. On the one hand, if E is analyzed as a stand alone entity, one of the purposes of this analysis is to detect E self reflective symmetry planes. However, E has no simple equation and must be discretized to extract some symmetry properties, which

requires discretization parameters and will be less robust than analytical treatments. In addition, studying E as a stand alone entity does not take into account the symmetry constraints deriving from F and F_a (see the beginning of section 5.4). Another limitation in studying E on its own holds in the fact that under specific locations of F and F_a, E becomes a planar curve and can be the locus of a bisector plane between F and F_a: such a global property would be difficult to extract through a digitized representation of E. On the other hand, if the symmetry properties of E are derived from that of F and F_a and their relative position, the resulting properties are compatible with E and F, F_a both, including the identification of bisector planes.

Indeed, all the loops bounding F result from an intersection between F and F_a. Here, only one loop is analyzed since the focus is set on single intersection curves.

Finally, studying the different categories of couples (F, F_a) help defining the symmetry properties of the intersection curves E forming ∂F. This, leads to a combinatorial study, as a first approach, and the commutative interaction between F and F_a reduces the combinations studied to those listed in Table 5.2. In a first place, this combinatorial study is reduced to O-CSPs only; bisector symmetry planes will be studied in detail at section 5.6.

F / F_a	Plane	Cylinder	Cone	Sphere	Torus
Plane	1:(P_1, P_2)	2:(P, Cy)	3:(P, Co)	4:(P, Sp)	5:(P, To)
Cyliner		6:(Cy, Cy)	7:(Cy, Co)	8:(Cy, Sp)	9:(Cy, To)
Cone			10:(Co, Co)	11:(Co, Sp)	12:(Co, To)
Sphere				13:(Sp, Sp)	14:(Sp, To)
Torus					15:(To, To)

Table 5.2: Combinations of two reference surfaces.

Unless stated otherwise, symmetry planes are valid for any intrinsic parameter of F (radius, angle,...) and F_a.

Symmetry planes of (F, F_a) intersections

Plane/Plane (P_1, P_2)**:** Because of the existence of an intersection curve E, P_1 and P_2 are not parallel to each other. P_1 and P_2 are bounded by E only, their intersection Γ_{P-P} reduces to a unique straight line. Since P_1 (F) and P_2 (F_a) belong to M_{MAX}, Γ_{P-P} is necessarily bounded to form E. There exists always a symmetry plane Π_1, orthogonal to P_1 and P_2 and located at the midpoint of the extreme points of E (see Figure 5.11).

Plane/Cylinder (P, Cy)**:** Configurations with P and Cy subdivide into three categories (see Table 5.3):

- P is orthogonal to the axis of Cy, A_c (see Figure 5.12a). There, Π_1 coinciding with P could be a symmetry plane, even if P and Cy are infinite, but the

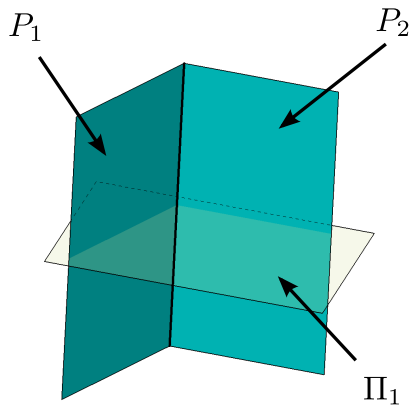


Figure 5.11: Symmetry plane derived from the intersection between two planes.

(F, F_a)	(P, Cy)	(P, Cy)	(P, Cy)
Geometric constraint	$P \perp A_c$	$P \parallel A_c$	P not $\perp A_c$, P not $\parallel A_c$
Π	∞: axisymmetry	Π_1: $A_c \perp \Pi_1$	Π_1:$A_c \subset \Pi_1$

Table 5.3: Configurations of symmetry planes for P and Cy.

neighborhood of Γ_{P-Cy} must be topologically equivalent to three half disks at least. This configuration is non-manifold and cannot be included in a CAD volume. Hence, the figure represents only the relation between the two surfaces: when Cy is effectively bounded by P at E. In this case, E is a circle and it is an edge without vertex to express axisymmetry. A_c is the symmetry axis. Axisymmetry is highlighted on Figure 5.12a with Π_∞;

- Now, if P is parallel to A_c, Figure 5.12b refers to only one intersection curve, as in Table 5.3 to conform to the content of the hypergraphs describing the object boundary, i.e. every edge in $G21$ and $G10$ can be associated with an O-CSP. Indeed, extending P, bounded by E, up to infinity can produce another intersection line (see Figure 5.12d). Consequently, another symmetry plane Π_2 appears. Indeed, Π_2 is not missing, it belongs to the loop symmetry CSP category introduced at section 5.3.2 that will be detailed later on;

- The most general configuration of P generates E as an ellipse (see Figure 5.12c). Only one O-CSP is valid for this configuration.

These configurations are summarized in Table 5.3 where the first line (F, F_a) designates the type of face, then line *Geometric constraint* expresses the geometric location

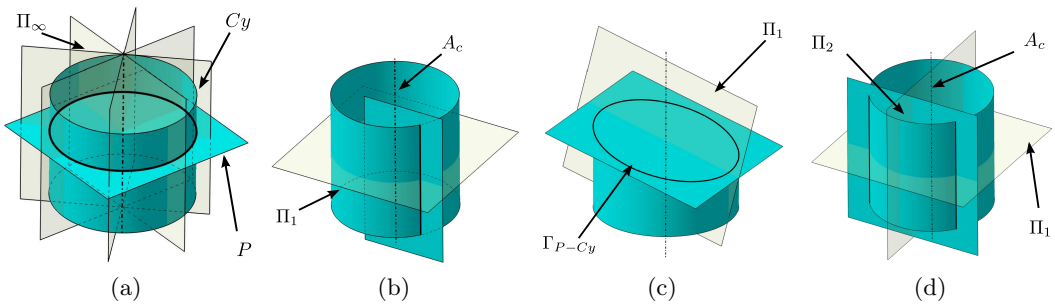


Figure 5.12: (a) P is orthogonal to Cy axis A_c; (b) P is parallel to A_c; (c) P has an arbitrary orientation with respect to Cy differing from (a) and (b); (d) if P is infinite it produces a two segments configuration.

of F_a with respect to F. Π states the maximum number and relative position of symmetry planes. An illustration of each configuration is given in Figure 5.12.

The enumeration of all the possible intersections only considers a full intersection curve. The original surfaces are infinite, though bounded by E. In Figure 5.12a, E is a full circle but to conform to the content of the hypergraphs, E may be bounded by vertices. Likewise the intersection between two planes, the location of these vertices will be taken into account to eliminate some of the CSPs as described at section 5.4.4.

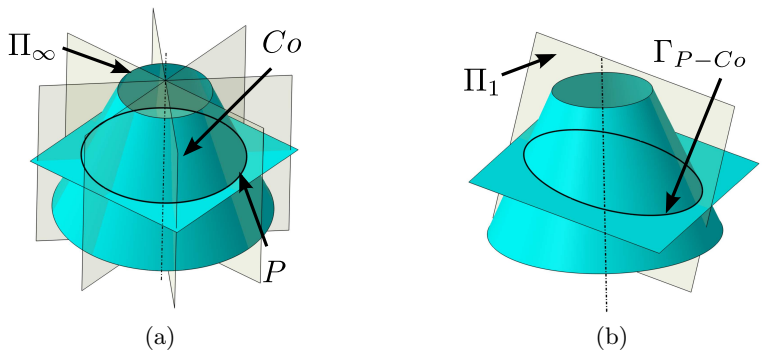


Figure 5.13: (a) P is orthogonal to Co axis, (b) P is not parallel to Co axis.

(F, F_a)	(P, Co)	(P, Co)
Geometric constraint	$P \perp A_c$	$P\ not \perp A_c$,
Π	∞: axisymmetry	Π_1: $A_c \subset \Pi_1$

Table 5.4: Configurations of symmetry planes for P and Co.

Plane/Cone (P, Co)**:** Compared to Cy, the symmetry planes intrinsically attached to Co misses symmetry planes orthogonal to its axis A_c when P goes through

Co apex (see Figure 5.13 and Table 5.4), which explains the difference compared to Cy. If P goes through Co apex, each straight line segment forming the intersection contributes to a unique maximal edge E. The corresponding symmetry plane is still of type O-CSP and can be seen as the limit configuration of P parallel to Co axis. Indeed, it is not attached to a vertex (Co apex) but it is not lost.

The thirteen combinatorial cases left are illustrated in Appendix A for sake of conciseness.

The tables and figures above synthesize all the configurations of the intersection between (F, F_a). However, they only address the complete intersection curves obtained with infinite surfaces (planes, cylinders, cones), full spheres or full tori. In this case, most intersection curves are closed and form one or more loops. Section 5.4.4 will take care of the reduction of intersection curves and the corresponding effect over the existence of CSPs.

5.4.3 CSAs generation

As mentioned at section 5.4.1, a CSA is a special case of O-CSP. O-CSPs should turn into a CSA whenever the intersection curve between adjacent surfaces F, F_a is a circle, meaning that the number of symmetry planes becomes infinite. Within the discussion of the O-CSP generation (see section 5.4.2), many cases create an infinity symmetry planes Π_∞, which indeed are CSAs. Similarly, a surface of revolution intersecting with any other type of reference surface can generate a circular intersection curve and the corresponding respective locations need be specified, which is the purpose of this section to synthesize the configurations and corresponding parameters producing CSAs. To this end, the synthesis follows the combinatorial presentation set up for O-CSP analysis and most of the relative positions of reference surfaces must have coinciding axes to produce CSAs.

Figure 5.14 shows the cases of a plane P intersecting with a cylinder Cy, a cone Co, a sphere Sp and a torus To. When P is orthogonal to the axis of surfaces of revolution, the intersection curve is a circle, possibly reduced to an arc. When P intersects Sp, a CSA is obtained whatever its orientation with respect to Sp.

Figure 5.15 illustrates a cylinder intersecting with a cone, a sphere and a torus. The CSA generation reduces to a constraint of coinciding axes with Co and To whereas it is a matter of coincidence between Cy axis and the center O_s of Sp.

Figure 5.16 describes the axisymmetry configurations of a cone intersecting with a cone, a sphere and a torus, respectively. Axis alignment for Co and To is the CSA constraint and the coincidence between Sp center, O_s, and the cone axis expresses the specific properties of Sp.

Figure 5.17 expresses the axisymmetry constraints between a sphere and a sphere,

a torus as well as the last configuration involving two tori. Indeed, all relative positions of spheres produce a CSA whereas its center O_s must coincide with To axis to generate a CSA. Two intersecting tori produce a CSA when their axes coincide.

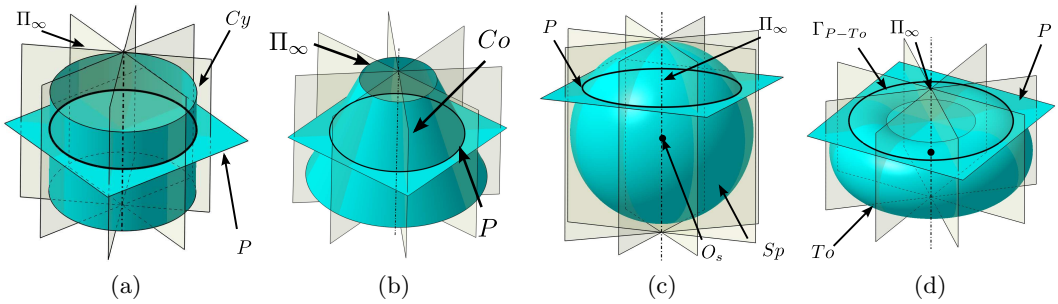


Figure 5.14: Configurations producing CSAs between a plane and a cylinder (a), a cone (b), a sphere (c) and a torus (d).

The set of configurations synthesized complements the set generating only a finite number of O-CSPs generated at section 5.4.2. Altogether, these sets span all the symmetry configurations where a CSP is initiated at an edge and cuts this edge.

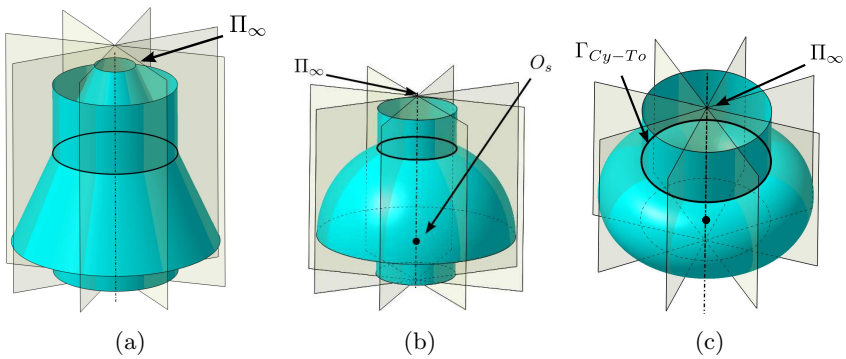


Figure 5.15: Configurations producing CSAs between a cylinder and a cone (a), a sphere (b) and a torus (c).

The existence of a CSA for each reference surface implies the existence of bounding edges E as circles. Depending on the type of surface and location of the circles, the reference surface can contain another symmetry plane, orthogonal to the CSA, of type LS-CSP. It must be the case for Cy because it is bounded by two external loops only. This never happen for Co. It can happen for Sp if it is bounded by two circles contained in parallel planes symmetrically located with respect to O_s. It can also exist for To, similarly to Sp when the circles are symmetrically located with respect to O_t.

In all these cases, the CSP is of type LS-CSP and it can be assigned to the reference surface right away with the CSA as a consequence of axisymmetry and surface type

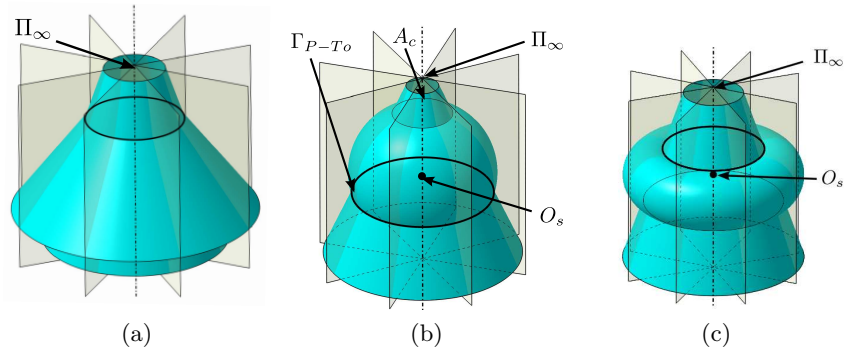


Figure 5.16: Configurations producing CSAs between a cone and a cone (a), a sphere (b) and a torus (c).

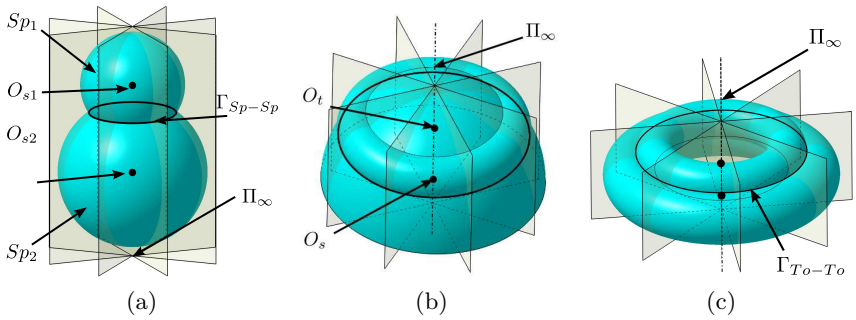


Figure 5.17: Configurations producing CSAs between a sphere and a sphere (a) and a sphere and a torus (b). Configuration producing a CSA between two tori (c).

combination. Taking advantage of this combination speeds up the divide phase. All configurations with surfaces bounded by two circles can be interpreted as bounds with two external loops, except the plane where one circle in necessarily an internal loop.

5.4.4 Incorporating edge bounds constraints in CSP generation

Because maximal edges of the object boundary form the input of the O-CSP generation process, the hypergraphs bring all the object boundary description to generate the most probable CSPs of the object. Indeed, a real surface F_r of an object M_{MAX} is barely an entire surface F (a sphere or a torus) and it cannot be infinite to define a volume, even though it is an infinite point set. Consequently, intersection curves E_r between these surfaces are only subsets of the intersection curves E described in section 5.4.2. So, the couple of adjacent surfaces (F_r, F_{ar}) of a real object M_{MAX} is a subset of reference surfaces F. Edges E_r are also a part of the intersection curves E used in the previous analysis while still forming infinite point sets. E_r can be either

open and bounded by two vertices V_1 and V_2 or closed with one or no vertex.

As a result, the following conditions hold for any edge E_r of the object boundary described in M_{MAX}:

- V_1 and V_2 form a finite point set that must satisfy the symmetry condition with regard to an O-CSP associated with E_r. Therefore, these vertices must be symmetric with respect to this O-CSP. If not, the entire edge E_r is also not symmetric with respect to this O-CSP. In Figure 5.18a, this condition is illustrated through a cone-torus configuration where two O-CSPs Π_1, Π_2, may exist. E_{Co-To} stands for E_r and the relative position of V_1 and V_2 with respect to Π_1 and Π_2 shows that Π_1 can be rejected whereas Π_2 satisfies the condition;

- E_r is closed and has no vertex as boundary from a topological point of view, then E_r coincides with E and the content of section 5.4.2 entirely applies. One such edge in the hypergraphs designates only one connected component of the intersection (F, F_a). Hence, symmetry planes of (F, F_a) that are not O-CSPs, e.g. LS-CSPs, have been already discarded and are not addressed here. Figure 5.18b illustrates this configuration with a cylinder-torus intersection. E_{Cy-To} is the connected component of the intersection taking part to the object boundary. Π_1 and Π_2 are the symmetry planes resulting from the intersection (F, F_a). Indeed, Π_1 is an LS-CSP, hence it is not processed here. Π_2 only is an O-CSP and is the only CSP attached to E_r;

- E_r is closed and has only one vertex V_1 as boundary. Here also, E_r coincides with E. The existence of only one vertex reflects a configuration where the maximal face/edge generation cannot remove this vertex. It is typically the case when (F_r, F_{ar}) are crossing each other (see section 4.6) and this crossing configuration reflects (F_r, F_{ar}) locations where they are tangent to each other at a point. This point is represented by vertex V_1. The content of section 5.4.2 applies and shows that V_1 lies always in an O-CSP: V_1 is not producing new constraint about O-CSP generation. If (F_r, F_{ar}) ate not crossing each other, the manifold vertex split operator combined with edge merge removes V_1, which brings us to the previous configuration.

As depicted at section 5.4.2 all the symmetry planes originated from (F, F_a) do not all belong to the O-CSP category or, more precisely, they do not always belong to the O-CSP category, depending on the relative position of F and F_a and of their intrinsic parameters (normal, radius, ...). The category change of a CSP essentially reflects topological changes in the intersection between F and F_a. This has been characterized in section 5.4.2 with the concept of stability of a CSP, i.e. an O-CSP is stable if, within the configuration of (F, F_a) where it appears, it always exists whatever the intrinsic parameters of F and F_a, as long as $F \cap F_a \neq \phi$.

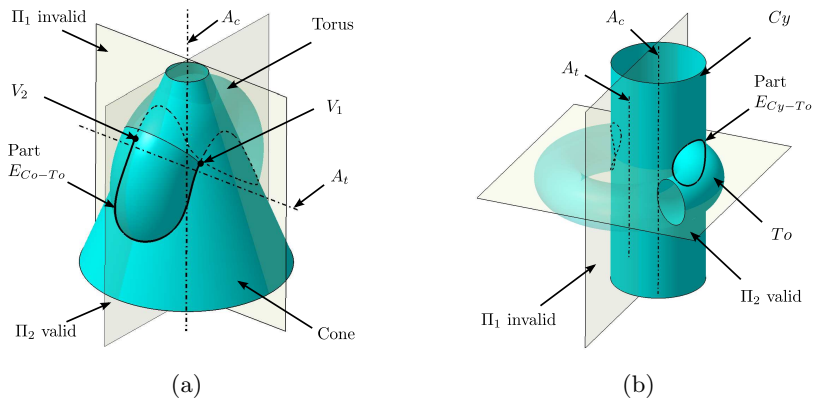


Figure 5.18: Illustrations of edge bounds constraints: (a) is a (Co, To) intersection bounded by V_1 and V_2, (b) is a (Cy, To) intersection where one loop only defines E_{Cy-To}.

When a symmetry plane is not stable, it is connected to the variation of the number of connected components in $F \cap F_a$, hence O-CSPs evolve into LS-CSPs. They are not lost but processed as LS-CSPs. In between the modification of the number of components, lies singular configurations where surfaces are tangent to each other. There, F and F_a are crossing each other and the O-CSP becomes an LB-CSP prior to an LS-CSP. Anyhow, there no symmetry information loss.

The O-CSP set is collected from all the possible orthogonal symmetry planes of the object boundary described by maximal faces and edges. O-CSPs being attached to maximal edges, they can be collected by scanning all these edges in hypergraphs $G21$ or $G10$. If E is a loop, it appears as an isolated node in $G10$ to be distinguished from those bounded by vertices. If E is a loop edge, it appears as a dangling edge in $G10$ and can be distinguished from the other configurations.

It has to be observed that maximal edges bounded by two vertices cannot be attached to more than one O-CSP, Π. Indeed, if $\vec{n}$ is the normal of Π, and V_1, V_2 are symmetric with respect to Π: $\vec{n} \wedge \overrightarrow{V_1V_2} = \vec{0}$. Hence, Π is unique. If E is a loop, i.e. it has no vertex, its maximal number of O-CSPs, n_o, cannot exceed two, as demonstrated by enumeration of the (F, F_a) combinations, when n_o is finite. Otherwise, n_o is infinite and expresses axisymmetry. If E is a loop edge, i.e. it has one vertex V_1, it is assigned only one O-CSP that contains V_1 because V_1 neighborhood is C^0 only and its two tangent vectors define only one symmetry plane. Let N_E, be the total number of maximal edges of an object boundary, $N_E = n_{el} + n_{eb} + n_{le}$ where n_{el} is the number of edges defining loops, n_{eb} the number of edges bounded by vertices and n_{le} the number of loop edges. Then, the maximum number of O-CSPs for this object is bounded by: $N_{O-CSP} \leqslant (2n_{el} + n_{eb} + n_{le})$.

5.4.5 Generation of loop bisector symmetry planes (LB-CSP) of a face by analyzing the surrounding surfaces at a vertex

A symmetry plane Π is not only appearing in the middle of a maximal edge, it can also appear at a vertex V of M_{MAX}. In this case, Π acts as a bisector plane of a maximal face F attached at V whose neighborhood is defined by:

1) two maximal edges E_1, E_2, each one bounded by two vertices, V being their common vertex (see Figure 5.19a);

2) two loop edges E_1, E_2 sharing their unique vertex V (see Figure 5.19b);

3) a combination of both configurations above (see Figure 5.19c).

Rather than relying on the explicit description of E_1 and E_2, it is referred to their adjacent surfaces, similarly to the analysis performed for the O-CSPs at section 5.4.2. Let F_{a1} and F_{a2} be:

- the faces adjacent to E_1 and E_2, respectively, when these edges fall into configuration 1 above;
- the faces bounded by E_1 and E_2, respectively, when describing configuration 2 above;
- the faces bounded by E_1 and adjacent to E_2, respectively, in configuration 3. The opposite assignment holds also for configuration 3;

their intrinsic parameters and relative position with respect to F entirely characterize E_1 and E_2. Then, for a boundary loop Γ of a face F of M_{MAX}, the plane $\Pi \in F$ passing through a vertex V of Γ is a symmetry plane and is called a Loop Bisector candidate symmetry plane of one loop (LB-CSP) if:

a) Π belongs to the intrinsic symmetry planes of F (see section 5.2);

b) F_{a1} and F_{a2} are of same type, have the same intrinsic parameters and are symmetrically located with respect to Π.

Figure 5.19a is an illustration of an LB-CSP configuration. F is the current surface. E_1 and E_2 are successive edges of Γ, they intersect at V and represent case 1. E_1 is the intersection between F and F_{a1}, E_2 is the intersection between F and F_{a2}.

Now, analyzing configuration 3 above and assuming that E_1 is loop edge face F_{a1}, it appears that the existence of one loop edge adjacent to F indicates the existence of a crossing configuration between F_{a1} and F whereas E_2 is not a loop edge. Therefore,

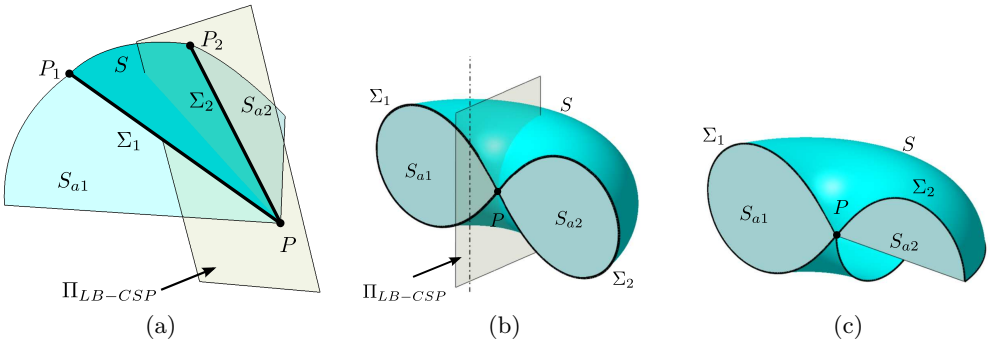


Figure 5.19: A generic configuration of the LB-CSPs. (a) is case 1: a LB-CSP with adjacent edges bounded by two vertices, (b) is case 2: a LB-CSP with adjacent loop edges, (c) is case 3: the combination between case 1 and case 2, but without CSP.

if F_{a2} is either identical or symmetric to F_{a1}, it satisfies condition b above. However, edges E_1 and E_2 don't have the same extent since E_2 is bounded by a vertex without a symmetric bound for E_1. As a result, configuration 3 cannot produce any LB-CSP.

Indeed, because F_{a1} and F_{a2} are of same type and same intrinsic parameters and intersect at least at V, their intersection defines a bisector symmetry plane, BS-CSP (see section 5.3.1), which will be defined more precisely at section 5.6. However, the present configuration shows that a LB-CSP emerges at V when the bisector plane of F_{a1} and F_{a2} coincides with an intrinsic symmetry plane of F at V.

Because there are five types of reference faces addressed, the number of reflective symmetry configurations of two surfaces equals five too (see section 5.6) to define BS-CSPs. Then, combining these five configurations with the five reference faces defining F produces 25 different configurations where LB-CSPs hold. These configurations can be summarized as follows (see also Figure 6.2):

- F is a plane: whatever the category of F_{ai} (planes, cylinders, ...), there exists always an LB-CSP because symmetry planes exist in every direction around a point on a plane;
- F is a cylinder Cy: whatever the category of F_{ai}, there exists an LB-CSP if the bisector plane of F_{ai} either contains the axis A_c of Cy or is orthogonal to A_c;
- F is a cone Co: for all categories of F_{ai}, an LB-CSP exists only if the bisector plane of F_{ai} contains the axis A_{co} of Co;
- F is a sphere Sp: for all categories of F_{ai}, an LB-CSP exists if the bisector plane of F_{ai} contains the center O_s of Sp;
- F is a torus To: for all categories of F_{ai}, an LB-CSP exists only if the bisector

plane of F_{ai} either contains the axis A_{to} of To or is orthogonal to A_{to} and contains its center O_t.

Because the extreme vertices V_1 and V_2 of the two edges E_1 and E_2, respectively, are opposite to their common vertex V (see Figure 5.19a), these vertices can be collected and easily used for symmetry purposes. The locations of V_1 and V_2 can take part to another symmetry constraint expressing the real symmetry of E_1 and E_2 with respect to Π, i.e. V_1 and V_2 must be symmetric with respect to Π. This constraint is similar to the one related to O-CSPs stated at section 5.4.4.

Configuration 2 is not subjected to this constraint and produces a valid LB-CSP with conditions a and b only.

LB-CSP are related to the vertices of M_{MAX}. In the hypergraphs Gij, a boundary loop Γ of a surface F is a data structure derived from $G10$. Regarding Γ, an LB-CSP collection process needs to scan all its nodes to set all its bisector symmetry possibilities. Combining the LB-CSPs with the set of O-CSPs, all the CSPs of Γ are covered:

$$[\Pi_\Gamma] \subset [\Pi_\Gamma]_{CSP} = [\Pi_\Gamma]_{O-CSP} \cup [\Pi_\Gamma]_{LB-CSP}. \tag{5.8}$$

Indeed, generating all the LB-CSPs of M_{MAX} requires scanning all its boundary vertices and, around each vertex, to create all the possible LB-CSPs. Extracting the neighborhood of a vertex V is equivalent to the generation of a dual graph around V (see section 4.6) to create the effective sequence of faces around V. Independently of the intrinsic parameters and relative positions of the surfaces around V, the distribution of surface types, i.e. plane, cylinder, ..., around V brings already clues about the existence of LB-CSPs at V. However, analyzing this distribution with regard to the number of surfaces and the number of surface types, is complex and partly relates to the propagation process of CSPs, which will be addressed at section 6.3 for a first level.

Now, at the global level of the object, having its boundary described with N_V vertices and assuming that there is no more than n_{sMAX} faces around a vertex, an upper bound of the amount of LB-CSPs is given by: $N_{LB-CSP} \leqslant N_V.n_{sMAX}$, which is linear with respect to N_V (see Figure 5.20).

The above analysis of conditions defining a LB-CSP are necessary ones and don't take into account the effective location of the bounded areas of the adjacent surfaces. This location needs to be uniquely characterized. To this end, the concept of orientation index is introduced in the following section.

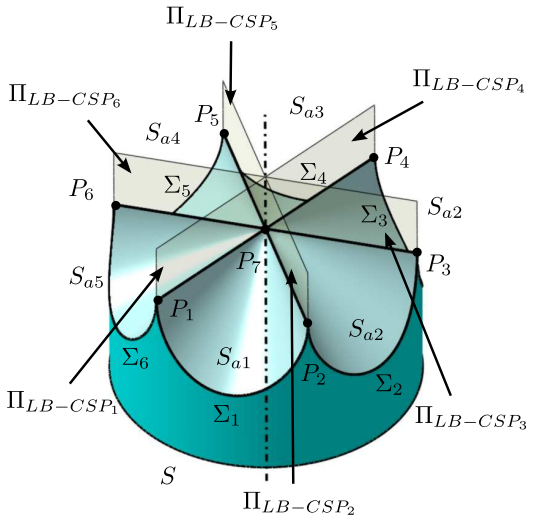


Figure 5.20: A loop combined by 6 edges has 6 LB-CSPs.

5.4.6 Characterizing the used area of a reference face using the orientation index

Indeed, only referring to the intrinsic parameters of the reference faces, their relative locations as well as extreme points of some of their boundary edges is not sufficient to characterize the symmetry properties of two reference faces with respect to a CSP Π. Infinite sets of points thus defined may not coincide with the maximal faces of M_{MAX}. As an example, a hemisphere bounded by a circle as edge loop is not explicitly defined by the sphere parameters, location and circle boundary. At this stage, the corresponding point set can still be either a protruded hemisphere or a hollow one. Such configurations with multiple solutions must be removed and it is the purpose of the orientation index.

Within the range of 3D volumes currently addressed here, available information to characterize surface areas are:

- the orientation of a face, i.e. a subset of a volume boundary that is also a subset of a reference face. This orientation follows the topological properties of the composite surface bounding a volume, which must be orientable. It contributes to the definition of a partition of $I\!R^3$ forming the volume of an object and is independent of the reference faces. This orientation is valid for the whole object boundary;

- the curvature properties of the reference surfaces. They are independent of their spatial location and are defined locally at a point. However, the reference surfaces being simple, their curvature distributions are also simple enough to

extract parameters that can be valid for a whole face.

Considering the curvature properties of the reference faces, they can be related to a bounded subset of a reference face as follows:

- a cylinder has a uniform curvature distribution, i.e. it is valid at any point $\vec{P}(u,v)$ on the cylinder, where one principal curvature direction is defined by its generatrices and the other one is orthogonal to the cylinder axis with a constant radius of curvature. Consequently, a center of curvature at $\vec{P}(u,v)$ can be uniquely defined by the vector originated from $\vec{P}$ to a point Q on the cylinder axis along the normal $\vec{n}$ at $\vec{P}$: $\vec{n} \wedge \overrightarrow{PQ} = \vec{0}$. This is valid for any subset of the cylinder forming a face of B-Rep object;
- a cone has a linearly variable curvature distribution excepted at its apex. One principal curvature direction is always coinciding with the cone generatrices. The other one is orthogonal to the cone axis with a curvature center Q lying on the cone axis and has a linearly variable curvature radius. This is also valid at any point $\vec{P}(u,v)$ of any subset of the cone forming a face. Here also the normal $\vec{n}$ at $\vec{P}$ satisfies: $\vec{n} \wedge \overrightarrow{PQ} = \vec{0}$;
- a sphere has constant radii of curvature and every point on the sphere is umbilic. At every point $\vec{P}(u,v)$, the center of curvature Q is the center of the sphere. This is applicable for any subset of the sphere forming a face as well as: $\vec{n} \wedge \overrightarrow{PQ} = \vec{0}$ where $\vec{n}$ is the normal at $\vec{P}$;
- a torus has a variable curvature distribution. One principal curvature direction is always contained in a plane passing through the torus axis and its corresponding center of curvature Q_1 is located on the circle of average radius. The other curvature direction is always contained in a plane orthogonal to the torus axis and its corresponding center of curvature Q_2 is located on the torus axis. Hence, at any point $\vec{P}(u,v)$, considering the center of curvature Q_1 and the normal $\vec{n}$ at $\vec{P}$: $\vec{n} \wedge \overrightarrow{PQ_1} = \vec{0}$ holds. This invariant does not hold with the quantity $\vec{n} \wedge \overrightarrow{PQ_2}$;
- a plane has a uniformly vanishing curvature distribution and curvature centers Q are located at infinity. The normal $\vec{n}$ at a point $\vec{P}(u,v)$ can be assigned on either side of the plane. The relation: $\vec{n} \wedge \overrightarrow{PQ} = \vec{0}$ holds also at any point of the planar domain defining a face of the object.

Summing up the above analysis shows that curvature related quantities, i.e. PQ or PQ_1, based on cylinder, cone axes, sphere center, torus average radius, plane normal, are invariant for any point of an arbitrary and bounded face, hence they can be used at any point to characterize an entire face. Now, this quantity, which is intrinsic to any subset of a reference surface, can be combined with the face orientation defined for topological purposes to characterize the symmetry properties of faces.

Let $\vec{n}_v$ be the unit normal vector obtained at each point P to define the oriented surface. Because of the orientation property of the object surface, either $\vec{n}_v$ points inside or outside the volume of the object uniformly over the surface. Let $\vec{n}_f$, be the unit normal to a face such that: $\overrightarrow{PQ} = \|\overrightarrow{PQ}\|\vec{n}_f$ or $\overrightarrow{PQ_1} = \|\overrightarrow{PQ_1}\|\vec{n}_f$. Then, the quantity:

$$O_i = \vec{n}_v.\vec{n}_f = \pm 1, \tag{5.9}$$

changes sign whenever two faces of the object have different concavities/convexities (see Figure 5.21). O_i is called the orientation index of a face F and can be used to compare the convexity/concavity of faces evaluated through symmetry properties.

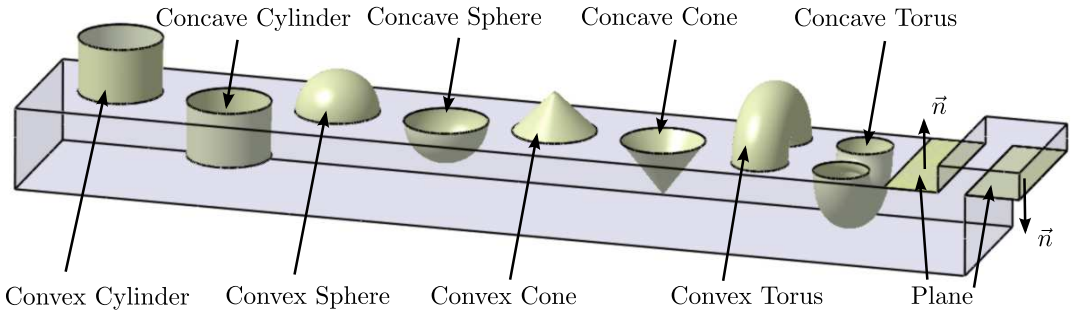


Figure 5.21: A summary of configurations to discriminate between faces of same geometric type. Cylinders, spheres and tori are distinguished using the orientation index O_i. Cones are distinguished using the apex coordinates and oriented axis. Planes are separated with the orientation of M_{MAX}.

However, O_i is not meaningful for a plane since $\vec{n}_f$ can be arbitrarily chosen. Complementary, a plane has a null curvature, hence there is no notion of convexity/-concavity and $\vec{n}_v$ alone is sufficient to compare two planar face since $\vec{n}_v$ determines the position of the interior/exterior of the object. If F_1 and F_2 are the faces to compare and $\vec{n}_{v_1}$, $\vec{n}_{v_2}$ their respective normals: $\vec{n}_{v_1}.\vec{n}_{v_2} = \pm 1$ can be used to discriminate them with regard to their interior/exterior status.

Consequently, the orientation index is a meaningful quantity that can be added to the generation of LB-CSPs to precisely characterize symmetric configurations to be kept. Though the input to the symmetry analysis process is a STEP file and this file format contains information about surfaces that may characterize their concavity/convexity, it is specific to this file format and may depend on the sending system, its generation process of primitives and the way they are combined together. On the one hand, there is no explicit parameter in STEP format describing the concavity/convexity of a surface. This means that a model may be influenced by the users modeling process and its STEP description may be influenced by the modeler. On the other hand, the above approach is self-contained and is effectively discriminating configurations as needed in the present analysis.

5.5 Reflective symmetry planes between loops bounding a surface (LS-CSP)

Regarding the discussion addressed at sections 5.4 and 5.3.6 about symmetry planes generated by multiple loops of a face F, if F is bounded by only one loop, the set $[\Pi^{slf}_{\Gamma^{ex}}]_{CSP}$ is covering all reflective symmetry possibilities. At the opposite and as an example, if F is bounded by two external loops, the symmetry plane $\Pi^{btw}_{\Gamma^{ex}_{(1,2)}}$ is missing (see section 5.3.6).

More generally, when F is bounded by several loops, they can interact with each other and produce new symmetry planes (see section 5.3.2). With each symmetry plane of F, the interactions between loops, either external and/or internal ones, can be subdivided into two distinct categories:

- symmetry planes cutting loops, i.e. more than one loop, through vertices and/or edges;
- symmetry planes separating two loops without having intersection with them.

In the first category, several infinite point sets interact with each other. Because the symmetry planes cut loops at some of their vertices and/or edges, the initial CSPs involved in the identification of these symmetry planes fall into the categories of O-CSPs, LB-CSPs or satisfy axisymmetry conditions, i.e. an edge is attached to a CSA (see section 5.4.1). If the CSPs and CSAs initiated at these loop vertices or edges are effective symmetry planes $[\Pi^{\Gamma}_{i}]$ of the corresponding loop Γ, these symmetry planes must coincide with some $[\Pi^{\Gamma^{ex}}_{j}]$ belonging to the outer loop Γ^{ex} of F. This coincidence establishes a connection between two infinite point sets, each of which corresponding to a loop. Then, facing repetitively these coincidences between loops help connecting together these disconnected point sets.

For a given CSP, $\Pi_{i\,CSP}$, belonging to the previous category and connecting the external loop(s) of F to a maximum amount of its internal loops designated by the set $[\Gamma^{in}_{ci\,MAX}]$, internal loops not falling in the above category belong necessarily to the second one and form the set $[\Gamma^{in}_{i\,LS}]$. The corresponding symmetry planes $[\Pi^{\Gamma^{in}_{i\,LS}}_{j}]$ appear on the basis of couples of internal loops $(\Gamma^{in}_{k\,LS}, \Gamma^{in}_{l\,LS})$ only. These CSPs, called LS-CSPs must interact with symmetry planes $[\Pi^{\Gamma^{ex}}_{k}]$ attached to the external loop(s) of F (see Figure 5.22). As shown at section 5.3.6, symmetry planes of F are contained into the set of planes derived from its external loop(s). More precisely, it can be stated:

$$[\Pi^{\Gamma^{in}_{i\,LS}}_{j}] \in ([\Pi^{\Gamma^{ex}}_{k}] \cup [\Pi^{\Gamma^{ex}}_{(1,2)}], \tag{5.10}$$

showing that LS-CSPs are initiated from the symmetry properties of F external

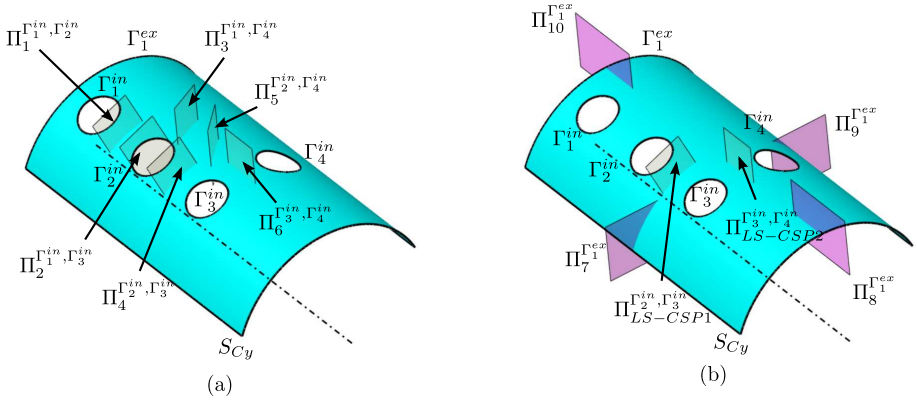


Figure 5.22: Illustrations of LS-CSPs and the corresponding sets of loops and symmetry planes. (a) is combinatorial result between all pairs of internal loops, (b) shows that with the constraints of external loop, only 2 planes are left as LS-CSPs.

loop and symmetry properties intrinsic to the embedding of F in $I\!R^3$ characterized by $[\Pi_{(1,2)}^{\Gamma^{ex}}]$. Indeed, $[\Pi_{(1,2)}^{\Gamma^{ex}}]$ designates the symmetry planes of type LS-CSP generated by two external loops, which applies to cylindrical or toroidal faces only within the current set of reference faces when two of these loops exist. This paragraph can be transposed also to spheres and tori with loops of unknown status when considering these loops as internal ones always compatible with virtual external ones.

In a first place, let us assume that F contains only one external loop Γ^{ex}. Let $[\Pi_j^{\Gamma_{LS}^{in}}]$ be decomposed into $[\Pi_{LS}]$, the set of F symmetry planes derived from internal loop symmetries, and $[\Pi_k^{\Gamma^{ex}}]$, the symmetry planes attached to its external loop $[\Pi_k^{\Gamma^{ex}}] \subset [\Pi_{LS}]$ then, having symmetry planes valid for F implies: $\exists \Pi_{LS\,j} \in [\Pi_k^{\Gamma^{ex}}]$. This observation helps reducing the number of LS-CSPs evaluated to check the symmetry of F.

Considering any valid CSP, $\Pi_{i\,CSP}$, of Γ^{ex}, $\Pi_{i\,CSP} \in [\Pi_k^{\Gamma^{ex}}]$ then, this CSP interacts with the internal loops of F as follows: $[\Gamma^{in}] = [\Gamma_{ci\,MAX}^{in}] \cup [\Gamma_{i\,LS}^{in}]$ and $[\Gamma_{ci\,MAX}^{in}] \cap [\Gamma_{i\,LS}^{in}] = \{\phi\}$. Then, it can be observed that if Card$([\Gamma_{i\,LS}^{in}])$ is odd, then $\Pi_{i\,CSP}$ is not valid for F.

As it appears above, generating LS-CSPs for all internal loops of F can be fairly combinatorial. However, the previous analysis shows that $\Pi_k^{\Gamma^{ex}}$ planes are prominent with respect to Π_{LS} ones since $\Pi_k^{\Gamma^{ex}}$ reduce the amount of internal loop combinations participating to the overall loop symmetry analysis of F.

Now, two configurations emerge from the previous observations:

1) there exist a finite set of CSPs, $[\Pi_k^{\Gamma^{ex}\Gamma_{c\,MAX}^{in}}]$ attached to Γ^{ex} and crossing a maximum number of internal loops $[\Gamma_{ci\,MAX}^{in}]$ of F;

2) there is no CSP crossing internal loops and the outer loop of F is axisymmetric.

Consequently, LS-CSP generation can be addressed either as checking that a CSP is valid for a group of loops or generating all the CSPs valid for a set of internal loops. Both LS-CSP generation processes share common steps that are described hereafter. Starting with a CSP $\forall \Pi_i \in [\Pi_k^{\Gamma^{ex}\Gamma^{in}_{c\,MAX}}]$, Card$([\Gamma^{in}_{ci\,MAX}])$ being even and a loop $\Gamma_k \in [\Gamma^{in}_{i\,LS}]$, Card$(\Gamma_k) = n_{\Gamma_k}$ is the number of vertices or edges defining Γ_k. A first step consists in selecting another loop $\Gamma_j \in [\Gamma^{in}_{i\,LS}]$, then:

- if $n_{\Gamma_j} = n_{\Gamma_k}$;
- and if the sequences of reference faces involved in the definition of Γ_j and Γ_k differ only with circular permutations from each other;

Γ_j can be analyzed further to proceed with the validation of Π_k after their sequence of reference surfaces have been aligned. If one of these conditions is not valid, there is no valid CSP for the current couple of loops and another loop Γ_j can be evaluated. If $\nexists \Gamma_j$ satisfying both conditions, Π_k is not a valid CSP for F.

If Π_k is still a valid CSP then, selecting an arbitrary vertex $V_t^{\Gamma_k} \in \Gamma_k$, $V_t^{\Gamma_k} \in E_t$ identifies the corresponding reference face in the sequence of Γ_k edges. Then, a vertex $V_r^{\Gamma_j} \in \Gamma_j$, homologous to $V_t^{\Gamma_k}$, can be selected and it can be checked whether $V_r^{\Gamma_j}$ is symmetric to $V_t^{\Gamma_k}$ with respect to Π_k or not. If so, this process carries on with the other homologous vertices and, furthermore, with the homologous edges. Processing homologous edges is described hereafter. When homologous vertices and edges are effectively symmetric with respect to Π_k, Π_k becomes a valid LS-CSP for Γ_k and Γ_j.

Now in configuration 2, there is no CSP Π_k available: symmetry planes intrinsic to F become the starting point to generate LS-CSPs for F. Having the external loop of F axisymmetric means that $V_t^{\Gamma_k}$ and $V_r^{\Gamma_j}$ must be symmetric with respect to a plane containing the axis of a cylinder, a cone or a torus if F matches one of these faces types, otherwise it must contain the center of a sphere or be orthogonal to a plane if F belongs to one of these categories. Once the face type specified, the generation process of Π_k follows steps similar to configuration 1 above. Π_k can be entirely defined once a first couple of vertices $V_t^{\Gamma_k}$ and $V_r^{\Gamma_j}$ has been from Γ_j and Γ_k.

Whatever the configuration considered, if the circular permutations generated to align Γ_k and Γ_j are not unique, the above processes have to be reiterated as many times as there exist circular permutations to align Γ_k and Γ_j. When the reference faces in Γ_k are all of the same type and order as in Γ_j, their edges can be processed as follows.

As a first simple case, if Γ_k and Γ_j have no vertex, it means they reduce to edge loops that have to be compared to Π_k. F and the two adjacent faces F_{ai} of two edges of Γ_k and Γ_j are the reference entities. If F_{ai} satisfy the reflective symmetry criterion

with respect to Π_k, indeed Π_k defines a bisector plane (BL-CSP). The corresponding configurations are defined in section 5.6. In addition, Π_k satisfies the intrinsic symmetry constraints originated from F (see Figure 5.23). As a result, Π_k is an LS-CSP.

Now when Γ_k and Γ_j contain vertices, homologous edges are processed similarly to the above paragraph.

Because coordinate comparison for symmetry properties is faster than comparing faces parameters and location, the traversal of vertices is faster than edges. That is why vertices are processed first.

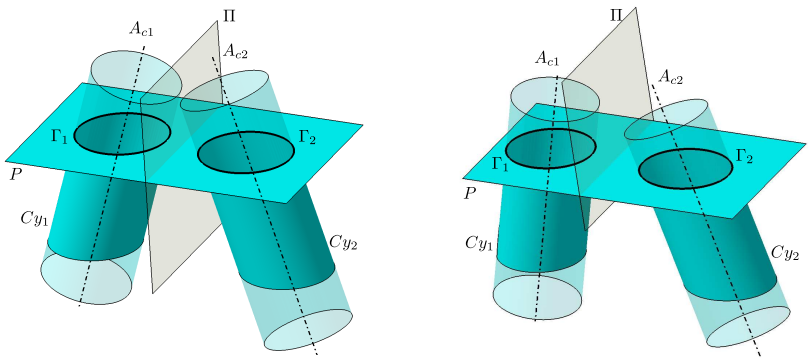


Figure 5.23: Reflective symmetry plane Π derived from loops Γ_1 and Γ_2 without vertex: (a) Cy_1 and Cy_2 satisfy the symmetry constraints with respect to Π. Π satisfies also the symmetry constraints of P, hence Π is an LS-CSP, (b) Π does not satisfy these symmetry constraints.

So far, F has been considered bounded by a single external loop. Processing reference surfaces owning two external loops can be achieved as follows. If $[\Pi_k^{\Gamma^{ex}}]$ designates the set of CSPs attached to Γ^{ex} when F contains only one external loop, now it is replaced by:

$$[\Pi_k^{\Gamma^{ex}}] = ([\Pi_i^{\Gamma^{ex_1}}] \cap [\Pi_i^{\Gamma^{ex_2}}]) \cup [\Pi_{(1,2)}^{\Gamma^{ex}}], \tag{5.11}$$

where $[\Pi_i^{\Gamma^{ex_j}}]$, $j \in \{1, 2\}$ are the CSPs of each external loop of F and $[\Pi_{(1,2)}^{\Gamma^{ex}}]$ is the possible CSP of type LS-CSP originated from the interaction between these external loops. Then, using this new expression of $[\Pi_k^{\Gamma^{ex}}]$ reduces this configuration to faces F having a single external loop.

Though the generation of CSPs is rather combinatorial since they are based on couples of loops, the maximal number of LS-CSPs generated from n_I loops depends on the type references surface. Section 5.3.2 showed that n_I loop amount to $2n_I$ LS-CSPs at most, showing that the number of LS-CSPs evolves linearly wiht respect to the

number of loops. The same remark applies to planes, cones and tori when extending the content of section 5.3.2 to sections 5.3.1, 5.3.3, 5.3.4. The only configuration of face where the generation of LS-CSPs can stay combinatorial is the spherical faces with a quadratic number: $\frac{n_I(n_I-1)}{2}$, of LS-CSPs.

The method to generate LS-CSPs can be summarized in two steps after structuring the loops involved in LS-CSPs. The first one relates to topological elements comparison. Loop vertex or edge numbers must be identical. The second step conforms to the analysis of the interactions between loops as described above.

5.6 Bisector symmetry plane between two surfaces (BS-CSP)

Often, an object boundary contains several reference surfaces, hence several loops. In order to detect its symmetry planes, the symmetry properties of loops are collected by detecting its O-CSPs and LB-CSPs. Then, the symmetry properties of loops interacting with each other are collected by detecting their LS-CSPs. However, the previous categories of CSPs don't cover all the symmetry properties of intersection curves. Indeed, when an intersection becomes a planar curve, it can produce another category of symmetry plane, the so-called bisector symmetry plane.

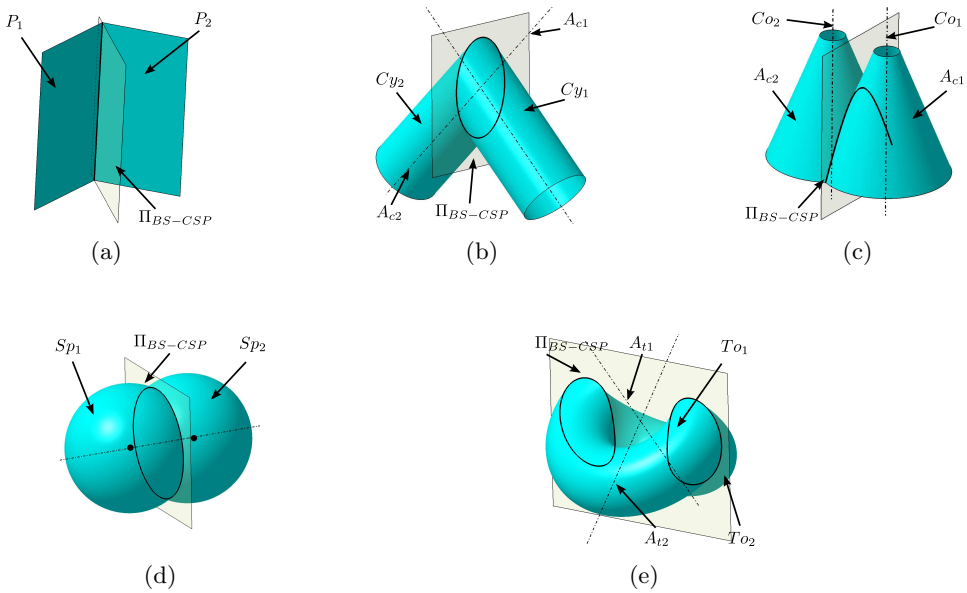


Figure 5.24: Bisector symmetry planes of (P, Cy, Co, Sp, To) and corresponding relative locations.

The symmetry detection in faces F bounded by several loops is covered by the previous section 5.5. There, pair of surfaces forming the loops interact with F and the symmetry between these faces is also a configuration where a bisector plane can generate an LS-CSP. Even though the CSP coming from this configuration is similar to a bisector symmetry plane between the surfaces forming the loops, the object boundary participating to the symmetry property is the surface containing the loops, F, not its adjacent surfaces.

Consequently, a CSP has to be attached, in some way, to a maximal edge or a vertex of the object boundary. This property effectively defines the bisector symmetry planes as a new category of CSP when the intersection curve between two adjacent faces F_{a_1}, F_{a_2} defines a maximal edge of the object boundary. As a result, the analysis of configurations where the adjacent faces to a maximal edge produce a planar intersection curve is sufficient to identify how reference faces F_{a_1}, F_{a_2} can produce Bisector symmetry planes (BS-CSP). Because the objective is to characterize the symmetry properties of M_{MAX}, not only the intersection curve must be planar but its adjacent faces F_{a_1}, F_{a_2}, as infinite point sets, must be symmetric to each other with respect to their intersection curve, which prescribes additional constraints on F_{a_1}, F_{a_2} intrinsic parameters and locations.

Indeed, the analysis of surfaces configurations between two faces F_{a_1}, F_{a_2} has been used to characterize O-CSPs. Here, the constraint of planar intersection curve reduces the category to a set of surface configurations having the same type. In addition, to effectively produce a symmetry plane valid for F_{a_1}, F_{a_2}, their intrinsic parameters must be identical, i.e. same radius for a cylinder or a sphere, same apex angle for a cone, same radii for a torus. Figure 5.24 shows bisector symmetry planes of the five reference surfaces. Under these dimensional constraints, the corresponding relative locations of F_{a_1} and F_{a_2} producing bisector planes can be stated as:

- Planes: any configuration of intersecting planes P_1, P_2;

- Cylinders: cylinders Cy_1, Cy_2 with intersecting or parallel axes. If the axes are parallel, the intersection considered here reduces to only one straight line representing one maximal edge of the object boundary;

- Cones: cones Co_1, Co_2 having the same apex angle and parallel axes A_1, A_2. O_1, O_2 being their respective apices, they must also satisfy $O_1O_2 \perp A_1$ or A_2. Co_1, Co_2 having axes intersecting is another possible configuration. The latter subdivides into two configurations: Co_1, Co_2 meet at their apices or, if L_b is the bisector line of A_1 and A_2, $O_1O_2 \perp L_b$ (see Figure 5.25);

- Spheres: spheres Sp_1, Sp_2, whatever their relative position as long as they intersect with each other;

- Tori: tori To_1, To_2 with coinciding centers O_{T_1}, O_{T_2} and axes intersecting at these points. O_{T_1}, O_{T_2} can be also distinct and the axes of To_1 and To_2 be parallel to each other.

Because BS-CSPs are attached to edges, they can be attached to edges where O-CSPs exist but they are independent of these O-CSPs, i.e. at any given edge, either an O-CSP may exist or a BS-CSP or both.

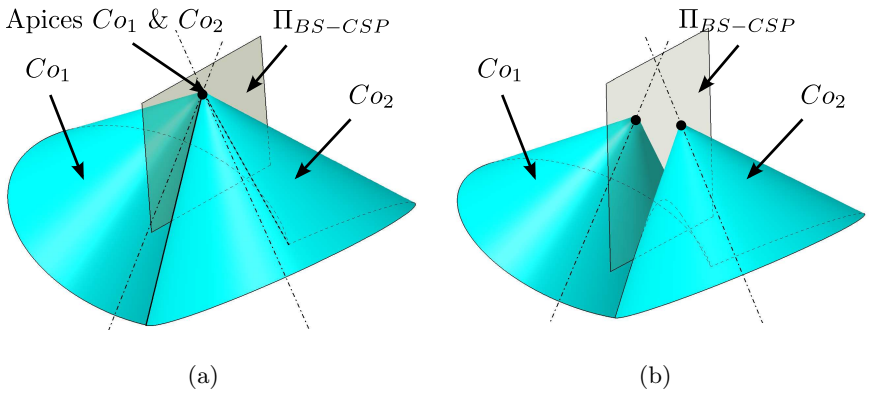


Figure 5.25: Refining configurations producing planar intersection curves between Co_1 and Co_2 where symmetry planes exist: (a) Co_1 and Co_2 intersect at their apices, (b) Co_1 and Co_2 intersect and their apices are symmetric with respect to L_b, i.e. $O_1O_2 \perp L_b$.

5.6.1 Processing multiple solutions of bisector symmetry planes

The BS-CSP of two faces may not be unique depending on the type of reference face. From Figure 5.24, it is clear to show that all reference faces are bounded by a single intersection curve, possibly composed of two loops. When considering the full surfaces, whether bounded or not, some other areas of these surfaces can create a second bisector symmetry plane. This illustrates the multiple solutions of BS-CSP and their ambiguity. As shown in Figure 5.26, the BS-CSP of two planes, two cylinders and two tori are ambiguous. However, depending on the surface embedding, some reference faces may have a unique BS-CSP. A cone, as defined in a STEP file or a B-Rep modeler, is always reduced to one nappe. Even though it is an unbounded surface, section 5.2 shows that an orientated axis can be uniquely assigned to a cone, so a bisector symmetry plane only exists between two oriented axes of Co_1, Co_2, hence BS-CSP is unique. A sphere has no reference axis and the intersection between Sp_1 and Sp_2 produces only one curve. This configuration results in a unique bisector plane, so there is no ambiguous configuration.

Apart from these configurations, to collect the right bisector plane of planes, cylinders and tori using only their intrinsic parameters is impossible. Some other constraints must be added as follows.

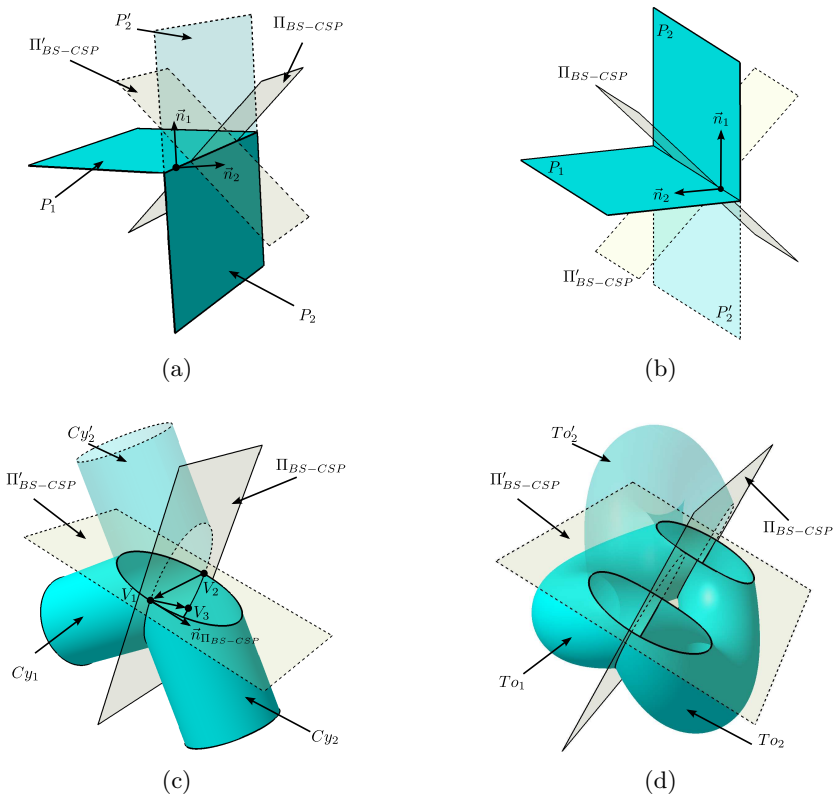


Figure 5.26: Examples of ambiguous configurations of bisector planes: (a) and (b) are planes, their normals provide a constraint to select the appropriate solution; examples of (c) cylinders and (d) tori with bisector planes. The planes containing the intersection curves are the constraints to identify the correct configuration.

In case of BS-CSP between planes P_1 and P_2, the surface orientation of M_{MAX} can be used to generate the right BS-CSP. Their intersection curve is unique and belongs to M_{MAX}. This curve reduces P_1 and P_2 to half planes. Because, P_1 and P_2 are part of a B-Rep model, this boundary is oriented and their unit normals $\vec{n}_1$ and $\vec{n}_2$, respectively, can be used to select the correct BS-CSP. Whatever the convex or concave configuration, the correct bisector plane only appears between the normals of P_1 and P_2, i.e. a vector contained in the BS-CSP is: $\vec{n}_{BS-CSP} = \vec{n}_1 + \vec{n}_2$ (see Figure 5.26a, b).

With cylinders and tori, intersection curves are multiple, i.e. these curves lie into two planes Π_1 and Π_2, whereas the maximal edge E contributing to the object

boundary and defining their BS-CSP varies according to the following configurations:

1) E is C^0 or C^1, bounded by two vertices and no more than one of them is coinciding with the intersection of Π_1 and Π_2;
2) E is C^0 and forms a loop lying in either Π_1 or Π_2;
3) E is C^0 or C^1 and contains a vertex at the intersection of Π_1 and Π_2;
4) E is only C^0 and does not contain a vertex at the intersection of Π_1 and Π_2.

Cases 1 and 2 reflect configurations with a unique symmetry plane. Case 3 is, in fact, a configuration where two symmetry planes can exist. Indeed, the faces F_{a_1} and F_{a_2} are crossing each other. Therefore, the manifold vertex split operator cannot operate in this case. Each intersection curve is split by the vertices located at the crossing points with CSPs. The symmetry planes result from the propagation process and can amount to two planes, thus conforming to the configuration with two symmetry planes. Finally, case 4 represents a configuration where the boundary of F_{a_1} and F_{a_2} is obtained from curves belonging to Π_1 and Π_2. Because F_{a_1} and F_{a_2} are the only faces around a vertex, if any along their boundary, the edges will be merged and produce only one maximal edge, E, partly lying in Π_1 and partly into Π_2. Hence, there will be no BS-CSP attached to this configuration.

This intersection curve is now used to characterize the right CSP or none in case 4. To this end, a set of four non aligned points P_1, P_2, P_3, P_4 is sufficient to uniquely identify either the solution plane or none in case 4. The candidate symmetry planes intersect each other along a straight line L. Any two distinct points on L can be designated as P_1, P_2. Consequently, the identification of the solution plane is left to P_3 and P_4 to check if there is no BS-CSP attached to E (see Figure 5.27). If E belongs to case 4, it contains several curves and one of their endpoint must coincide either with P_1 or P_2. This process must be able to recognise E.

Defining the point P_3 depends on the number of vertices on the intersection curve and on their locations. If at least one vertex exists and does not lie on L, this point becomes P_3, hence the BS-CSP is uniquely defined with P_1, P_2, P_3. If all the vertices available lie on L or if none is available, assuming that the parameterization of the intersection curve is regular, any distinct point from L in its parameter space can be produced to define P_3. Then, this triplet uniquely defines the BS-CSP (see for example Figure 5.26c). If no curve endpoint of E has been found coinciding with P_1 or P_2, E is C^1 and the BS-CSP has been effectively identified. Otherwise, an arbitrary point P_4, located opposite to P_3 with respect to P_1 or P_2 is defined. If P_4 is coplanar with P_1, P_2, P_3 the corresponding BS-CSP can be assigned to E else there is no BS-CSP attached to E because E falls in case 4.

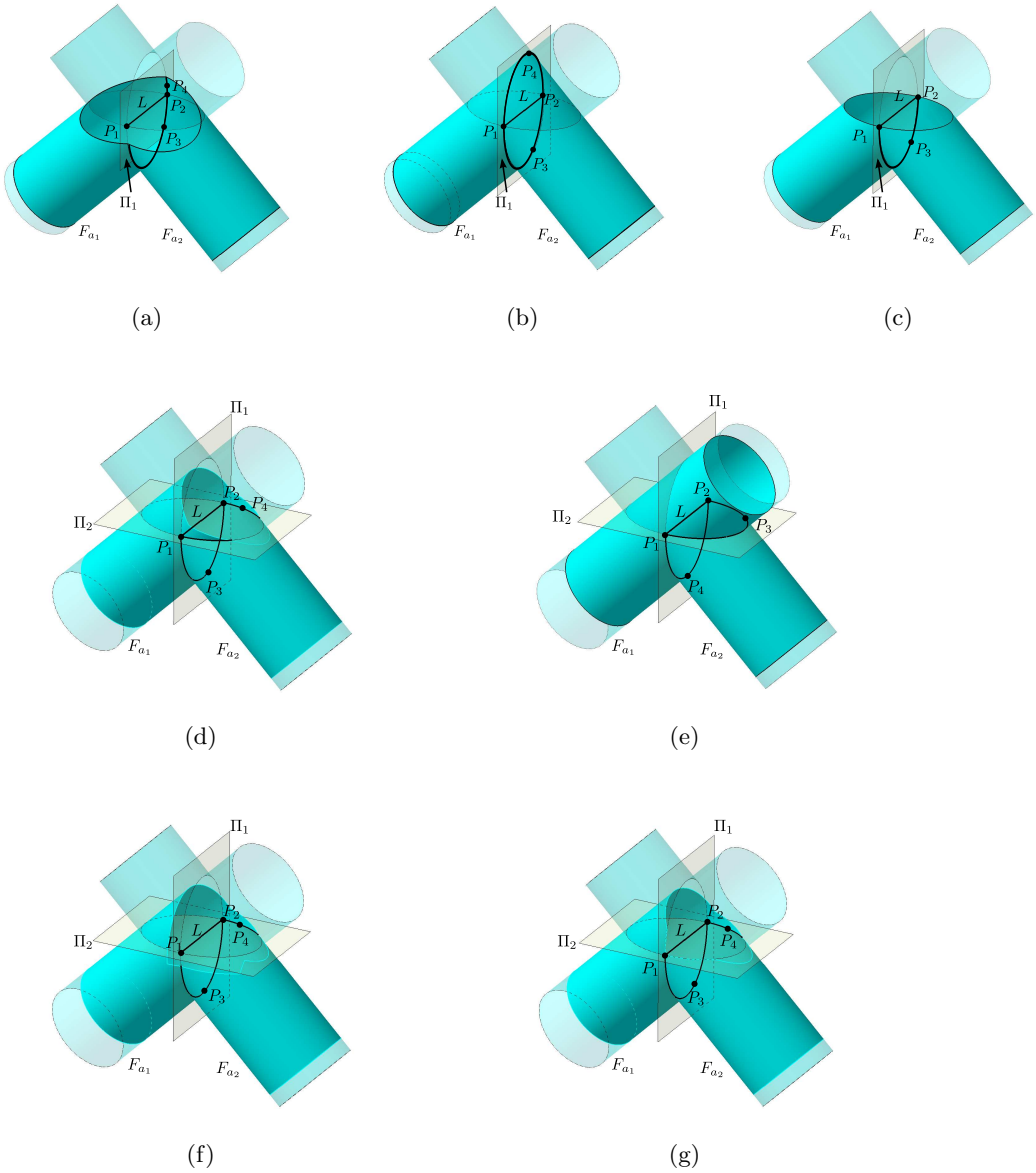


Figure 5.27: Processing multiple solutions of bisector planes to select the right one or assign none if E is not C^0.

5.7 Defining the status of face boundary loops

Sections 5.3 and 5.5 have extensively referred to loops bounding faces as well as their status in terms of internal or external loops. This loop classification can be advantageously taken into account to analyze an object and/or speed up the algorithms. As an example, section 5.3 showed that independently of the number of internal loops, the type of external loop on a cylinder influences the number of symmetry planes: if there is only one external loop, the face cannot have more than one symmetry plane containing the cylinder axis whereas a configuration with two external loops can produce a number of symmetry planes linearly bounded by the number of internal loops. Here, the purpose is to set up the analysis of loops that leads to an unambiguous designation of loops as internal or external.

Let F be the face where loops are being classified. F belongs to one of the five reference surfaces. The status of loops depends on their number and position over a reference face. The number and category of loops are influenced by the other faces F_{a_i}, hence their corresponding reference faces, interacting with F and by their relative position.

Complementary, loops properties must be extracted from loops made up from several maximal edges, i.e. when F interacts with several adjacent surfaces F_{a_i} through a loop.

The two above configurations lead to two complementary approaches: the first one is a combinatorial approach between F and F_a and the second one is the analysis of a composite loop containing several maximal edges.

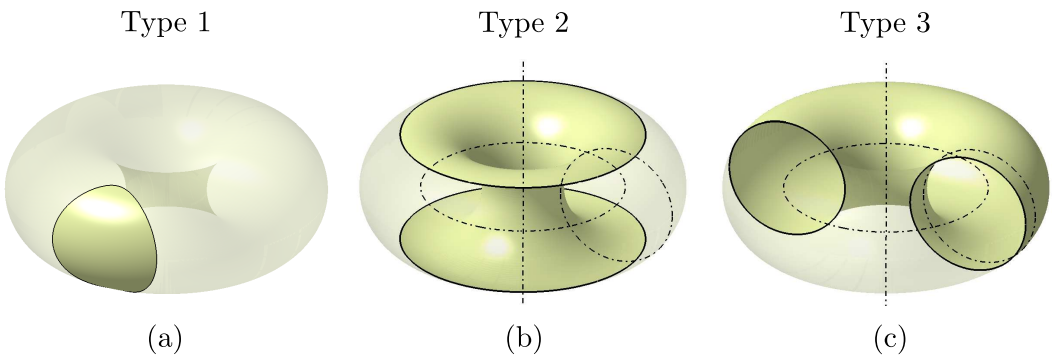


Figure 5.28: Loop type on the surface. (a) is loop type 1; (b) is loop type 2; (c) is loop type 3.

5.7.1 Deriving the status of face boundary loops from maximal edges without vertices

There are 3 types loop on the surface for different surface. The 3 types loop are defined by wether it is around the major revolution axis, the minor revolution axis or not around. As explained in Figure 5.4, different loop type has its own external and internal property if the loop is one edge loop. Loop type 1 is not around any axis, see Figure 5.28a. It can appear on all type surfaces. For plane, cylinder and cone, it can be external loop or internal loop respectively. Loop type 2 is around major revolution axis and all revolute surfaces have it, except plane. A full type 2 loop only can be external loop of the surface. Loop type 3 is around minor axis which it only appears on torus. A full type 3 loop is external loop, too. The edge is the intersection between two surfaces. The combinations of surfaces to be analyzed is summarized in Table 5.5 where the 14 configurations are summarized. Compared to Table 5.2, one configuration does not appear, (P_1, P_2) since the intersection between planes cannot produce a loop.

F_r / F_a	Plane	Cylinder	Cone	Sphere	Torus
Cylinder	1:(P, Cy)	2:(Cy, Cy)	3:(Cy, Co)	4:(Cy, Sp)	5:(Cy, To)
Cone	6:(P, Co)		7:(Co, Co)	8:(Co, Sp)	9:(Co, To)
Sphere	10:(P, Sp)			11:(Sp, Sp)	12:(Sp, To)
Torus	13:(P, To)				14:(To, To)

Table 5.5: Combinations of two reference surfaces where their intersection curves can produce a maximal edge without vertex.

Now, the purpose of this section is to define the geometric constraints of relative surface locations so that their intersection forms a loop that will be represented as a maximal edge without a vertex in the hypergraphs describing M_{MAX}. In addition, loops will be analyzed to classify them as loop types whenever it is meaningful. Throughout this analysis, the tori considered are falling in the configuration where their average radius R is greater or equal to their small radius r. The influence of other configurations, if any, have not been investigated yet. The location and dimensional conditions listed hereunder essentially express how the transition between loop categories operates. It is not intended here to give details about the general process of loop status identification since it is combinatorial. Appendix B gives more details about these processes.

Plane/Cylinder (P, Cy)**:** Because the intersection curve entirely exists as a maximal edge, it exists as a loop as long as the plane normal $\vec{n}$ is not orthogonal to the cylinder axis $\vec{A}_c$: $\vec{n} \cdot \vec{A}_c \neq 0$ (see Figure 5.29). Any loop configuration belongs to the external loop category for Cy.

Cylinder/Cylinder (Cy_1, Cy_2)**:** Without loss of generality, it is assumed that

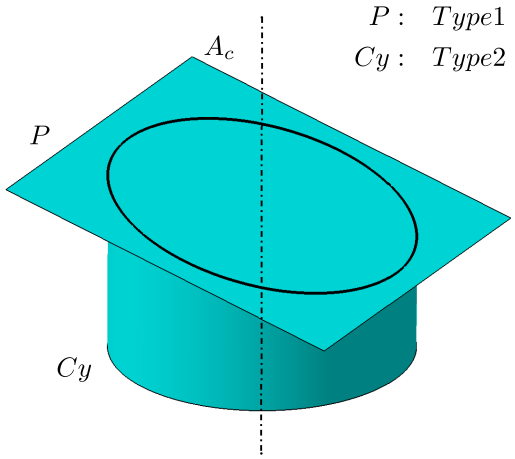


Figure 5.29: Loop formed by a (P, Cy) intersection classified as loop type 2 of cylinder and loop type 1 of plane.

$r_1 > r_2$. In this case, if their axes A_{c1} and A_{c2} intersect $((A_{c1} \cap A_{c2} \neq \phi))$, one loop can be described as: loop type 2 for Cy_2 and loop type 1 one for Cy_1 (see Figure 5.30). If A_{c1} and A_{c2} don't intersect and are not parallel to each other, the loop status is set by the minimal distance between A_{c1} and A_{c2}: d_{12} (see Table 5.6). If A_{c1} and A_{c2} are parallel to each other, they don't generate a loop so this configuration is not relevant.

If $r_1 = r_2$ and the intersection curve between Cy_1 and Cy_2 reduces to only one maximal edge, the loop status is external for Cy_1 and Cy_2 both when the bisector plane simultaneously appears.

When $d_{12} = (r_1 - r_2)$,

(Cy_1, Cy_2)	$(A_{c1} \cap A_{c2} \neq \phi)$		$(A_{c1} \cap A_{c2} = \phi)$	
Geometric constraint	$r_1 \geqslant r_2$	$r_1 = r_2$ non-regular	$d_{12} > (r_1 - r_2)$	$d_{12} \leqslant (r_1 - r_2)$
Loop status	Cy_1: T1 Cy_2: T2	Cy_1: T2 Cy_2: T2	Cy_1: T1 Cy_2: T1	Cy_1: T1 Cy_2: T2

Table 5.6: Configurations of loops with an intersection between Cy_1, Cy_2.

Cylinder/Cone (Cy, Co)**:** The configurations can be structured according to the relative position of the cone apex A with respect to the cylinder axis A_c and the apex angle α. First of all, the position of A is located either inside or outside Cy. This is expressed with:

$$\|\overrightarrow{AP} + (\overrightarrow{AP} \cdot \vec{A}_c) \cdot \vec{A}_c\| < R \tag{5.12}$$

where P is an arbitrary point on the axis of Co, A_c is a unit vector defining the axis of C_o and the inequality expresses the fact that A is inside Cy. Within this relative

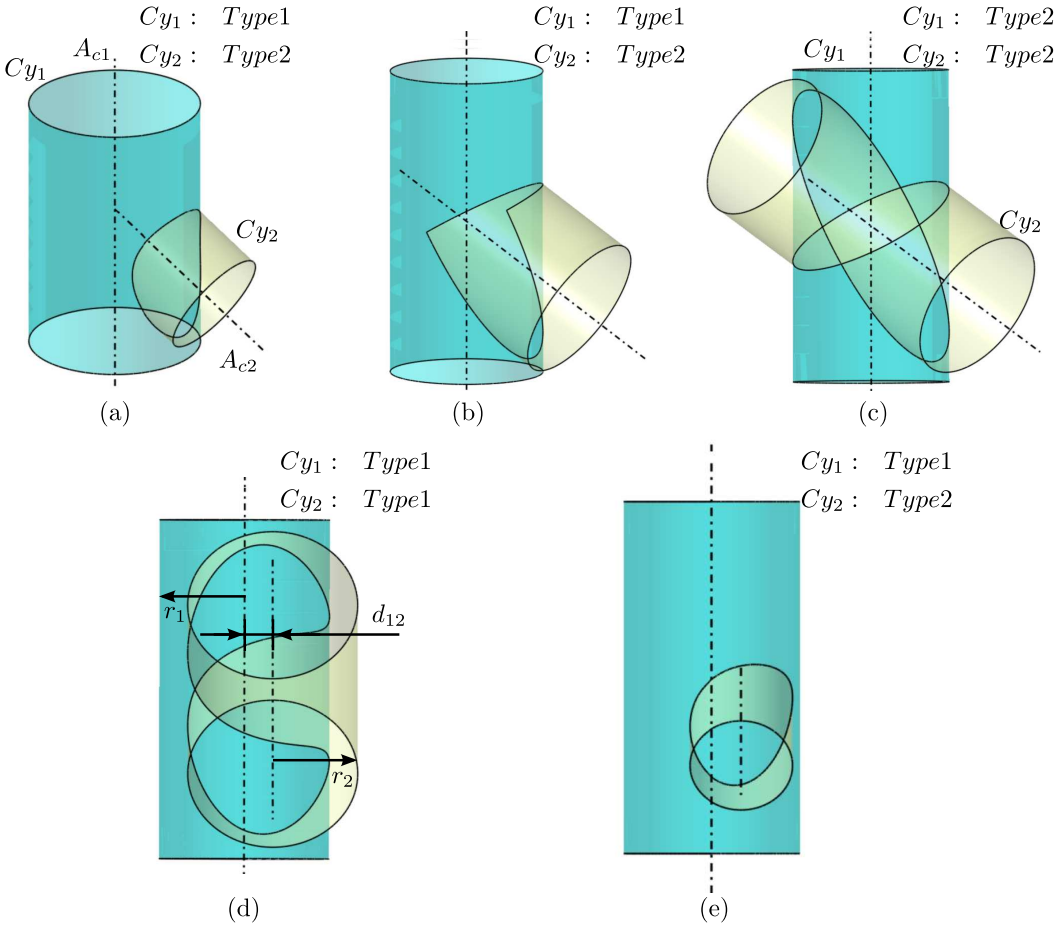


Figure 5.30: Loop formed by (Cy_1, Cy_2) intersections: (a)(b)(c) A_{c1} and A_{c2} intersect, (d)(e) A_{c1} and A_{c2} don't intersect.

location of Co and Cy, the relative angle between Co and Cy is the second parameter, e.g. $\widehat{A_{co}A_c} \geqslant \alpha$ (see Figure 5.31).

Table 5.7 summarizes the major configurations and the corresponding loop status for Cy and Co. It has to be noticed that the configuration where Cy does not contain A is characterized by a transition configuration where Co is tangent to Cy. However, this characterization is subjected to a fair amount of parameters and cannot straightforwardly expressed. The Appendix B gives the details defining the corresponding condition, which can be expressed analytically.

Cylinder/Sphere (Cy, Sp)**:** The loop status for Cy and Sp interaction is monitored by the relative position of the center C_s of Sp with respect to the axis A_c of Cy. It reduces to two categories with the first one:

$$(r_c + d) \leqslant r_s \tag{5.13}$$

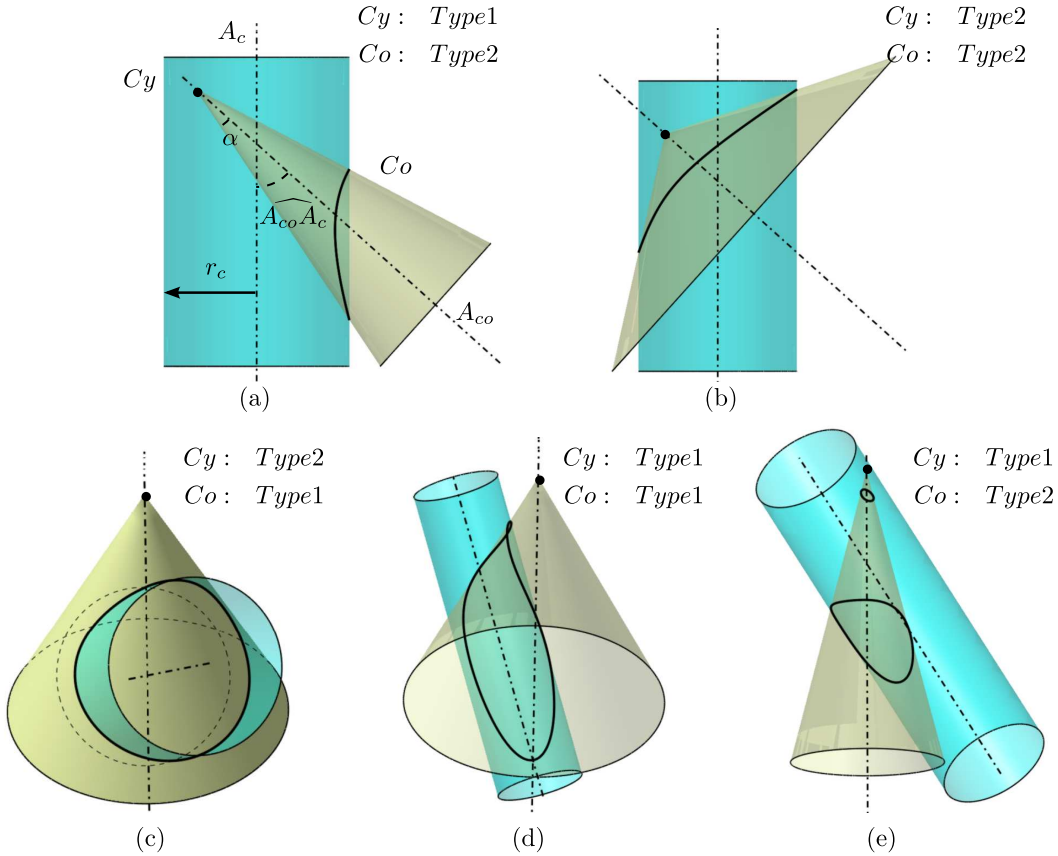


Figure 5.31: Loop formed by (Cy, Co) intersections: (a)(b) Cy contains the apex A of Co, (c)(d)(e) A is outside Cy.

(Cy, Co)	$(A \subset Cy)$		$(A \not\subset Cy)$		
Geometric constraint	$\widehat{A_{co}A_c} \geqslant \alpha$	$\widehat{A_{co}A_c} < \alpha$	Cy inside Co	Cy partly outside Co	Co cross Cy
Loop status	Cy: T1 Co: T2	Cy: T2 Co: T2	Cy: T2 Co: T1	Cy: T1 Co: T1	Cy: T1 Co: T2

Table 5.7: Configurations of loops with an intersection between Cy, Co.

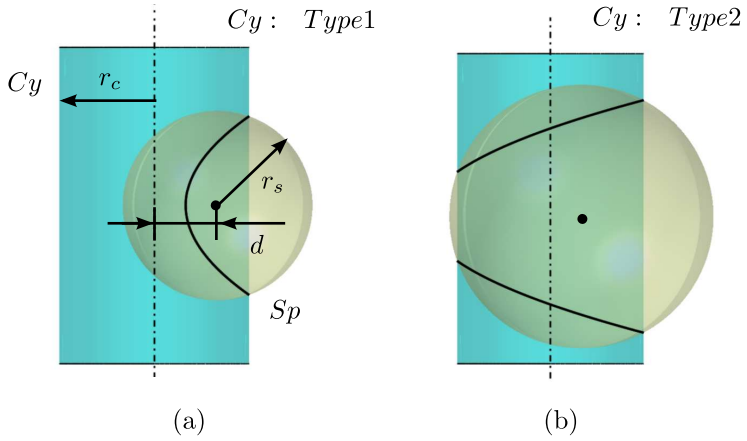


Figure 5.32: Loop formed by (Cy, Sp) intersections: (a) Cy satisfies Eq. 5.13, (b) complementary configuration.

where r_c and r_s are the cylinder and sphere radii, respectively and d, $d \in \mathbb{R}^+$, is the shortest distance between A_c and C_s. The other category is obtained with the complementary inequality and the categories are illustrated with Figure 5.32. There is no loop status for Sp since the loops over this surface cannot be categorized (see section 5.3). Table 5.8 summarizes the loop status for Cy.

(Cy, Co)	Cy 'inside' Sp	Cy 'outside' Sp
Geometric constraint	$(r_c + d) \leqslant r_s$	$(r_c + d) > r_s$
Loop status	Cy: T2	Cy: T1

Table 5.8: Configurations of loops with an intersection between Cy, Sp.

Cylinder/Torus (Cy, To)**:** There are three types of loops over To and two types for Cy (see Figure 5.33a,b,c). Likewise for the interaction (P, To), it is necessary to cover all the configurations generating the different categories of loops over To. Looking at the configuration where the cylinder generates loops of type 1, a cylinder with an axis A_c slanted with respect to To axis A_t is similar to the configuration studied for (P, To). However, the curvature of the cylinder, R_c produces a configuration close to the interaction taking place during a machining operation where the curvature of the tool and the curvature of the machined surface are not independent of each other (see Figure 5.33d,e).

As a consequence, the characterization of tangent configurations becomes more complex and cannot be achieved with an analytical approach because the governing equations become non linear and require a numerical approach. Appendix B gives more details about the corresponding equations governing some tangent configurations. The occurrence of loop type becomes also more complex because multiple

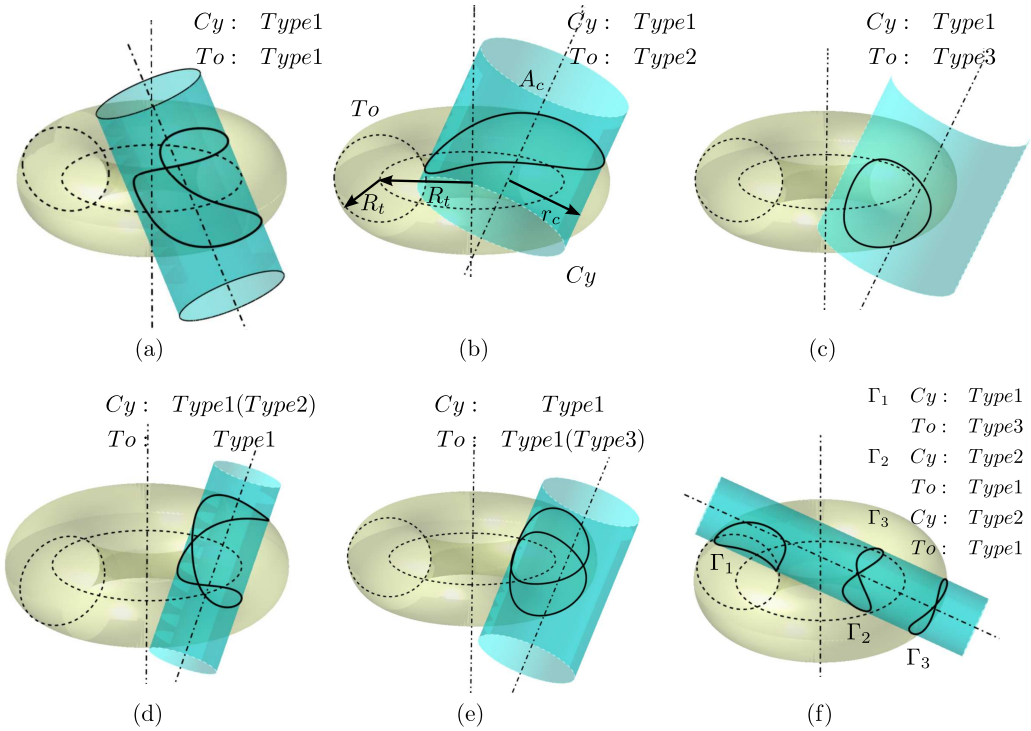


Figure 5.33: (a)(b)(c) An illustration of the different loop types over To generated with Cy, (d)(e) tangent configuration illustrating the curvature interactions between To and Cy to produce different loop types, (f) an example of configuration where different loop types appear simultaneously.

intersection curves can produce different loop types (see Figure 5.33f). Further treatments are required to separate these categories to focus on the right one.

Because the radius of Cy influence the intersection curves, dimensional configurations of To and Cy may not be able to produce some loop types, e.g. if Cy is smaller than the small radius of To, $R_c < r$ there cannot be loops of type 2.

Regarding the definition of loops, the identification of a current configuration uses the dimensions of To and Cy and a reference configuration where To and Cy share some tangent configurations, if possible, with a relative orientation conform to they real orientation. From that configuration, the purpose is to determine the displacements needed to meet the real configuration of To and Cy, the corresponding displacement identifying the corresponding loop type (see Figure 5.34). The cone/cone interaction hereafter gives an example of this process with more details. The remarks about the cone/cone interaction apply also to the To and Cy one.

Plane/Cone (P, Co)**:** Differently from the interaction (P, Cy), intersection curves for (P, Co) can be either closed or open when P forms an angle greater or smaller

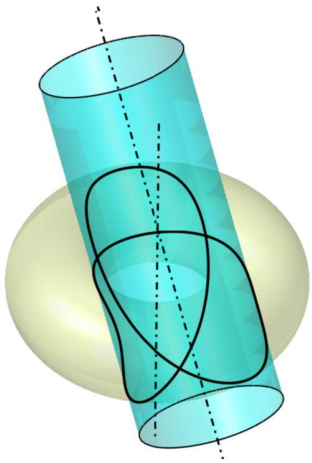


Figure 5.34: Principle of the loop type determination using a reference configuration with a tangent configuration.

than α with respect to the cone axis A_c, respectively (see Figure 5.35). When the intersection curve is open, it cannot define a maximal edge without a vertex; hence this configuration is not addressed here. The conditions for loop configurations reduce to:

$$1 \geqslant \vec{n} \cdot \vec{A}_c > \sin\alpha \tag{5.14}$$

when the plane normal conforms to the orientation of Figure 5.35.

The loop status in that configuration is of type 2 for Co.

Cone/Cone (Co_1, Co_2)**:** The (Co_1, Co_2) can produce three different types of intersection curves: open curves that should be discarded in the present study, closed curves classified either as type 2 loops when they cycle around a cone axis or as type 1 loops otherwise. Open curves are discarded because they cannot produce edge loops is the hypergraphs. Regarding the categories of closed curves, the loop type can be synthesized in accordance with the relative position of an apex, say A_1, with respect to the other cone, i.e. A_1 is either inside or outside Co_2. This observation applies also to A_2 for Co_2. As a result, the number of configurations to distinguish loops ends up to four (see Figure 5.36a).

The two first configurations are based on the fact that A_2 is inside Co_1. Checking the position of A_2 as inside Co_1 can be operated as follows (see Figure 5.36b). Assuming that cones axes A_{Ci} are normalized: $\|\vec{A}_{Ci}\| = 1$ and the quantities:

$$h = \overrightarrow{A_1A_2} \cdot \vec{A}_{C1}, \tag{5.15}$$

$$\cos\beta = \frac{h}{\|\overrightarrow{A_1A_2}\|}, \tag{5.16}$$

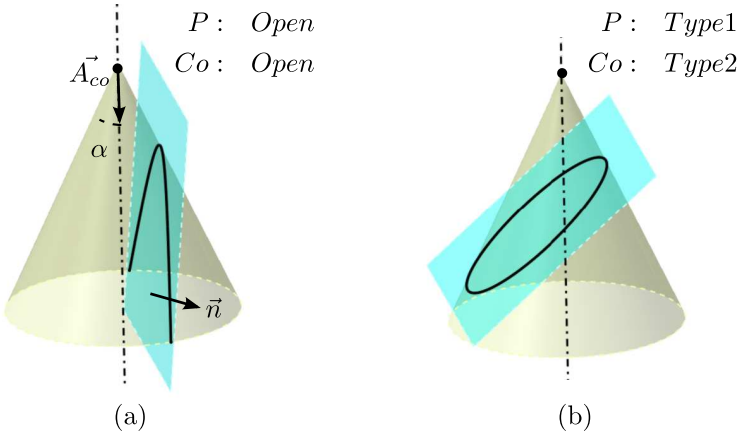


Figure 5.35: (a) intersection curves for (P, Co) with open and closed configurations, (b) Loop formed by a (P, Co) intersection classified as external loop.

are evaluated, then A_2 is inside Co_1 if:

$$h > 0, \text{ and } \beta < \alpha_1, \tag{5.17}$$

where α_1 is the apex angle of Co_1. If A_2 only is inside Co_1, it means the loop is type 1 for Co_1 and type 2 for Co_2. In addition, if A_1 is inside Co_2, the loops are type 2 for Co_1 and Co_2 both. When A_2 and A_1 are both outside Co_1 and Co_2, respectively, two configurations must be distinguished. Without loss of generality, lest us assume that $\alpha_1 > \alpha_2$. Then, the two configurations can be characterized by Co_2 partly penetrates Co_1 or Co_2 entirely crosses Co_1. This distinction is achieved using a reference configuration where the two cones are tangent to each other with an arbitrary orientation and Co_2 entirely inside Co_1 (see Figure 5.37).

From that reference configuration, a distance parameter d can be used to evaluate the difference between this reference configuration and the real one. If the real configuration requires a variation of d so that d increases, Co_2 partly penetrates Co_1 and the loop status for both is of type internal. If the real configuration differs from the reference one when d decreases, it indicates that Co_2 entirely crosses Co_1 and the corresponding loop status is internal for Co_1 and external for Co_2. Table 5.9 summarizes these configurations and loop status.

Indeed, characterizing the reference configuration is no longer analytical (see Appendix B for more details) because computing d is achieved through a non linear trigonometric equation. This numerical requirement illustrates the increased complexity to define the loop status. Arbitrary relative positions of Co_1 and Co_2 is not very common in mechanical components so the generic configuration can be subdivided into simpler ones that can be processed analytically, hence quickly.

Cone/Sphere (Co, Sp)**:** The interaction (Co, Sp) produces two categories of

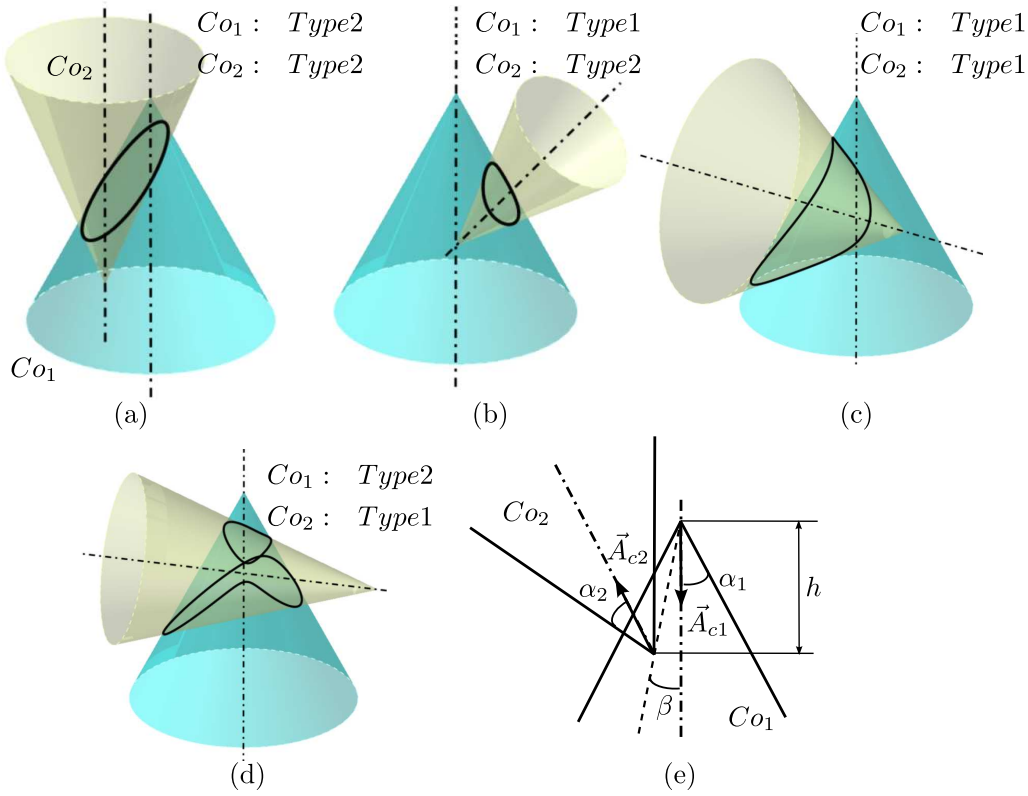


Figure 5.36: (a) A summary of the possible configurations of Co_1 and Co_2, (b) geometric elements to A_2 with respect to Co_1.

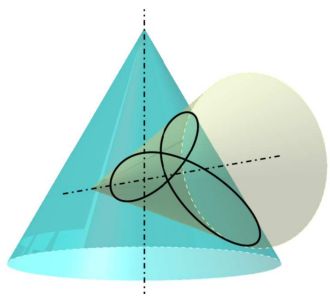


Figure 5.37: Reference configuration of Co_1 and Co_2 in a tangent setting and arbitrary configuration.

(Co_1, Co_2)	apex (ices) inside Co_1 or Co_1		Co_2 'penetrates' or 'crosses' Co_1	
Geometric constraint	A_2 inside Co_1	A_2 inside Co_1 and A_1 inside Co_2	Co_2 partly penetrates Co_1	Co_2 entirely crosses Co_1
Loop status	Co_1: T1 Co_2: T2	Co_1: T2 Co_2: T2	Co_1: T1 Co_2: T1	Co_1: T1 Co_2: T2

Table 5.9: Configurations of loops with an intersection between Co_1, Co_2.

loops for Co. The loop category depends on the singular configurations produced by the relative position of the center C_s of Sp with respect to Co. Sp can be either tangent to Co or in contact with its apex A. This produces three types of singular configurations forming the following conditions using d and h to locate C_s:

- C_s is located inside Co, i.e. $0 < d < h \tan\alpha$, $h \geqslant 0$:

$$h \sin\alpha - d\cos\alpha < R_s < h\sin\alpha + d\cos\alpha, \tag{5.18}$$

  characterizes a loop type 1 and:

$$R_s \geqslant h\sin\alpha + d\cos\alpha, \tag{5.19}$$

  a loop type 2;

- C_s is located outside Co, i.e. $d > h\tan\alpha$, $h \geqslant 0$ and $d > R_s\cos\alpha$, $h > -R_s\sin\alpha$

$$d\cos\alpha - h\sin\alpha < R_s < d\cos\alpha + h\sin\alpha, \tag{5.20}$$

  produces a loop type 1 and:

$$R_s \geqslant h\sin\alpha + d\cos\alpha, \tag{5.21}$$

  a loop type 2;

- C_s is located outside Co within a cone of apex A and angle $(\frac{\pi}{2} - \alpha)$. Inside this cone, C_s satisfies: $0 < d < R_s\cos\alpha$, $h < -R_s\sin\alpha$. Then:

$$R_s > \frac{d}{\sin\beta}, \tag{5.22}$$

  where β is the angle formed by C_s with respect to Co axis. This produces an external loop.

(Co, Sp)	$h > -R_s \sin\alpha$	$h < -R_s \sin\alpha$ and $0 < d < R_s \cos\alpha$ or $h \geqslant 0$
Geometric constraint	$0 < d < h \tan\alpha$ and Eq. 5.18 or $d > h \tan\alpha$ and Eq. 5.20	$R_s > \frac{d}{\sin\beta}$ or Eq. 5.21
Loop status	Co: T1	Co: T2

Table 5.10: Configurations of loops with an intersection between Cy, Sp.

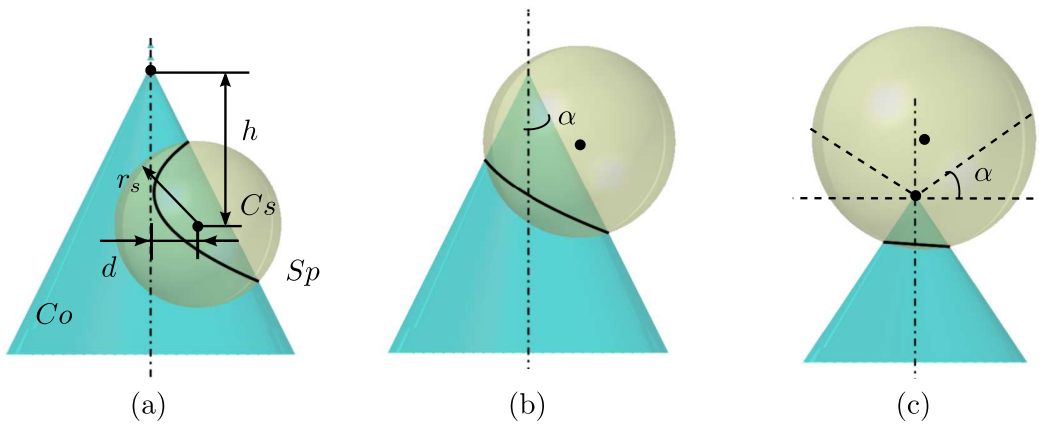


Figure 5.38: (a) loop type 1 formed with C_s internal to Co, (b) loop type 2 generated when C_s is external to Co, (c) loop type 2 when C_s is located inside the cone of angle $(\frac{\pi}{2} - \alpha)$.

Table 5.10 synthesizes these configurations and Figure 5.38 illustrates some typical configurations.

Cone/Torus (Co, To)**:** Similarly to the (Cy, To) interaction, the (Co, To) is rather complex and is not detailed here (see Appendix B). The loop determination shares principles similar to the (Co, To) and the (Co_1, Co_2) ones. The specific features of Co can be also exploited to simplify the classification of reference configurations, e.g. locating the cone apex with respect to To when To is defined as an implicit surface (see Figure 5.39).

The other remarks stated for Cy and To and (Co_1, Co_2) apply also here.

Plane/Sphere (P, Sp)**:** The interaction (P, Sp) reduces to a unique type of loop, whose existence is simply defined by: $d < r$, where r is the radius of Sp and d is the minimal distance between P and the center C of Sp. There is no classification of this loop possible on the sphere, independently of the interacting surface (see section 5.3).

Sphere/Sphere (Sp, Sp)**:** There is loop classification for this configuration since Sp does not produce loop categorization.

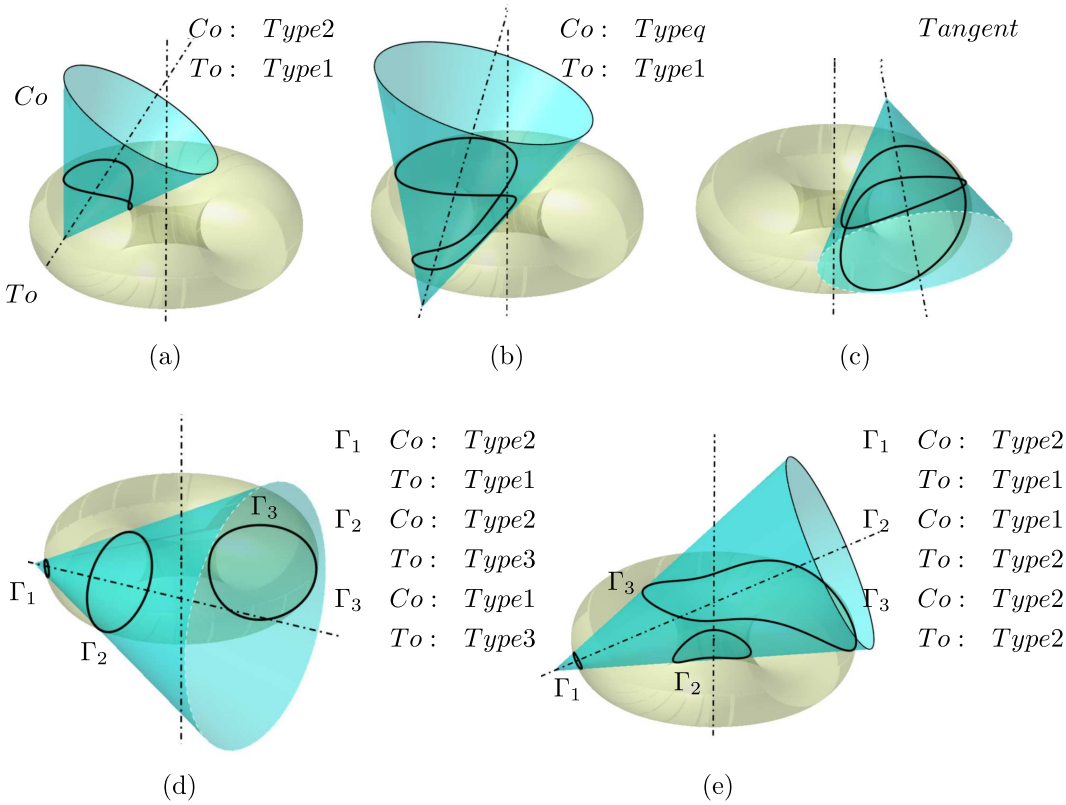


Figure 5.39: Some reference configurations of the (Co, To) interaction.

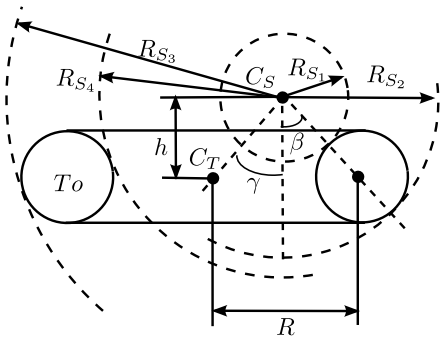


Figure 5.40: The sphere radii configurations of (Sp, To).

Sphere/Torus (Sp, To)**:** Here also, the loop classification addresses only To and can be studied through the relative position of C_s, the center of Sp with respect to To using its radial distance d and height h from the center O_T along the axis A_T. Then, singular positions forming transitions are tangent ones at points with $d > 0$ and $h > 0$. These tangency conditions express:

- based on the angle β defined as: $\tan\beta = \left(\frac{R-d}{h}\right)$, two tangency conditions for R_s are expressed as:

$$d + (R_s - r)\sin\beta = R, \tag{5.23}$$
$$d + (R_s + r)\sin\beta = R; \tag{5.24}$$

- and based on the angle γ defined as: $\tan\gamma = \left(\frac{R+d}{h}\right)$, two other tangency conditions for R_s are obtained:

$$d + R = (R_s - r)\sin\gamma, \tag{5.25}$$
$$d + R = (R_s + r)\sin\gamma; \tag{5.26}$$

- when $d = 0$, the tangency configuration takes place along a circle, which reduces the four above configurations to two governed with the angle β defined as $\tan\beta = \frac{R}{h}$:

$$(R_s + r)\sin\beta = R, \tag{5.27}$$
$$(R_s - r)\sin\beta = R. \tag{5.28}$$

Studying Eqs. 5.23 and 5.24, which produce values R_{s1} and R_{s2}, show that: $0 \leqslant R_{s1} \leqslant R_{s2}$, $\forall d, \beta$. Similarly, Eqs. 5.25 and 5.26, producing values R_{s3} and R_{s4}, are such that: $0 \leqslant R_{s3} \leqslant R_{s4}$, $\forall d, \gamma$. In addition, it can be showed that: $R_{s1} \leqslant R_{s3}$, $\forall d, \beta, \gamma$ (see Figure 5.40).

As a result, two sets of configurations emerge to characterize the loops formed by the (Sp, To) interaction: either $R_{s1} < R_{s2} < R_{s3} < R_{s4}$ (a) or $R_{s1} < R_{s3} < R_{s2} < R_{s4}$ (b). The loop types derived from these inequalities are:

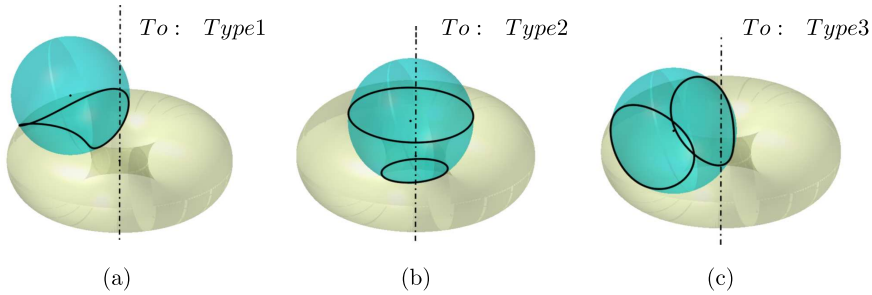


Figure 5.41: Illustration of loop types resulting from the interaction (Sp, To): (a) an example of loop of type 1, (b) an example of loop of type 2, (c) an example of loop of type 3.

- loop type 3 with $R_s \in [R_{s1}, R_{s2}]$, loop type 2 with $R_s \in [R_{s2}, R_{s3}]$, loop type 3 with $R_s \in [R_{s3}, R_{s4}]$;
- loop type 3 with $R_s \in [R_{s1}, R_{s3}]$, loop type 1 with $R_s \in [R_{s1}, R_{s2}]$, loop type 3 with $R_s \in [R_{s2}, R_{s4}]$.

The tangency condition with $d = 0$ can be studied the same way and Figure 5.41 illustrates some loop configurations.

Plane/Torus (P, To)**:** Loop classification is meaningful only for To where three categories of loop exist and are reached through tangent configurations between P and To. Type 2 loops are produced under the conditions:

$$(d + R\sin\beta) \leqslant r, \tag{5.29}$$
$$\beta \leqslant \alpha, \tag{5.30}$$

where β defines the angle between the axis A_t of To and the normal of P, d is the minimal distance between the center O_t and P. The quantity α is defined by: $\arcsin\left(\frac{r}{R}\right)$.

Type 3 loops are governed by the following conditions:

$$(R - r)\sin\beta > d, \tag{5.31}$$
$$(\pi - \alpha) > \beta \geqslant \alpha, \tag{5.32}$$

of relative position of P and To. Finally, type 1 loops are characterized by the condition:

$$(d + R\sin\beta) \geqslant r, \tag{5.33}$$
$$\beta < \alpha, \tag{5.34}$$

where P cuts To rather orthogonally to A_t and the condition:

$$d \left(R - \frac{r}{\sin\beta}\right) \geqslant 0, \tag{5.35}$$

$$\beta > \alpha, \tag{5.36}$$

when P cuts To rather parallel to A_t.

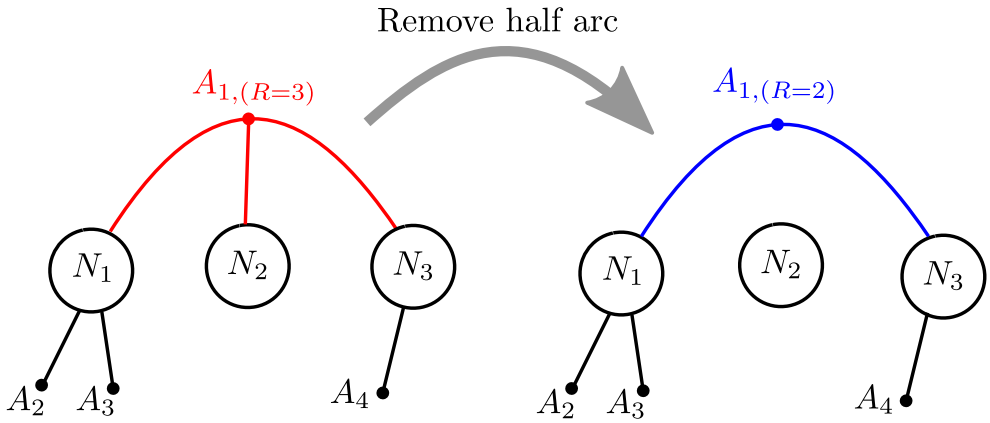


Figure 5.42: Intersections configuration for (P, To): (a) a configuration producing a type 1 loop, (b) a configuration producing a type 2 loop, (c) a configuration producing a type 3 loop, (d) location parameters of plane and torus.

Torus/Torus (To_1, To_2)**:** This configuration is the most complex one since each entity has three categories of loops that must be classified and identified. The existence of loops is also getting more complex since it has to be studied through each category of loops to evaluate whether or not the three categories of loops can be obtained from this initial category. Appendix B gives more details about this configuration and the way it is processed.

Due to the fair complexity of the analysis of interactions between surfaces, the reduction of arbitrary configurations to a subset that is sufficient to cover the shapes of mechanical components is of great interest for the most complex interactions like those involving To and Cy, Co and To. This simplification reduces the development complexity and simultaneously it increases the efficiency of the code to process loops.

5.7.2 Deriving the status of face boundary loops from composite maximal edges

The previous subsection has focused on interactions reduced to loops as they can appear in the hypergraphs since they are mandatory to avoid loosing symmetry properties when vertices break down loops as well as local symmetry properties of the object analyzed. Now, the purpose is to focus on configurations where the loops are no longer formed of an edge loop but they contain a set of maximal edges bounded by vertices.

In order to classify the loops, this configuration with composite maximal edges must benefit properties to take advantage of the loop classification so that the efficiency of the symmetry analysis of the object can be improved. As seen in the beginning of this section, the purpose of this analysis is to assign a status to loops so that they are either external or internal for a cylinder or a cone or of type 1, 2 or 3 for a torus. These surfaces share a common property of axisymmetry that can be used to evaluate the status of a loop. Each of these surfaces is designated as S_{ref}. The sphere is not addressed since there is no loop classification for it.

Each edge forming the loop is an intersection curve between S_{ref} and one of the reference surfaces. This curve can be either open, which essentially applies to cylinders and cones, or closed: the most common configuration. Then, the purpose of the loop analysis is to determine whether a loop is circling around the axis of S_{ref} or not to determine the loop status as (see Figure 5.43):

- External when the loop circles around S_{ref} and S_{ref} is a cylinder or a cone. Otherwise, the loop is internal over these surfaces;
- Type 1 loop when it circles around S_{ref} and S_{ref} is a torus. Otherwise, the loop can be of type 2 or 3 over To. To determine whether the loop is of type 2 or 3 can be achieved with a similar approach since a type 2 loop is somehow circling around the average radius of To. Using this property, the proposed approach can be applied in cylindrical coordinates where a type 2 loop can be regarded as circling around the average circle of To. Otherwise, the loop will be a type 3 one.

Then, determining the status of a loop Γ_i lying on S_{ref} and containing n maximal edges E_j, $j \in \{1, \ldots, n\}$, sequentially connected to each other is performed as follows. Edges of Γ_i are oriented and their orientation is characterized by v_j and $v_{(j+1) \mod n}$, the initial and final vertices of E_j, respectively.

Now, considering the 2D reference frame R_L defined either by a plane orthogonal to the symmetry axis of Cy, Co or To or containing the axis of To and the origin O_L of this frame coinciding with the axis of Cy, Co or To or lying on the average circle of

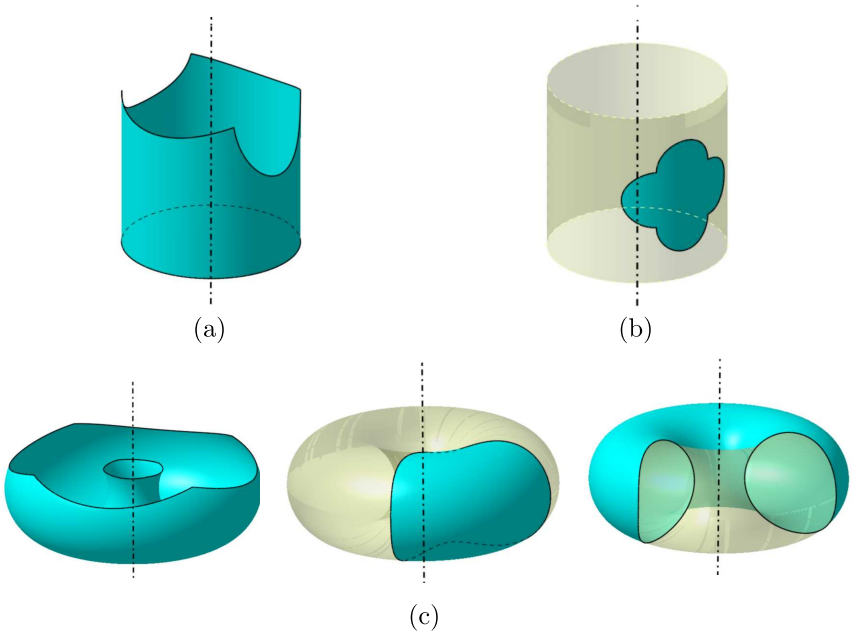


Figure 5.43: Examples of composite loops and their status determined using axisymmetry properties of S_{ref}. (a)(b)axisymmetry used on cylinders and cones, (c) symmetries used on tori.

To. The reference axes $\vec{x}_L$, $\vec{y}_L$ of this frame share the origin O_L and can be arbitrarily oriented. The choice is the orthogonal projection of v_1 lies on $\vec{x}_L$ (see Figure 5.44).

Any vertex v_j of Γ_i can be projected orthogonally onto the plane of R_L and then, with a central projection, onto a unit circle C_L centered on O_L to produce its image $\hat{v}_j$ (see Figure 5.44). Based on this transformation, the relative location of $\hat{v}_j$ and $\hat{v}_{j+1 \mod n}$ on C_L misses the orientation of E_j. Applying the same transformation to any point of the curve(s) forming E_j maps E_j into a continuous arc of circle, possibly containing cusps, on C_L.

As mentioned earlier, the curve defining E_j can belong to either an open curve or a loop that can be either external or internal or of type 1, 2 or 3 for S_{ref}. Independently on this status, the orientation of E_j can be assigned to its image $\hat{E}_j$ using the integral angle between $\hat{v}_j$ and $\hat{v}_{j+1 \mod n}$. This angle is defined as:

$$\hat{\theta}_j = \int_{\hat{v}_j}^{\hat{v}_{j+1}} d\theta = \theta_{\hat{v}_{j+1}} - \theta_{\hat{v}_j}, \tag{5.37}$$

and it defines which sector of C_L represents $\hat{E}_j$ and sets the corresponding orientation of $\hat{E}_j$ (see Figure 5.45a). Rather than computing this integral to take into account the possible cusps of the projection of E_j, the loop status analysis described in the previous section can be exploited with further parameters to define the arc containing

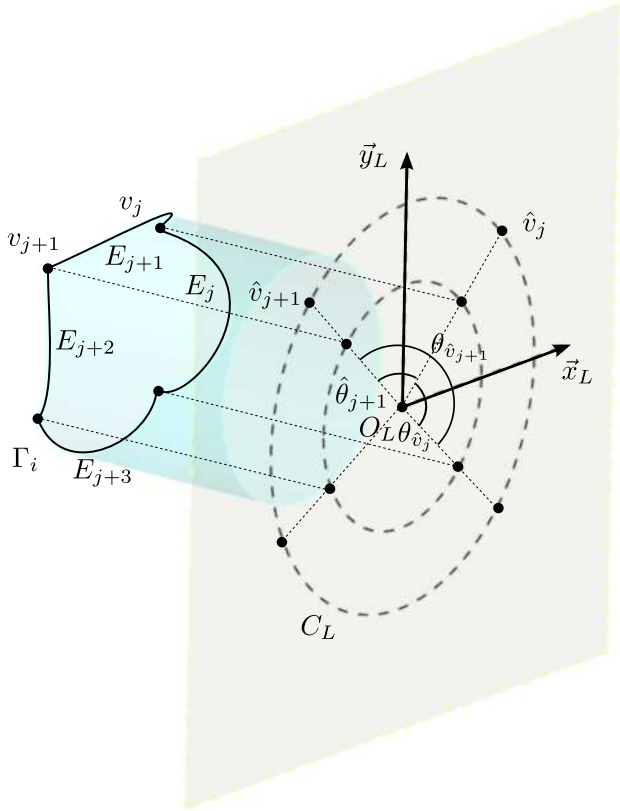


Figure 5.44: Setting up the reference frame to analyze Γ_i and defining the image $\hat{v}_j$ of each vertex v_j.

$\hat{E}_j$. If E_j is:

- An open curve: there exist angular values $\hat{\theta}_j$ that cannot be reached by this curve over C_L (see Figure 5.45b). At least one of these values, $\hat{\theta}_j^O$, can be obtained from the loop status analysis of the previous section. Then, the correct value for $\hat{\theta}_j$ is the arc not containing $\hat{\theta}_j^O$;

- An external loop or a loop of type 1 on To: the analysis of the interactions between two reference surfaces shows that the production of an intersection as external loop or a loop of type 1 on S_{ref} is a monotonic curve with respect to its angular parameterization. Consequently, its image $\hat{E}_j$ does not contain cusps and any point interior to E_j has an image, $\hat{\theta}_j^E$, on $\hat{E}_j$ inside $[\theta_{\hat{v}_j}, \theta_{\hat{v}_{j+1}}]$ on C_L. Then, the correct value for $\hat{\theta}_j$ is the arc containing $\hat{\theta}_j^E$;

- A loop of type 2 on To: here also, the analysis of the interactions between To and

another reference surface shows that the production of an intersection as a type 2 loop for To is a monotonic curve with respect to its angular parameterization. Consequently, its image $\hat{E}_j$ does not contain cusps and any point interior to E_j has an image, $\hat{\theta}_j^{T_2}$, on $\hat{E}_j$ inside $[\theta_{\hat{v}_j}, \theta_{\hat{v}_{j+1}}]$ on C_L. Then, the correct value for $\hat{\theta}_j$ is the arc containing $\hat{\theta}_j^{T_2}$;

- An internal loop or a loop of type 3 on To: in this case, the loop classification provides information about the boundary angles limiting the projection of the intersection curve on R_L. Let these boundary angles projected onto R_L be $\hat{\theta}_{jMin}^{T_3}$ and $\hat{\theta}_{jMAX}^{T_3}$. These angles define a sector on C_L and the interval $[\theta_{\hat{v}_j}, \theta_{\hat{v}_{j+1}}]$ is necessarily inside the interval $[\hat{\theta}_{jMin}^{T_3}, \hat{\theta}_{jMAX}^{T_3}]$ where the intersection curve lies. This configuration uniquely defines the arc of C_L corresponding to the integral angle.

  It has to be noticed that the value and orientation of the integral angle $\hat{\theta}_j$ is not influenced by the projection of v_j and v_{j+1}, lying on the intersection curve defining E_j, i.e. if v_j and v_{j+1} have different locations on the intersection curve but their projection on R_L stay identical, the value and orientation of $\hat{\theta}_j$ is unchanged. The reasoning process described above covers all the possible edge configurations bounded by v_j and v_{j+1} (see Figure 5.46).

Having defined the orientation of the image $\hat{E}_j$ of E_j on C_L, the composite loop Γ_i is cycling around O_L if:

$$\sum_{j=1}^{j=n} \hat{\theta}_j = \sum_j \theta_{\hat{v}_{j+1}} - \theta_{\hat{v}_j} = 2\pi \tag{5.38}$$

hence the corresponding status of the loop: external, type 1 or type 2. Otherwise, Eq. 5.38 equals 0 and the loop status is internal or of type 3.

The loop analysis performed in the current section shows that the loop status can be obtained from the parameters of the reference surfaces. This operation can be fast, most of the time, using the surface parameters and locations. The configurations requiring numerical processing can be simplified and restricted to fairly uncommon ones.

The loop status is obviously an operation that is linear with respect to the number of maximal edges, which does not alter the complexity of the divide phase of the analysis process.

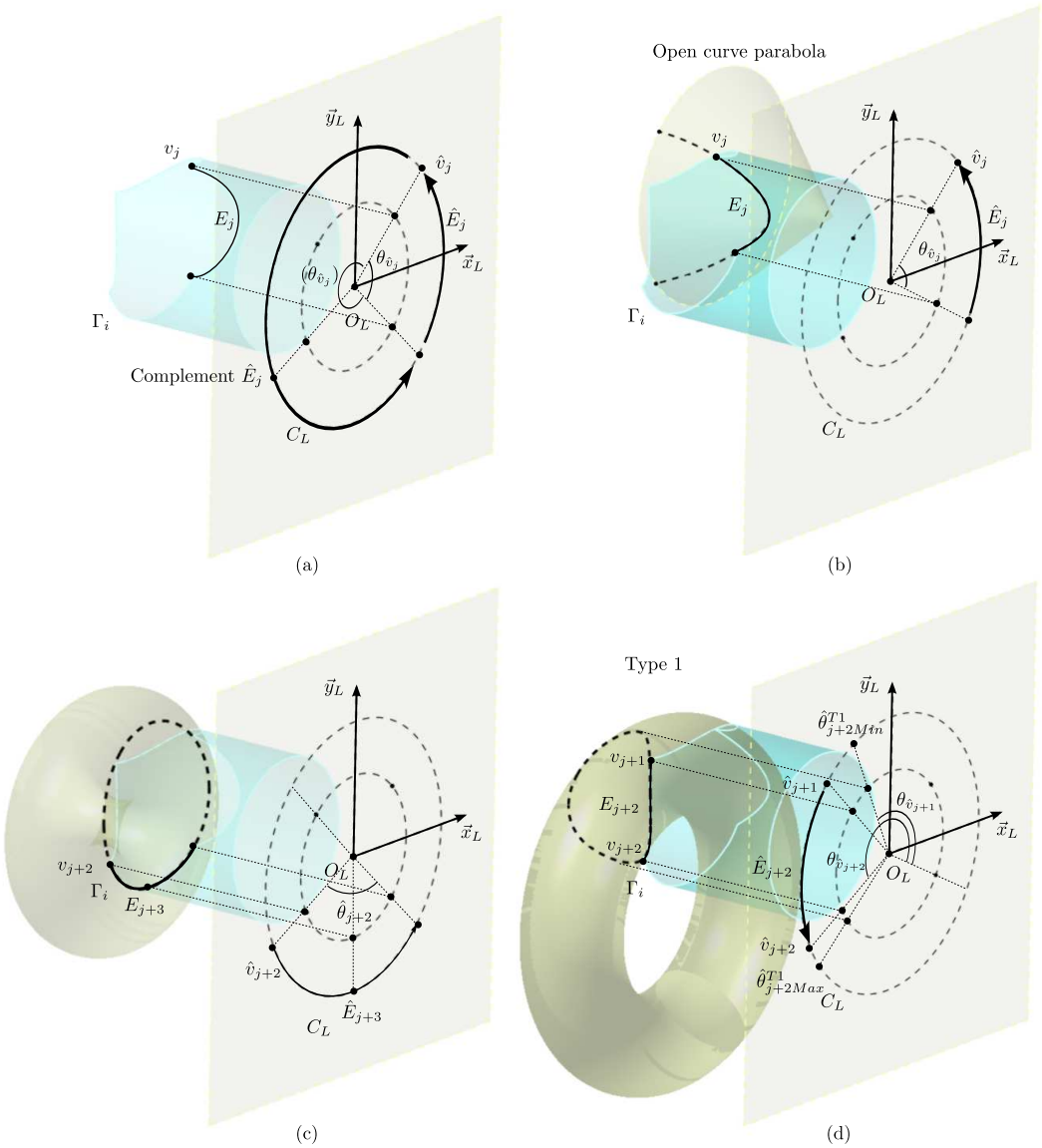


Figure 5.45: Selecting the right integral angle for the image $\hat{E}_j$ of E_j. (a) highlighting the two possible arcs, (b) defining the integral angle when E_j is located on an open curve, (c) defining the integral angle when E_j is located on an external loop or a loop of type 1 or 2, (d) angle boundaries derived from the loop analysis to define the integral angle of E_j.

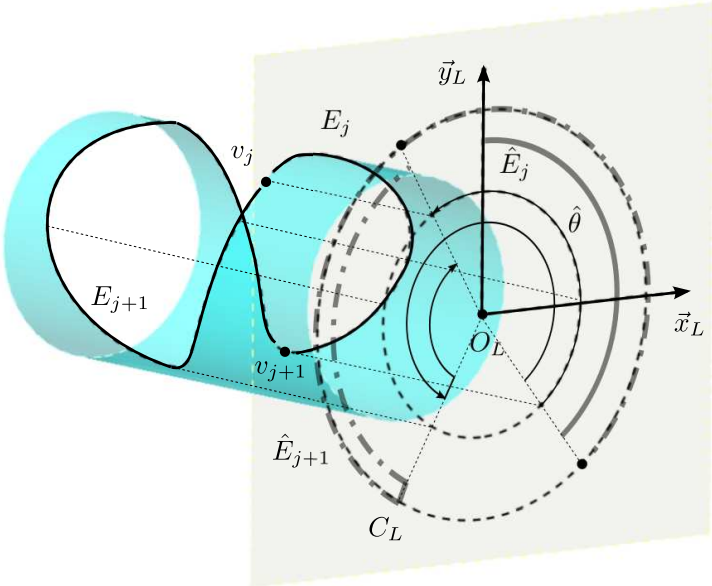


Figure 5.46: An illustration of the invariance of $\hat{E}_j$ of E_j under the possible locations of v_j and v_{j+1} when their projection on R_L is unchanged.

5.8 Conclusion

Collecting CSPs is a major step of the symmetry plane detection algorithm. The analysis of maximal edges, as available in the hypergraphs and resulting from the interaction between adjacent surfaces, help generating all the O-CSPs of the object analyzed. Though the configurations producing O-CSP are rather combinatorial with respect to categories of reference surfaces, it appears critical to take into account the intrinsic symmetry properties of each reference surface when it is embedded in $I\!R^3$ as well as its geometric properties. This explains the necessary combinatorial approach rather than simply relying on a parameterized representation of surfaces as they are available in common B-Rep modelers.

At the same time, BS-CSPs can be collected too, since they are also attached to the maximal edges of the object boundary. Here again, some combinatorial configurations must be addressed to take into account the symmetry and geometric properties of some reference surfaces. Then, traversing the vertices describing the object boundary as part of the hypergraph $G10$, the BL-CSPs can be generated. There, combinatorial configurations are less prominent but there still exists some as an effect of each reference surface properties.

The LS-CSPs only appear when a surface loop number is greater than one. Here,

some interactions between the symmetry properties of each loop have been studied. The effective generation of LS-CSPs will be addressed through the CSP propagation process to characterize the extent of the symmetry properties of the object analyzed. CSAs are special CSPs which provide an infinite number of CSPs. Each type CSP has its own generation criterion. The final CSP set contains all the possible reflective symmetry properties of the object whether they are finite or not. Global symmetry planes, if they exist are in this set. This step of CSP generation can be seen as the initialization of the conquer step of the algorithm analyzing the symmetry of an object. The next step is to filter the CSPs and highlight the global symmetry properties of the object. It stands for the conquer phase of the divide-and-conquer algorithm proposed and it relies on propagation processes. They are introduced in next chapter.

Chapter 6

The propagation of symmetry properties as formal process to the symmetry analysis of an object

The set of CSPs generated during the divide phase uses the whole set of maximal edges and vertices of M_{MAX} and the CSPs have been categorized to structure their areas of validity. These CSPs are valid for a minimal set of entities of ∂M_{MAX}. The purpose of this chapter concentrates on the conquer phase of the symmetry analysis process. The corresponding propagation process is described and aims at extending the area of validity of a CSP. This propagation process subdivides into several complementary stages: a propagation process looking for CSPs coinciding with a reference one, a propagation process devoted to CSAs interacting with CSPs, a propagation process focusing on a new category of CSPs (SS-CSPs) and a propagation process processing LS-CSPs when they are not coinciding with other categories of CSPs. As a result, the symmetry analysis process provide answers to the identification of global symmetries in M_{MAX} as well as local symmetries and their area extent. This conquer phase always ends since it is bounded by the finite number of edges, vertices and faces defining M_{MAX}.

6.1 Introduction

The divide phase of the symmetry analysis has produced various categories of CSPs and CSAs attached to different entities of ∂M_{MAX} forming initial point sets where reflective symmetry properties are valid at these entities and for all the faces involved in the definition of the corresponding CSP or CSA. Namely, they list as:

- O-CSP: symmetry planes that are orthogonal at some point of the intersection curve between two adjacent maximal faces F_i and F_k. The O-CSPs are attached to each maximal edge E_j of ∂M_{MAX}. The point of E_j, where the O-CSP lies, is strictly interior to E_j since vertices, if any, bounding E_j indicate discontinuities whereas the neighborhood of an O-CSP is continuous. There can be up to two

O-CSPs attached to the same edge E_j if E_j is an loop edge and there is only one O-CSP if E_j is bounded by one or two vertices;

- LB-CSP: loop bisector symmetry planes containing a vertex V_j of ∂M_{MAX} lying at the intersection of at least three maximal surfaces: F, the reference face where the loop is analyzed, F_i and F_k the faces adjacent to F and the two edges sharing V_j. Obviously, there is no LB-CSP for a loop edge. The maximum number of LB-CSPs at V_j is equal to the number of faces adjacent at V_j;

- BS-CSP: symmetry planes containing a planar intersection curve between two adjacent maximal faces F_i and F_k. There is only one BS-CSP attached to an edge E_j and there can be as many BS-CSP as there are maximal edges in ∂M_{MAX}. An edge E_j can carry simultaneously O-CSPs and a BS-CSP;

- LS-CSP: symmetry planes devoted to faces bounded by multiple loops. They are attached to the reference face F where the loops are lying since an LS-CSP is valid for all the loops bounding F. The number of LS-CSPs attached to F cannot exceed the number of CSPs attached to the external loop(s) Γ_{ext} (or $\Gamma_{1,2}$) bounding F where external loop(s) is applicable. If Γ_{ext} (or $\Gamma_{1,2}$) contains n edges, then the maximum number of LS-CSPs is $2n$ with no more than n O-CSPs and n BS-CSPs. With F represented as a sphere, the amount of LS-CSPs can become quadratic with respect to the number of loops;

- CSA: symmetry axes that characterize the existence of an infinite number of O-CSPs at an edge E_j adjacent to faces F_i and F_k. There are two specific configurations of attachment of CSAs to entities of ∂M_{MAX}: spheres and tori. Indeed, if the object analyzed reduces to a torus, ∂M_{MAX} has no edge: the symmetry axis must be attached to the face defining the torus. Now, if the object is a sphere, ∂M_{MAX} has also no edge and the sphere has an infinite number of CSAs. This infinite number of CSAs has to be attached to the face representing the sphere.

Having defined all the CSPs attached to ∂M_{MAX}, the purpose of the propagation process is to effectively perform the analysis of M_{MAX} to provide answers to one or more questions set as objectives in Chapter 2. The propagation process forming the conquer phase of the algorithm is not just one process but it is a set of propagation mechanisms to take advantage of properties derived from the various categories of CSPs and CSAs. As a preliminary stage, the analysis of the extent of validity of CSPs/CSAs stands for an 'initialization' of the propagation process.

6.2 The symmetric area surrounding a CSP or a CSA

First of all, it seems necessary to define precisely the extent of the symmetric area valid for a current CSP or CSA. For the sake of simplicity of presentation, all the previous entities are collectively designated as CSPs unless some distinction is mandatory. Also, it has to be recalled that the generation of maximal faces and edges produces a decomposition of ∂M_{MAX} such that all faces are bounded by one loop at least, i.e. there is no isolated vertex or edge. Each loop possibly reduces to a loop edge having one or no vertex at all.

Now, starting with the simplest configuration where only two surfaces F_1, F_2, involved in the generation of CSPs define ∂M_{MAX}. CSA, O-CSP, BS-CSP are the only possibilities and they must be associated with a closed intersection curve I_c forming a loop edge and a resulting surface that is closed. It has to be noticed that any couple of reference surfaces may not satisfy all these constraints. Then, the divide phase assigns the possible CSA, O-CSPs or BS-CSP. Because these planes or axis are defined from the intrinsic parameters of F_1 and F_2, they are valid for I_c as well as for the entire area of F_1 and F_2. Hence, these planes and axis are valid for M_{MAX} entirely.

Then, generalizing to surfaces with a symmetric boundary, the CSP is the real symmetry plane of F_1 and F_2 since their boundaries are effectively taken into account. When the boundary of F_1 or F_2 is reduced to an edge bounded by two vertices, this boundary is open and the validity of the symmetry property can be analyzed as follows (see Figure 6.1a) after a decomposition of the interaction between F_1 and F_2 into entities of different dimensions. This decomposition conforms to the topological one into faces, edges and vertices. Then, the validity of symmetry properties is analyzed for each category of CSP and focuses on edges as infinite point sets.

Starting from their intersection curve I_c (edge E_1) and considering an O-CSP attached to I_c, the symmetry of I_c with respect to the O-CSP is valid from the point V_{start} until its extent reaches its extreme points V_1, V_2. Hence, checking their symmetry is the first level to validate this property over a bounded domain (see Chapter 5) when creating the CSP.

Processing an LB-CSP is rather similar (see Figure 6.1b); it is attached to three faces F_1, F_2, F_3 and two intersection curves E_1, E_2 because it is assigned at their intersection point V_{start} and their extreme vertices V_1, V_2 must be symmetric with respect to the LB-CSP. A BS-CSP coincides with the intersection curve I_c (edge E_1) and the symmetry property is valid for all of its points (see Figure 6.1c).

The case of an LS-CSP is slightly different. An LS-CSP acts on a rather global domain: it is applied to two loops Γ_1, Γ_2 lying in a face F_1 (see Figure 6.1e, f). Each loop can be defined either by one loop edge or be composite. The symmetry

property of the LS-CSP is valid for every point of Γ_1, Γ_2. If Γ_i are loop edges, it means that the symmetry property is addressed globally through the relative position and intrinsic parameters of F_2 and F_3, the faces defining Γ_1, Γ_2 and adjacent to F_1 (see Figure 6.1f). Otherwise, if Γ_i are composite loops edges is coming from closest vertices, the symmetry property can be decomposed through two sets of entities: vertices of Γ_i and then, edges. Checking the symmetry of these sets of vertices can be conducted first (see Figure 6.1e) and then, using the surface parameters and locations of the faces adjacent to F_1 and defining the edges of Γ_i, the symmetry of the sets of edges can be analyzed.

Finally, a CSA appears as a particular configuration of O-CSP where I_c is represented by a loop edge with no vertex. Hence, the symmetry property is valid for the whole loop edge E_1.

All the CSP configurations involving point sets corresponding to edges have now been analyzed. Now the purpose is to focus on faces: another type of infinite point sets. Indeed, the generation of CSPs incorporate the spatial location and parameters of the reference surfaces adjacent to each edge involved in the definition of a CSP. The symmetric area around a CSP is not only along its associated edges, the surrounding surfaces of these edges is symmetric, too. Consequently, in a small area around each point on the symmetric curve(s), the symmetry property apply to the adjacent faces: F_1, F_2 for an O-CSP, a BS-CSP, a CSA; F_1, F_2, F_3 for a LB-CSP; F_i the faces adjacent to F_1 in case of LS-CSP. Because all the points of these adjacent faces benefit of the same geometric properties as those in the neighborhood of their common edges, the symmetry property extends to these entire surfaces, whether they are bounded or not.

Then, each face F_i of M_{MAX} being bounded by one loop at least, it is mandatory to insert repetitively all the CSPs attached to the vertices, edges and to F_i that form the boundary of F_i. Adding new vertices and edges to a first edge, means that new faces adjacent to F_i are also added until all the loops of the surface are covered. Then, the area of F_i is closed and all the constraints are set to analyze the symmetry over F_i. In this case, the validity of the current CSP for the whole face can be evaluated through the relative positions of the faces adjacent to F_i that model each edge of the loops bounding F_i. If these are symmetrically set with respect to the CSP, then F_i is effectively symmetric with respect to this CSP. However, if any of these surfaces is not symmetrically set with respect to the CSP, the propagation stops. The symmetry property is lost.

It has also to be pointed out that if the CSP is valid for F_i, the symmetry property extends outside F_i to its adjacent faces, forming an open domain outside F_i where the symmetry holds.

Figure 6.1a is the illustration of the symmetry area around an O-CSP. V_{start} is the intersection point between E_1 and the O-CSP. The gray area around E_1 illustrates the extension of the symmetry property along F_1 and F_2. Then, this symmetry needs

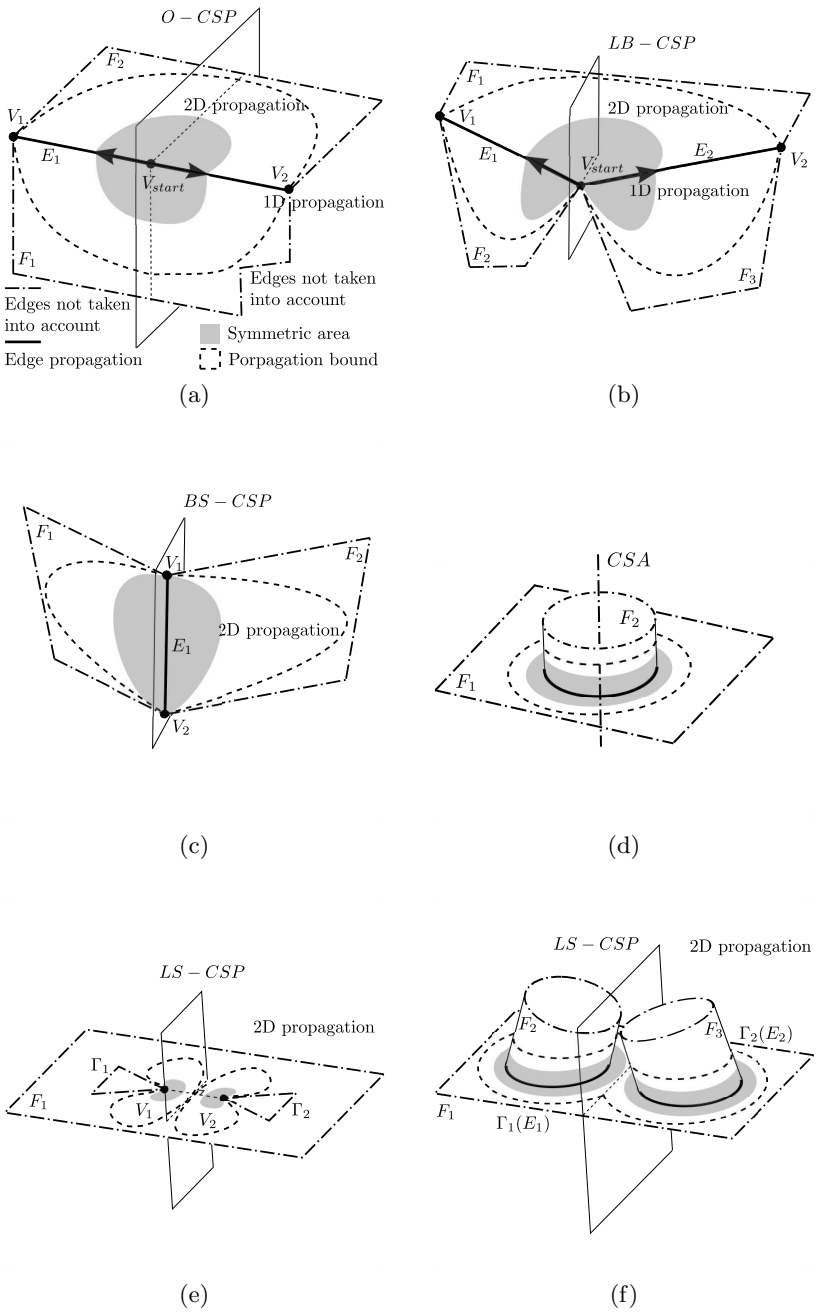


Figure 6.1: Symmetry area of each CSP category: (a) symmetry area of an O-CSP; (b) symmetry area of an LB-CSP; (c) are symmetry area of a BS-CSP; (e) and (f) are symmetry areas of an LS-CSP.

to be analyzed with respect to the complementary boundary entities of F_1 and F_2. In F_2, all boundary curves are symmetric with respect to the O-CSP. This plane is valid for F_2. But in F_1, its boundary loop is asymmetric, hence the symmetry property cannot be expanded to the finite area defined by F_2.

Figure 6.1b is the illustration about the symmetry area of an LB-CSP. LB-CSP is generated from faces F_1, F_2, F_3. The gray area shows the extent of the symmetry around V_{start} and the edges E_1 and E_2, which are symmetric with respect to the LB-CSP.

Figure 6.1c shows the symmetric area around a BS-CSP. In this chapter's introduction, it has been recalled that a BS-CSP is related to an edge E_1 , intersection of F_1 and F_2. The gray area indicates the propagation of the symmetry property, starting from E_1 and progressing over F_1 and F_2.

In Figure 6.1d, the symmetry area represented originates from a CSA. This symmetry axis can be seen as attached to the intersection curve E_1 between F_1 and F_2. E_1 is necessarily a loop edge without vertex corresponding to a circle. The symmetry area, in gray, expands on both sides of E_1 to propagate the symmetry axis from E_1.

Figure 6.1e and f depicts the symmetry areas attached to an LS-CSP. Depending on the type of the internal loops, they can be either composite or loop edges. Figure 6.1e illustrates the symmetry area restricted to vertices when they are considered as an independent set of entities, distinct from the edges. The symmetry area propagates from two symmetric vertices of the two loops. In Figure 6.1f, the loops are loop edges, the LS-CSP originates from F_2 and F_3 So, $\Gamma_1(E_1)$ and $\Gamma_2(E_2)$ are symmetric with respect to the LS-CSP and the symmetry area expands from these two edges. Considering the bounded area of F_1, the symmetry of the LS-CSP is also valid globally for F_1.

When processing an LS-CSP as well as during the second level of symmetry propagation (see Section 6.7), it is necessary to check the symmetry of vertex pairs, edges pairs and face pairs with respect to this plane that is not attached a vertex or an edge.

Evaluating the symmetry of vertices, edges and faces with respect to a symmetry plane can be addressed through a bottom-up approach. In this case, the Figure 6.2 is an illustration with all the successive steps to analyze the symmetry of edges E_1 and E_2 and their adjacent faces, when E_1 and E_2 are bounded by two vertices. The first step processes the extreme points two the edges E_1 and E_2. Choosing arbitrary V_{11} as starting vertex and the other vertex as V_{22}, if they are asymmetric, then changing it to V_{21}, this time, the vertices must be symmetric otherwise the configuration is not symmetric (see Figure 6.2a and b). Thus, assuming that the vertex pairs are symmetric, Figure 6.2c illustrates the beginning of second step. With the help of hypergraph $G21$, the surfaces adjacent to E_1 and E_2 are picked up. Then, the intrinsic parameters and locations of F_{11} and F_{21}, both cylinders (Cy_1 and Cy_2), are evaluated for symme-

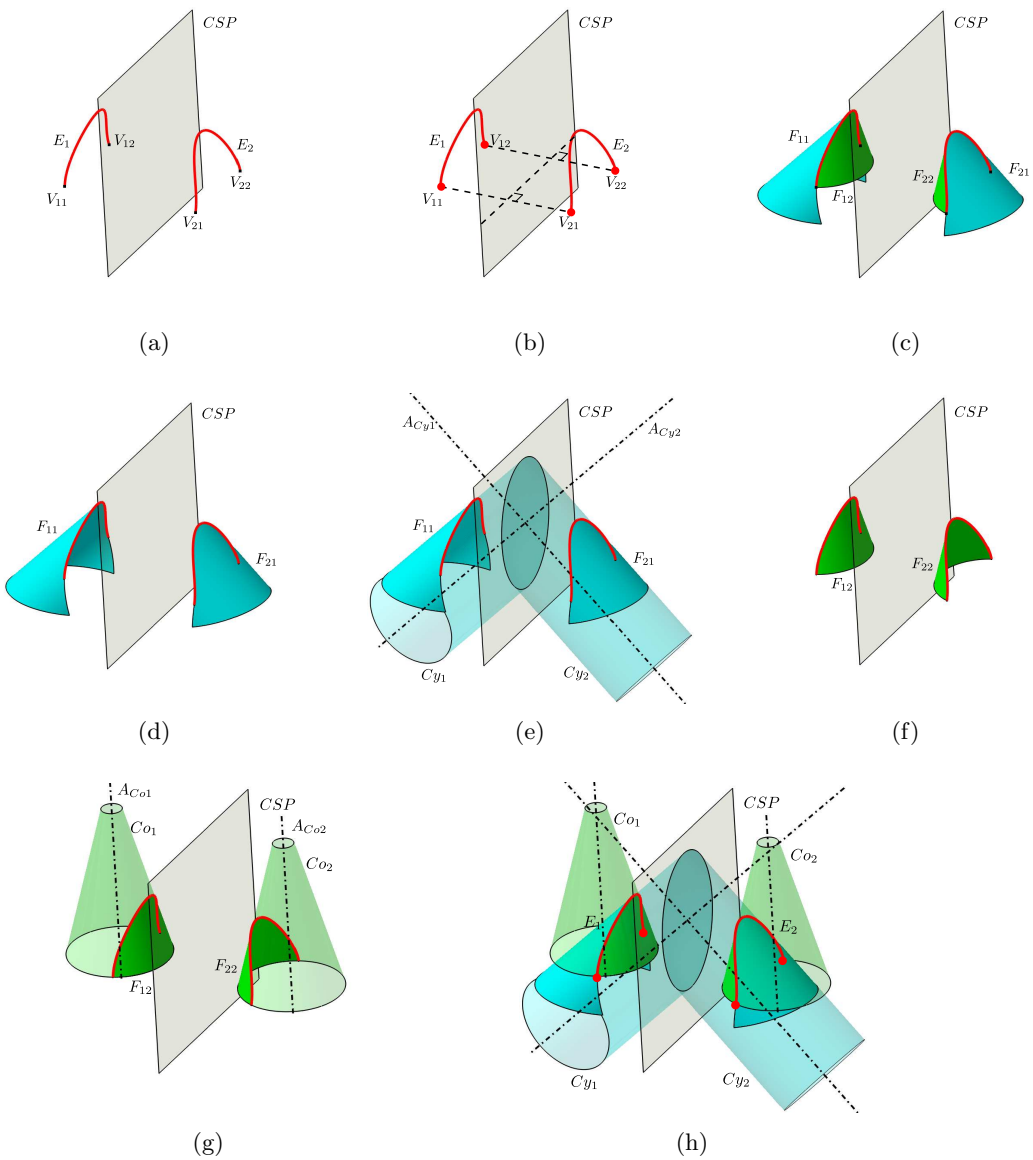


Figure 6.2: The symmetry analysis process for two edges E_1 and E_2: (a) current edges and CSP; (b) symmetry analysis of extreme points; (c) adjacent surfaces at each edge; (d) selection of one surface pairs; (e) the symmetrical cylinders; (f) the second pair of surfaces; (g) the symmetrical cones; (h) all the entities leading the symmetry of E_1 and E_2.

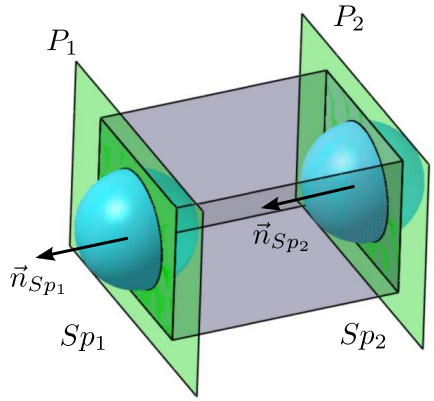


Figure 6.3: An illustration of the need to distinguish concave/convex areas during the symmetry analysis of an object.

try. Assuming the symmetry, the homologous cylinders are shown in Figure 6.2e and they are symmetric. Then, selecting the other pair of surfaces F_{12} and F_{22}, they are cones (Co_1 and Co_2), and their are symmetric with respect to the symmetry plane, as shown in Figure 6.2g. Finally, this plane or CSP is valid for E_1 and E_2 as well as for its neighboring surfaces.

If E_1 and E_2 are loop edges, the above symmetry analysis can be reduced to surface comparison when the edges have no vertex. Indeed, the symmetry property, if valid, applies to all the points of Cy_1 and Cy_2, Co_1 and Co_2. The B-Rep model of the object contains information to describe the used area of a surface. The parametric representation of these surfaces and the external loop defined in their associated parametric space characterizes their used area. However, this information does not produce a characterization of the concavity/convexity of the corresponding face. This is mandatory to separate configurations when analyzing the symmetry of an object (see Figure 6.3). To this end, the concept of orientation index (see Section 5.4.6) is used to separate the ambiguous configurations. In addition to the geometric conditions stated previously, couples of faces must have the same orientation index to effectively meet the symmetry property.

In M_{MAX}, each edge has two neighbor surfaces. So, the propagation mechanisms described in this section, combined with the adjacency relations available in the hypergraphs can be used to conquer, i.e. to cover, the entire boundary, ∂M_{MAX}. In this case, the propagation process overlaps the face boundary and when all faces of ∂M_{MAX} are covered without any asymmetry, the CSP is upgraded to the status of *Global Symmetry Plane* (GSP). Similarly, a CSA extending to the whole boundary of M_{MAX} becomes a GSA. A first propagation process is now described in the following section.

6.3 First level propagation and CSP chains

Considering the fact that the object category studied here is 2-manifold B-Reps, this means that GSPs, if there exists any, must cut M_{MAX}. In addition, this intersection forms one closed curve at least over M_{MAX}. If there is more than one loop, they are disconnected from each other. Consequently, either or both following conditions must hold:

a) There exists at least an edge or a face of M_{MAX} cut by a GSP. In case of a face, it means that the GSP cuts the face without cutting its boundary;

b) There exists at least a vertex or an edge of M_{MAX} lying in a GSP.

Now, observing locally the configuration of a GSP, if it satisfies condition a, it is locally similar to an O-CSP if it cuts an edge or it is similar to an LS-CSP if it cuts a face, e.g. LS-CSP orthogonal to the CSA of a cylinder. If a GSP satisfies condition b, it can be locally similar to a LB-CSP if the GSP contains a vertex and if it contains an edge it is locally similar to a BS-CSP.

As a result, the four categories of CSPs can be used to initiate a propagation process producing a GSP. It has to be observed that the four categories of CSPs are necessary and sufficient to cover all the configurations representing the intersection of a GSP with M_{MAX}.

The above conditions don't incorporate the configuration where a GSP interacts with a CSA. Such an interaction reduces to the selection of one symmetry plane from the infinity available with the CSA. Indeed, the plane selected produces a configuration locally similar to an O-CSP since a CSA is attached to an edge like an O-CSP. If M_{MAX} contains only CSAs, they can propagate from each other to form a GSA, showing that M_{MAX} is axisymmetric. Again, this analysis shows that the CSA category is necessary and sufficient to characterize the axisymmetry of M_{MAX} and this category enlarges the diversity of objects that can be analyzed to produce GSPs since the interaction of a CSA with a GSP reduces it to an O-CSP. This means also that a CSA cannot initiate the propagation process to determining the faces intersecting with a GSP.

Detecting the faces of M_{MAX} that reflect the interaction between a GSP and M_{MAX} is the objective of the first propagation process. It means, if there is an asymmetric configuration appearing, the corresponding CSP cannot become a GSP and the propagation process stops.

The starting point of this propagation process is the lists of CSPs: $[\Pi_O]$ for the O-CSPs, $[\Pi_{BS}]$ for the BS-CSPs, $[\Pi_{LB}]$ for the LB-CSPs, generated during the divide phase. At this stage, there is no LS-CSP. Indeed, only LS-CSPs not intersecting the external loop(s) have to be considered. Because all faces of M_{MAX} are bounded, it means that these LS-CSPs cutting the face they are attached to, produce a closed

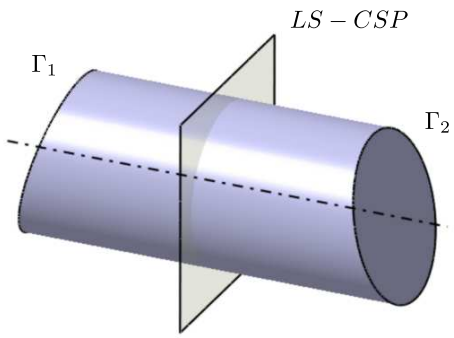


Figure 6.4: LS-CSP cutting a cylindrical face without intersecting its external loops.

curve (see Figure 6.4). Because these LS-CSPs produce directly a closed curve, there is no need of propagation to produce a closed path, the second propagation process can start directly using these CSPs.

This first propagation process aims at reducing the number of CSPs as soon as possible to improve the efficiency of the algorithm. If two CSPs of the above lists coincide or are co-planar, using the symmetry property of the area around a CSP, other edges and neighboring faces can be identified with the help of the adjacency relations expressed by the hypergraphs. So, this propagation process relies on a criterion of coinciding CSPs.

Now, let us study the behavior of the propagation process with respect to the various categories of CSPs. Starting with an O-CSP, the corresponding edge E_1 is self symmetric. Using a neighboring face of E_1, it is possible to select a loop Γ which contains the current O-CSP and to look for another CSP coinciding with the O-CSP. Regarding CSPs attached to entities of a loop, they are among the categories O-CSP, BS-CSP and LB-CSP. A loop having no self intersection, a property can be stated regarding the CSPs:

Property 3 *CSP propagation conditions over a loop Γ: Given a loop Γ without self intersection, Γ contains n edges and lies on face F, and let us consider a first configuration where there exists an O-CSP attached to the edge E_1. If n is even, the O-CSP must coincide with one and only one other O-CSP to able to propagate the symmetry property across F. If n is odd, the O-CSP must coincide with one and only one LB-CSP.*

Let us consider also a second configuration where there exists an LB-CSP attached to the vertex V_1 of Γ. If n is even, the LB-CSP must coincide with one and only one O-CSP in Γ to propagate the symmetry property across F. If n is odd, the LB-CSP must coincide with one and only one other LB-CSP.

There cannot be any symmetry plane of type BS-CSP in Γ.

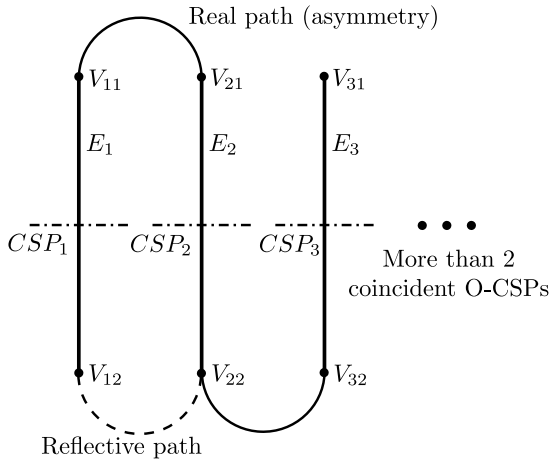


Figure 6.5: No more than one O-CSP in a loop coincides with an initial O-CSP.

As shown in Figure 6.5, E_1, E_2 and E_3 belong to Γ, which has no self intersection. Let us consider an O-CSP attached to E_1 as reference CSP. E_1 can be used to divide the $(n-1)$ other edges and n vertices of Γ into two subsets of similar entities $[E_i^1]$, $[E_i^2]$; $[V_i^1]$, $[V_i^2]$. The cardinality of each set $[E_i^j]$, $[V_i^j]$ is identical.

If n is even, $\text{Card}[E_i^j] = \frac{n-2}{2}$ and the edge left, E_k, is the farthest one from E_1 by adjacency; $\text{Card}[V_i^j] = \frac{n}{2}$. These sets represent the left hand side and right hand side of Γ with respect to E_1 (see Figure 6.5). Then, assuming that there exists another O-CSP in $[E_i^1]$, Π_{O1}, coinciding with the reference CSP, if F is symmetric with respect to this reference CSP then, there exists another O-CSP in $[E_i^2]$, Π_{O2}, that coincides with the reference CSP and has the same adjacency position as Π_{O1}. If so, the edges corresponding with Π_{O1} and Π_{O2} must coincide, which shows that Γ self intersects along these edges. This contradicts the initial hypothesis along which Γ has no self intersection, hence there no O-CSP in $[E_i^1]$ coinciding with the reference one. The propagation process can take place only if E_k has an O-CSP that coincides with the reference one.

Now, if n is odd, $\text{Card}[E_i^j] = \frac{n}{2}$; $\text{Card}[V_i^j] = \frac{n-1}{2}$ and the vertex left, V_k, is the farthest one from V_1 by adjacency. Then, assuming that there exists an LB-CSP in $[V_i^1]$, Π_{O1}, coinciding with the reference CSP, if F is symmetric with respect to this reference CSP then, there exists another LB-CSP in $[V_i^2]$, Π_{O2}, that coincides with the reference CSP and has the same adjacency position as Π_{O1}. If so, the vertices corresponding with Π_{O1} and Π_{O2} must coincide, which shows that Γ self intersects at these vertices. This contradicts the initial hypothesis along which Γ has no self

intersection, hence there no LB-CSP in $[V_i^1]$ coinciding with the reference CSP. The propagation process can take place only if V_k has an LB-CSP that coincides with the reference CSP.

These two analyses show that no LB-CSP can coincide with the reference CSP when n is even and that no O-CSP can coincide with the reference CSP when n is odd.

Following a similar reasoning process, when the reference CSP is an LB-CSP and the reference entity is a vertex V_1 leads to the results stated for the second configuration.

Now, whatever the cardinality of Γ, let us assume that $[E_i^1]$ contains an edge, E_k, having a BS-CSP coinciding with a reference CSP attached to an edge or a vertex, then $[E_i^2]$ must contain an edge, E_l, having a BS-CSP also coinciding with the reference CSP and located at the same adjacency position as in $[E_i^1]$. If so, it means that E_k and E_l coincide, hence F self intersects, which contradicts also the hypothesis. If n is odd so that there is an edge left outside $[E_i^1]$ and $[E_i^2]$ if the reference CSP is attached to a vertex, this edge cannot have a BS-CSP coinciding with the reference one since these extreme vertices would lie in the reference CSP, which contradicts the symmetry property of Γ that is needed to propagate from F to adjacent faces.

This demonstrates the above property of the propagation process of CSPs over a loop.

Property 4 *CSP propagation conditions through a vertex V_i: Given a vertex V_i whose neighboring faces define an LB-CSP forming three adjacent faces where F is the reference face sharing the two reference edges E_1, E_2 with its adjacent faces. V_i as well as its surrounding faces and edges don't self intersect. V_i has n, $n \geqslant 3$, surrounding faces altogether.*

Let us consider a first configuration where n is even, then the LB-CSP must coincide with one and only one other LB-CSP to be able to propagate the symmetry property across V_i. If n is odd, the LB-CSP must coincide with one and only one BS-CSP.

Let us consider also a second configuration where there exists a BS-CSP attached to the vertex V_i with E_1 the corresponding reference edge. If n is even, the BS-CSP must coincide with one and only one other BS-CSP to be able to propagate the symmetry property across V_i. If n is odd, the BS-CSP must coincide with one and only one other LB-CSP.

As shown in Figure 6.6b, E_2 and E_8 are the reference edges defining an LB-CSP at V_1 together with F_1. Let us consider this plane as reference CSP. F_1 can be used to divide the $(n-2)$ other edges and $(n-3)$ faces at V_1 into two subsets of similar entities $[E_i^1]$, $[E_i^2]$; $[F_i^1]$, $[F_i^2]$. The cardinality of each set $[E_i^j]$, $[F_i^j]$ is identical.

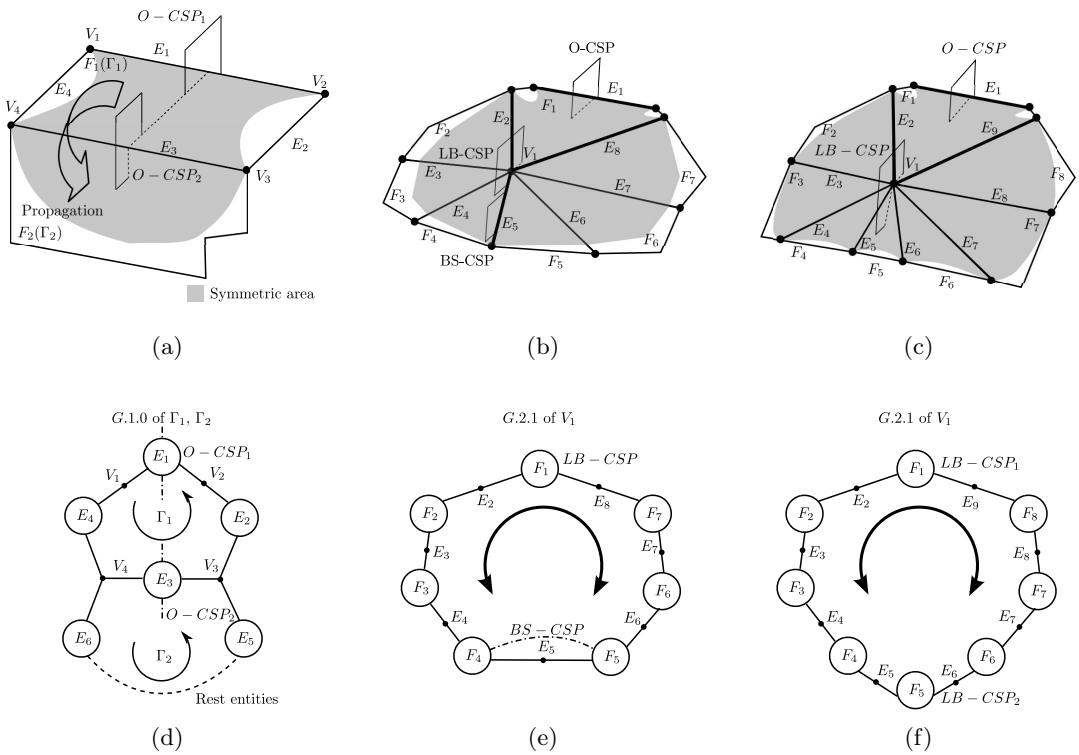


Figure 6.6: The propagation mechanisms for the first level propagation: (a) and (d) propagation from an O-CSP to an O-CSP; (b), (e) are O-CSP to LB-CSP, and then to BS-CSP; (c) and (f) are O-CSP to LB-CSP.

If n is even, $\text{Card}[E_i^j] = \frac{n-2}{2}$; $\text{Card}[F_i^j] = \frac{n-3}{2}$ and the face left, F_k, is the farthest one from F_1 by adjacency. These sets represent the left hand side and right hand side of V_i with respect to F_1 (see Figure 6.6b). Then, assuming that there exists another LB-CSP in $[F_i^1]$, Π_{LB1}, coinciding with the reference CSP, if F_1 is symmetric with respect to this reference CSP then, there exists another LB-CSP in $[F_i^2]$, Π_{LB2}, that coincides with the reference CSP and has the same adjacency position as Π_{LB1}. If so, the edges defining Π_{LB1} and Π_{LB2} must coincide, which shows that these edges including V_i self intersect. This contradicts the initial hypothesis along which the neighborhood of V_i is self intersection free, hence there no LB-CSP in $[F_i^1]$ coinciding with the reference one. The propagation process can take place only if F_k has an LB-CSP that coincides with the reference one.

Now, if n is odd, $\text{Card}[E_i^j] = \frac{n-2}{2}$ and the edge left, E_k, is the farthest one from F_1 by adjacency; $\text{Card}[F_i^j] = \frac{n-3}{2}$. Then, assuming that there exists an BS-CSP in $[E_i^1]$, Π_{BS1}, coinciding with the reference CSP, if F_1 is symmetric with respect to this reference CSP then, there exists another BS-CSP in $[E_i^2]$, Π_{BS2}, that coincides with

the reference CSP and has the same adjacency position as Π_{BS1}. If so, the edges corresponding with Π_{BS1} and Π_{BS2} must coincide, which shows that these edges including V_1 self intersect. This contradicts the initial hypothesis along which the V_1 neighborhood has no self intersection, hence there no BS-CSP in $[E_i^1]$ coinciding with the reference CSP. The propagation process can take place only if E_k has a BS-CSP that coincides with the reference CSP.

Changing the reference CSP into a BS-CSP and starting over the same reasoning process the property reduced to the second configuration.

Figure 6.6e and f show the hypergraph $G21$ reduced to the neighborhood of V_1 and illustrate configurations where n is respectively odd or even around V_1. Consequently, a new edge E_5 can be added to the symmetric elements set in Figure 6.6e and a new face F_5 is added in Figure 6.6f.

The above two properties show that one step of the propagation process can extend the symmetry property to:

- A new edge and its adjacent face if a new O-CSP coincides with the reference CSP;
- two new edges and their adjacent faces that bound the face of the reference CSP when a new LB-CSP coincides with the reference CSP when the propagation is performed over a loop;
- A new face and its two adjacent edges and the two faces adjacent these new face and edges when a new LB-CSP coincides with the reference CSP when the propagation is performed across a vertex;
- A new edge and its two adjacent faces when a new BS-CSP coincides with the reference CSP.

As a result of the above synthesis, propagation rules can be set up as follows:

- If the propagation takes place over a loop, the last CSP incorporated through the propagation process is either an O-CSP or an LB-CSP, then the next one, Π, can be found only in the $[\Pi_O]$ or the $[\Pi_{LB}]$ lists and the next reference entity for the propagation process is:
  - A loop and, more precisely, an edge in this loop if $\Pi \in [\Pi_O]$;
  - A vertex if $\Pi \in [\Pi_{LB}]$;
- If the propagation takes place across a vertex, the last CSP incorporated through the propagation process is either an LB-CSP or a BS-CSP, then the next one, Π, can be found only in the $[\Pi_{LB}]$ or the $[\Pi_{BS}]$ lists and the next reference entity for the propagation process is:

– A loop and, more precisely, an vertex in this loop if $\Pi \in [\Pi_{LB}]$;
– A vertex if $\Pi \in [\Pi_{BS}]$.

Figure 6.6a through c are illustrations of some of the previous configurations.

Now that all the possible configurations of CSP propagation under the coincidence constraint have been studied when this propagation takes place:

- Over a loop;
- Across a vertex;

and the propagation rules have been set up on the CSP category basis as well as from the type of entities of M_{MAX} supporting each new CSP, the concept of CSP chain is introduced when ∂M_{MAX} solely contains faces bounded single loops.

Definition 8 *CSP Chain A CSP chain contains a sequence of coinciding CSPs selected from the three categories: O-CSP, LB-CSP, BS-CSP. The sequence is a chain of items where each of them contains the CSP type and the corresponding entities of ∂M_{MAX} where the symmetry property is extended. A CSP chain can be open or closed when the propagation process stops because there is an asymmetric configuration encountered or when the last CSP found is the CSP having initiated the chain, respectively.*

Because the propagation process scans ∂M_{MAX} on the basis of faces (more precisely the loop bounding each face) or vertices, the CSP Chain extension needs two different geometric functions: checking coincidence of CSPs within a face; checking coincidence of CSPs around a vertex.

Based on the property of CSPs over a loop and across a vertex, a CSP chain cannot contain an element defined by a BS-CSP that is adjacent to element defined by an O-CSP.

When a CSP chain is closed, the corresponding entities of ∂M_{MAX} that define the CSPs of this chain can be used to define a loop over ∂M_{MAX} that intersects with the CSP having initiated this chain.

During this propagation process, each CSP added to a CSP chain is removed from the list corresponding to its category, i.e. either $[\Pi_O]$, $[\Pi_{LB}]$ or $[\Pi_{BS}]$, hence reducing the number CSPs available as initiators of CSP chains. Consequently, the first level propagation process stops when there no CSP left is the lists $[\Pi_O]$, $[\Pi_{LB}]$ or $[\Pi_{BS}]$.

Now, considering the example model of Figure 6.7 that is presently reduced to a simple cube to meet the current constraint where faces are bounded by single loops, $O - CSP_1$ and $LB - CSP_1$ are two example CSPs used to illustrate the propagation

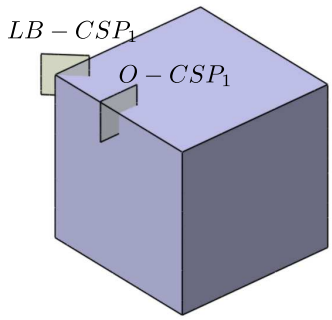


Figure 6.7: An example model with a subset of CSPs chosen as initiators of CSP chains.

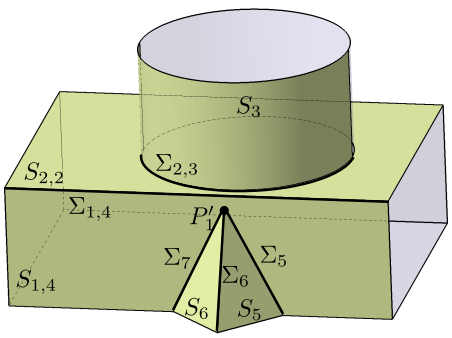


Figure 6.8: 2 closed CSP chains of the example model at Figure 6.7 with their 3D and unfolded representations.

process. Figure 6.8a and b contains the 3D representation of the entities involved in the propagation starting with $O - CSP_1$ and the unfolded representation of the same entities. The **Start** and **End** mark the start and end edges of a propagation process. The dotted lines indicate areas not taken into account. Figure 6.8g and h contains also the 3D representation of the entities in the propagation initiated by $LB - CSP_1$.

In accordance with the above analysis of the propagation process, the complexity of each step, whether it is a propagation over a loop or across a vertex, is able to define a new CSP, if any, in a constant time. As a result, the complexity of the propagation process is linear with respect to the number of faces and vertices of M_{MAX} even though the number of faces and vertices involved in each propagation process is significantly lower than the total amount of faces and vertices defining M_{MAX}.

As stated previously, the current description of the propagation process addresses ∂M_{MAX} under the hypothesis where all the faces are bounded by a unique loop. Let us now remove this condition to process faces bounded by an arbitrary number of loops. The next section focuses on this new configuration.

6.4 Propagation of first level with faces bounded by multiple loops

In a first place, let us observe how the propagation process is influenced when the loops are not classified at all, i.e. there is no use of external / internal status. As a variant to the previous propagation, this new class of input model M_{MAX} can contain symmetry axes but this aspect is not addressed here yet. The main purpose now, is to address the generation of LS-CSPs.

Let us consider that processing faces with multiple loops takes place during a second phase of the propagation process, the first one being the propagation process with faces bounded by a unique loop. $[\Pi_{BS}]$ are not influenced by faces with multiple loops since they are attached to edges only and they cannot cross any face to coincide with an LS-CSP.

Having structured the CSP lists, the first phase of the propagation process can take place and produces a set of CSP chains. Now, these CSP chains are stopping under the following conditions:

a) The CSP chain forms a loop;

b) An asymmetric configuration is encountered;

c) A face bounded by multiple loops is encountered.

The new configuration now is c. The second phase of the propagation process can take

place to proceed with the faces containing multiple loops. Because the loops have no status, they cannot be distinguished from each other, hence it is not possible to take advantage of external loops to reduce the complexity for determining the LS-CSPs.

This second phase processes the CSP chains satisfying condition c. Starting from the CSPs attached to the boundary of each loop bounding a face, the LS-CSPs of this face that coincide with these boundary CSPs are determined combinatorially (see Section 5.5). Processing these faces produces a set of LS-CSPs that form a list: $[\Pi_{LS}]$. Then, these CSPs can be used to carry on the propagation process and extend the corresponding CSP chains.

Regarding the example of Figure 6.8, the loops left during the first phase are now processed as illustrated in Figure 6.8c, d and e, f. Then, the generation of the LS-CSPs for these faces merges some of the existing CSP chains.

The above configuration describes a general scheme that can be improved if the loop status can be added to loops of each face. As described in the previous section and the beginning of this section, if the propagation process has to take advantage of the loop status, this status must be available right away at the beginning of the propagation. Then, each loop is assigned the status external or internal unless the loops are attached to a spherical face where no loop status can be assigned.

Having the loop status assigned, the boundary of M_{MAX} forms a set of adjacent loops since each edge of a loop is adjacent to two faces. Therefore, a loop has a status related to the face F where it lies in and a status with regard to its adjacent faces F_{ai}. Indeed, the status with regard to the adjacent faces is the same for all faces, i.e. either internal or external, since it refers to a property representing disconnected components. This means that a loop status, with regard to F_{ai}, has a unique status, i.e. $\forall(i,j)$, the edges shared with F_{ai} and F_{aj} have the same status. As a result, each loop has two status assigned, one for F and one for F_{ai}. Hence, the combinations deriving from that observation can be distinct couple formed from: external, internal, unknown.

Then, the first phase of the propagation process can use as initiators only the entities of loops having a status external/external to generate the CSP chains. Afterward, the second phase of the propagation can take advantage of the loop status available to determine the LS-CSPs that can be reduced to the set of valid CSPs that are attached to entities of the loops whose status is external with respect to F. As described at Section 5.5, exploiting the loop status reduces the complexity of determining the LS-CSPs that can carry on the extension of CSP chains. This extension is obtained by the insertion of LS-CSPs at the end of CSP chains and/or by merging together pre-existing CSP chains when an LS-CSP has to be added. Loops with a status internal or unknown with respect to its reference face are processed in a second row to take advantage of LS-CSP generation of their adjacent faces that reduces the number of CSPs.

Compared to the configuration where the loop status is not used, when the second phase uses this loop status, it produces an algorithm applicable even if not all the loops have a loop status of type external of internal: loops with an unknown status can be processed too. The complexity of the algorithm is not increased by this diversity of configurations, which confirms the interest of using the loop status to reduce the complexity of generating the valid LS-CSPs.

Within this second phase of the propagation process can be inserted the CSA processing since the axisymmetry appearing at some edges often reduces to the generation of multiple loops and cannot be treated in the first phase. It is the purpose of the next section to describe the specific aspects of propagating CSAs.

6.5 Axisymmetry processing during the first level propagation

In Chapter 2, axisymmetry analysis and detection has been stated as one objective of the shape symmetry analysis. Axisymmetry is characterized by an infinite number of symmetry planes intersecting each other along the same line. In order to study the propagation process of symmetry axes, a first step consists in studying the interactions between CSPs and CSAs when they appear over the same face, which is characterized by a multiple loop configuration over that face.

6.5.1 Interactions between CSAs and CSPs

When reviewing the interactions between the reference surfaces (see Section 5.4.3), CSAs appear when the edge forming the intersection curve between two surfaces is a full circle, then a CSA is attached to this edge. A CSA characterizes a symmetry property with an infinite number of symmetry planes cutting each other along the CSA.

When processing faces with multiples loops, a configuration can appear where one loop is a loop edge and has a CSA attached to it and another loop, a composite one, where O-CSPs, LB-CSPs are attached to it that form CSPs globally valid for this loop.

It has pointed out that a BS-CSP attached to an edge of a loop cannot be a CSP expressing symmetry properties valid for the face where the loop lies, all the other edges of this loop and all the other faces adjacent to these edges. Indeed, if a loop reduces to a loop edge, it means that the BS-CSP contains the whole loop, hence it cannot be valid to define a symmetry plane separating this loop into two halves. If the loop is a composite one, the previous observation attached to the edge containing the BS-CSP shows that this CSP cannot separate the loop into two halves. Consequently,

BS-CSPs cannot contribute to the generation of symmetry planes valid for one or more loops, i.e. in this last case, they don't need to take part to the LS-CSP generation process.

Focusing now on the interaction between CSPs and CSAs, Figure 6.9 is an illustration of an elementary configuration between them. The surface S_1 is planar and bounded by a circle and a square hole. These boundaries form two loops $\Gamma^{ex}_{1.1}$ and $\Gamma^{in}_{1.2}$. Without loss of generality, one can assume that the propagation process moves from the $O - CSP_{\Sigma_1}$, $\Sigma_1 \in \Gamma^{in}_{1.2}$ toward CSA_{Σ_5}. All entities belonging to $\Gamma^{in}_{1.2}$ are symmetric with respect to $O - CSP_{\Sigma_1}$. To propagate $O - CSP_{\Sigma_1}$ across the loops of S_1, Σ_5 and its loop $\Gamma^{ex}_{1.1}$ are selected for multiple loops processing. $\Gamma^{ex}_{1.1}$ has only one curve Σ_5, which is a circle, and has a CSA_{Σ_5} coinciding with $O - CSP_{\Sigma_1}$. Here, the objective is to assign the resulting symmetry to S_1. Because CSA_{Σ_5} coincides with $O - CSP_{\Sigma_1}$, the only common symmetry plane between CSA_{Σ_5} and $O - CSP_{\Sigma_1}$ is indeed $O - CSP_{\Sigma_1}$. As a result, the symmetry properties of S_1 reduce to the symmetry plane $O - CSP_{\Sigma_1}$, the CSA_{Σ_5} is no longer valid.

The geometric constraint for the propagation process between a CSA and a CSP, whatever its type, can be stated as: the axis defining the CSA must lie in the plane corresponding to the CSP. The resulting symmetry for the whole loops is characterized by: there is only one symmetry plane 'preserving' the CSP.

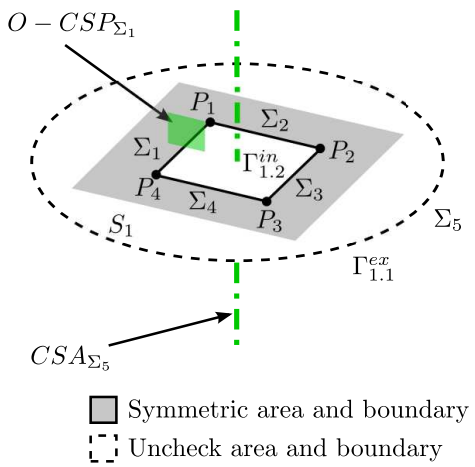


Figure 6.9: An example of multiple loops propagation within a face, between a CSP and a CSA: the CSA degrades into a CSP.

When a CSA interacts with another one that is parallel to it but not coincident, is another configuration where CSA(s) interaction reduce to a single CSP. Figure 6.18

is an example of this configuration. The model only contains CSAs. CSA_{Σ_2} does not coincide with CSA_{Σ_3}, so the axisymmetry cannot propagate across S_3. CSA_{Σ_2} and CSA_{Σ_3} are not collinear but parallel to each other, they reduce to a CSP. This CSP results from the propagation of the CSAs and connects them. Because all the faces of the object have been taken into account, their edges are also symmetric with respect to the CSP. Hence, the CSP is indeed a GSP.

The geometric configurations analyzed above cover all the configurations able to a CSP as a result of interactions between CSAs and CSPs. Now, the purpose is to describe the propagation process devoted to CSAs.

6.5.2 CSA propagation process

As a consequence of the previous section, axisymmetry detection is purely based on CSAs and the only geometric configuration enabling the propagation of axisymmetry is the collinearity of CSAs. This means at a global level, that M_{MAX} can benefit axisymmetry $\iff$ all its maximal edges reduce to loop edges and each of them is attached to a CSA and all these CSAs are collinear to each other. Then, these CSAs convert into a GSA (Global Symmetry Axis).

The integration of CSA processing within the first level propagation process can obtained as follows. The first phase can process some of the CSAs as CSP chain initiators since axisymmetric faces bounded by a single loop can appear for planes, cones and spheres. However, the propagation process can take place in only one configuration when the adjacent face to the first axisymmetric one is also bounded by a single loop. In this case, there cannot be any other CSA or CSP attached to the object, which means that the CSA becomes a GSA. Figures 5.14c and 5.17a are object examples falling in this case.

In all other configurations, the CSP chain initiators having axisymmetry property do not propagate because they are adjacent to faces having multiple loops. During the second phase of the propagation process, there may exist any number of CSP chains containing only their initiator that is a CSA. If there is none, it means that axisymmetric areas may exist but each of them is located in between non axisymmetric areas. If there is n, each of them defines an extremity of an axisymmetric area.

Then the second phase of the propagation process either expands the axisymmetric areas from their extremities, if any, or looks for the initiators of axisymmetric areas that must be chosen among the faces bounded by two loops whose types are loop edges and both have assigned a CSA. Starting from these initiators, the propagation operates using $G21$ to determine the adjacent faces. All the faces processed must have either two loops and a CSA assigned to each of them or they must be axisymmetry extremities, otherwise the axisymmetry propagation stops.

Figure 6.10 illustrates the axisymmetry propagation process. The model has CSAs only. CSA_{Σ_1} and CSA_{Σ_5} are CSP chain initiators obtained from the first phase. Following $G21$ content and choosing CSA_{Σ_1}, it propagates to F_2 through E_1. Because F_2 belongs to the correct class of face with two loops and CSA_{Σ_2} is collinear to CSA_{Σ_1}, the propagation takes place and $G21$ can be used to identify F_3. Then, the propagation carries on until it reaches F_6 that contains another axisymmetry extremity CSA_{Σ_5}. The propagation covers goes all the faces and edges. The analysis ends up with a GSA.

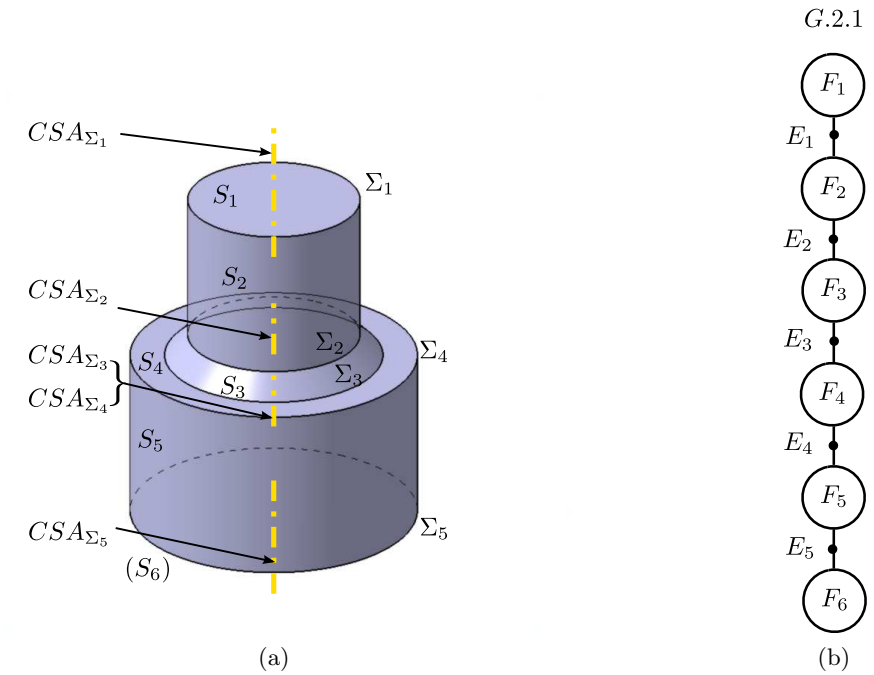


Figure 6.10: The axisymmetry propagation process and its connection with $G21$ content.

The CSA propagation process is mandatory but not sufficient to cover all the possible symmetry properties, another propagation process must take place that is related to LS-CSPs. First of all, the LS-CSPs must be initiated. Conditions for their existence have been stated in Section 5.4.3. These CSPs are always orthogonal to the CSAs and they must be propagated to determine whether or not an LS-CSP can be valid for the whole axisymmetric area. BS-CSPs are also CSPs that can initiate a propagation.

Assuming that the axisymmetric area extends over n faces, the following configurations have to be evaluated:

- The axisymmetric area has only one extremity: there is no LS-CSP or BS-CSP globally valid for this area;
- The axisymmetric area has two extremities. If n is even, the symmetry plane must a BS-CSP. The BS-CSP, if any, must be located at the $(\frac{n}{2}+1)^{th}$ edge in terms of adjacency position from either extremities. If n is odd, the symmetry plane must be an LS-CSP. If any, this plane is located at the $(\frac{n}{2}+1)^{th}$ face in terms of adjacency position from either extremities;
- The axisymmetric area has no extremity. The symmetry plane follows the same types and locations as previously whether n is even or odd.

Having defined and located the CSP, its propagation process can take place to check, on both sides of this CSP, if the faces are symmetrically laid out, i.e. they must be of same type, symmetrically located and with the same intrinsic parameters. If this propagation process succeeds to cover the axisymmetric area, this symmetry plane:

- Becomes a GSP if the axisymmetric area has two ends;
- Stays a CSP if the axisymmetric area has no ends. Here, the extremities of this area become the source of the propagation process of second level to carry on the symmetry analysis.

Indeed, during this propagation the initial CSP is compared with a new category of CSPs that will be addressed at Section 6.7. It has to be noticed that this propagation process must take place first during the second phase of propagation, before processing all the other faces with multiple loops.

6.6 Synthesis of the first level propagation process

Using the description of the previous processes devoted to phases one and two, the effect of using the loop status to speed up the generation of LS-CSPs, Figure 6.11 summarizes the major processes taking part to the first level propagation process.

First of all, CSP chain propagation starts with and propagates using the $[\Pi_O]$, $[\Pi_{LB}]$, $[\Pi_{BS}]$ lists for entities attached to loops having the status external for their reference face as well as for their adjacent ones. This process forms the first phase of this propagation process.

Then, the second phase starts with the CSA propagation. The $[\Pi_{SA}]$ list is exploited to extend the pre-existing CSP chains, which can stop the whole propagation process if M_{MAX} is axisymmetric because a GSA has been generated. The list $[\Pi_{LS}]$

of LS-CSPs is also initiated at this stage and exploited to create new CSP chains that can propagate and produce a GSP orthogonal to the GSA. At this stage of the second phase, all the axisymmetric objects have been processed.

The next treatment is now devoted to the faces with multiple loops to carry on extending the CSP chains. In a first place, the faces having a loop whose status is external with respect to their reference face, are processed and $[\Pi_{LS}]$ list is updated when LS-CSP are generated. Accordingly, when these CSPs are generated, $[\Pi_O]$, $[\Pi_{LB}]$ are decreased if the LS-CSPs coincide with O-CSPs or LB-CSPs or if these CSPs are no longer valid for the loops of the face being processed. Also, CSP chains are extended with these faces. Then, faces with multiples loops whose status is unknown with respect to their reference face. $[\Pi_{LS}]$ is updated when LS-CSP are generated and lists are decreased as long as the propagation, over each such face and with their adjacent ones, carries on. Finally, faces bounded by a single loop and having a status internal or unknown with respect to their adjacent faces, are processed to carry on the extension of CSP chains. Having processed the other categories of loops before takes benefit from the generation of LS-CSPs that removes CSPs from the $[\Pi_O]$, $[\Pi_{LB}]$ lists.

At the end of these two phases, the CSP chains obtained fall into one the following configurations:

- The CSP chain contains more than one item and it is a closed chain. This configuration describes a loop over M_{MAX}. The corresponding CSP can still generate a GSP;
- The CSP chain contains a unique item that is attached to an LS-CSP and this plane does not intersect any loop of the reference face. This CSP can still generate a GSP;
- The CSP chain contains a unique item that is attached to a BS-CSP and this plane entirely contains a loop of the reference face. This CSP can still generate a GSP;
- The CSP chain contains more than one item and it is an open chain. This CSP cannot generate a GSP.

Figure 6.11 highlights a last process since the CSP chains contain no more than one loop whereas the intersection between M_{MAX} and a CSP can generate several loops. There it is mandatory to merge the CSP chains lying in the same plane so that the second propagation process can be as efficient as possible and produces the correct number of CSPs. Without this merging process, several GSPs would be produced that would coincide, which incorrect. In addition, the second propagation process would cover twice the surface of M_{MAX}, which is not optimal.

It has to be noticed that Figure 6.11 indicates that even CSP chains containing open loops are subjected to the merging process because it is a contribution to the

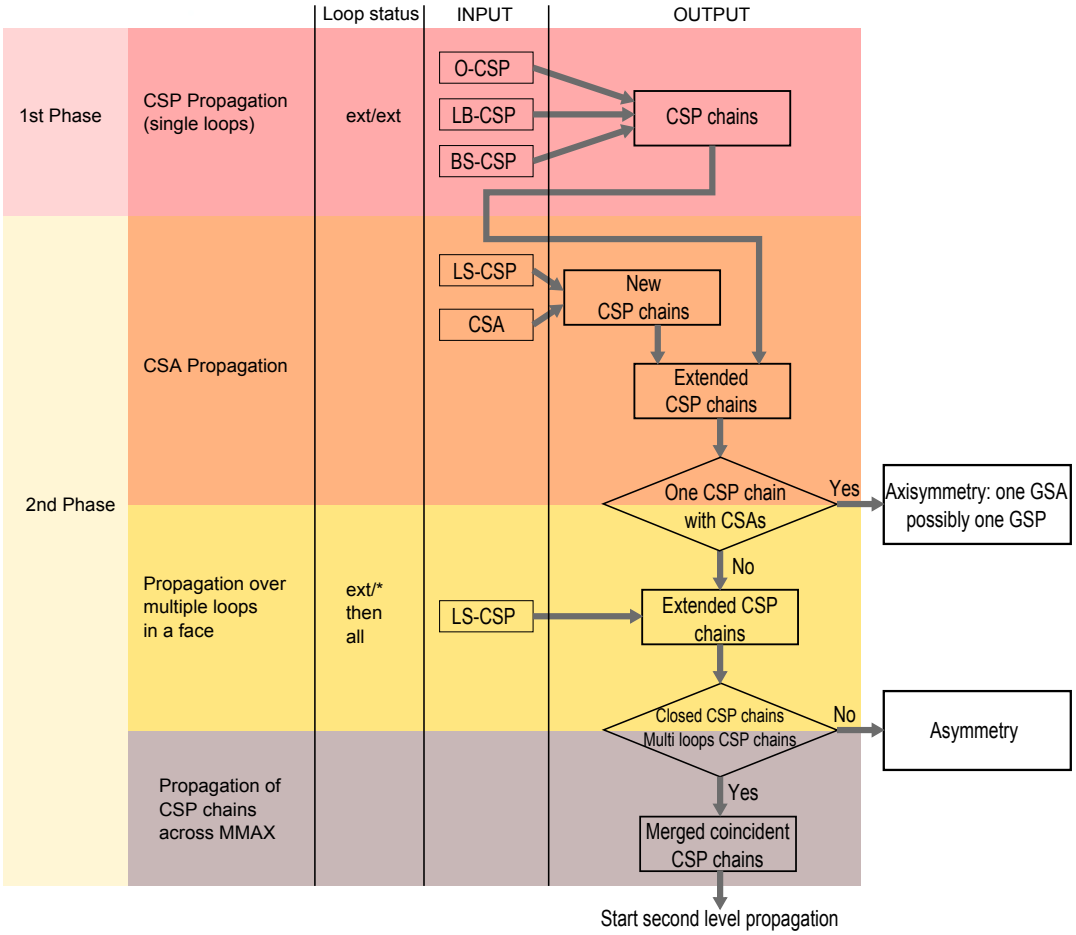


Figure 6.11: An overview of the first level propagation process with the major processes involved.

objective of the symmetry analysis to determine the areas of symmetry and asymmetry over M_{MAX}. If the global symmetry is the only objective, this process can be avoided.

As illustrated in Figure 6.11, the loop status is not mandatory to maintain the proposed process flow. Using the loop status is effectively a complementary process used to speed up the propagation.

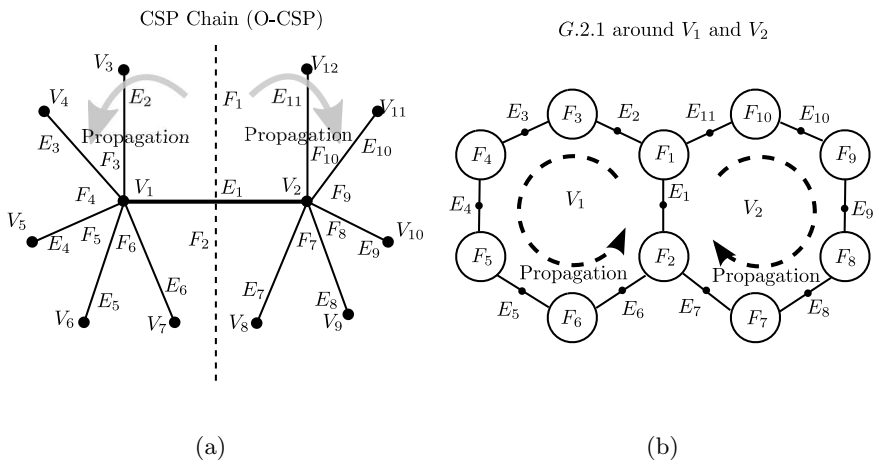
Now, the propagation process has to cover the areas of ∂M_{MAX} left: it is the purpose of the 'two sides' propagation that is described now.

6.7 Two sides propagation

From the unfolded drawings of Figure 6.8, one can clearly see CSP chains only valid for some of the edges of M_{MAX} only and some surface areas. Opposite edges of loops haven't been analyzed for symmetry as well as opposite surfaces that are adjacent to the faces already checked for symmetry. To validate the edges and faces left, there is a need for another propagation process that is named 'two sides' propagation.

6.7.1 Principle of the 'two sides' propagation process

Just as its name describes it, two sides propagation extends from both sides of the CSP chains generated by the first level propagation. This process is in fact similar to the propagation described for LS-CSP or BS-CSP that can be attached to a axisymmetric object. This process can be decomposed into two categories of configurations: faces bounded by a single loop and faces containing multiple loops. Figure 6.12 gives a first illustration of the principle of this propagation.



(a) (b)

Figure 6.12: The principle of 'two sides' propagation starting from an O-CSP: (a) is the local area of the B-Rep model around the O-CSP; (b) is the neighborhood of E_1 in $G21$ that represents the same area of the B-Rep model.

Let us now illustrate some details of this propagation process. The starting point is the content of a CSP chain originated from the first level propagation process. Assuming that the first element of this chain is an O-CSP attached to edge E_1 (see Figure 6.12a), it means that V_1 is symmetric to V_2. Similarly, F_1 is symmetric with respect to the O-CSP, as well as its adjacent face F_2.

Because it is an O-CSP, the two sides propagation can start from (V_1, V_2): the

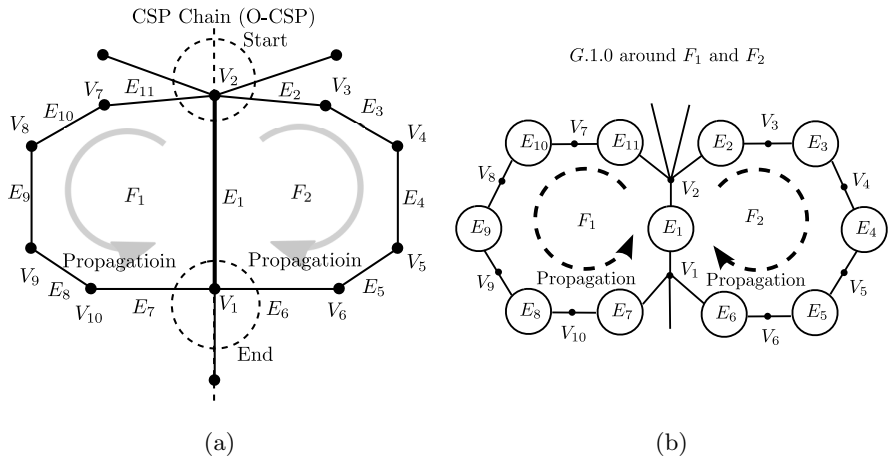
couple of reference vertices. The purpose of the propagation is to check whether neighboring faces of (F_1, F_2) are symmetric with respect to the CSP chain, or not. A first global condition holds in the numbers of faces surrounding V_1 and V_2, respectively. If they differ, their neighborhoods are asymmetric. Otherwise, the propagation carries on.

Within F_1, the loop attached to E_1 contains E_2 and E_{11} as adjacent edges to E_1 through vertices V_1 and V_2. Using again adjacencies, (E_2, E_{11}) determines the adjacent faces F_3 through E_2 and F_{10} through E_{11}. Consequently, F_3 and $F_{10}$are homologous, E_2 and E_{11} as well, i.e. in the hypergraph $G21$ these entities are symmetric with respect to F_1 (see Figure 6.12b). To finalize the symmetry analysis of this configuration, it is mandatory to use geometric informations. Thus, if the surface type of F_3 and E_2 are identical as well as their intrinsic parameters, if the location of F_3 is symmetric to F_{10} with respect to the CSP chain, then E_2 and E_{11} are also symmetric with respect to this CSP chain if and V_3 is symmetric to and V_{12}. Indeed, this last conditions is dropped down for now. The above process stands for one step of propagation and the symmetric area now extends to F_3 and F_{10}.

Now, from the hypergraph $G21$ or, more precisely from the dual loops of V_1 and V_2, the next edge pair is (E_3, E_{10}). With the same process, the symmetry area can propagate to F_4 and F_9, etc.

Coming back to the symmetry condition dropped for V_3 and V_{12}, the neighboring faces of F_1 and F_3 define edges that intersect with E_2 at V_3. Hence, if these neighboring faces satisfy the symmetry property, V_3 is also symmetric because it lies at the intersection of edges that are symmetric. It means that if this propagation process covers all the faces of M_{MAX}, apart from those already addressed during the first level propagation, all the surfaces forming the boundary of M_{MAX} are symmetric with respect to the CSP chain. Because, every edge of M_{MAX} is exactly adjacent to two faces, the intersection curves between these faces contain the edge of M_{MAX}. Hence, the symmetry of surfaces contains the symmetry of the curves supporting the edges. If so, benefiting of the symmetry property for the curves implies that some of the intersections of these curves defines the vertices of M_{MAX}. So, the symmetry property of surfaces is sufficient to assign the symmetry property to edges and vertices of M_{MAX} considered as independent sets.

However, edges bound faces to characterize the area of the surfaces used in each face of M_{MAX}. This interaction is not taken into account in the previous analysis. To be able to compare the used areas of surfaces with respect to a symmetry property, Section 5.4.6 has introduced the concept of orientation index. Consequently, if surfaces defining the faces of M_{MAX} are symmetric with respect to a CSP chain and if symmetric faces have the same orientation index, then it is sufficient to ensure the symmetry of M_{MAX} with respect to the CSP chain, i.e. the CSP chain generates a GSP.



(a) (b)

Figure 6.13: The principle of 'two sides' propagation starting from a BS-CSP: (a) is the local area of the B-Rep model around the BS-CSP; (b) is the neighborhood of E_1 in $G10$ that represents the same area of the B-Rep model.

The above analysis shows that the combination of the first and second propagation processes entails the algorithm to terminates and to produce a correct answer with regard the existence of GSPs.

The above description of the propagation process has started from an O-CSP but other initial configurations can be encountered: BS-CSPs and LB-CSPs. Considering the configuration with a BS-CSP (see Figure 6.13), the edge E_1 it is attached to lies into the CSP chain. Faces F_1 and F_2 sharing E_1 have already been checked for symmetry (see Figure 6.13a). To check the symmetry of other faces, it is mandatory to identify a pair of edges. Let us select V_1, an arbitrary vertex of E_1. Then, looking for edges adjacent to F_1 and F_2 in hypergraph $G10$, the subset of $G10$ reduced to F_1 and F_2 uniquely defines the couple (E_2, E_{11}) (see Figure 6.12b). From then, adjacent faces can be identified as input for the propagation process.

If the starting item is an LB-CSP, the previous analysis can be extended to the extreme vertices of the two edges defining the LB-CSP.

After current iteration of the propagation process, a new couple of faces can be identified when exploring the faces around the couple of reference vertices: (V_1, V_2) for the O-CSP and (V_1, V_1) for the BS-CSP. If all the faces have been visited, then the propagation can move symmetrically to new reference vertices using edges like E_2 and E_{11} of O-CSP configuration, to find new faces. If there is none, a new couple of reference vertices can be found using the unvisited faces.

Applying this propagation to the elementary example of Figure 6.8 produces the result illustrated in Figure 6.14 where the dotted lines indicate the loops that have

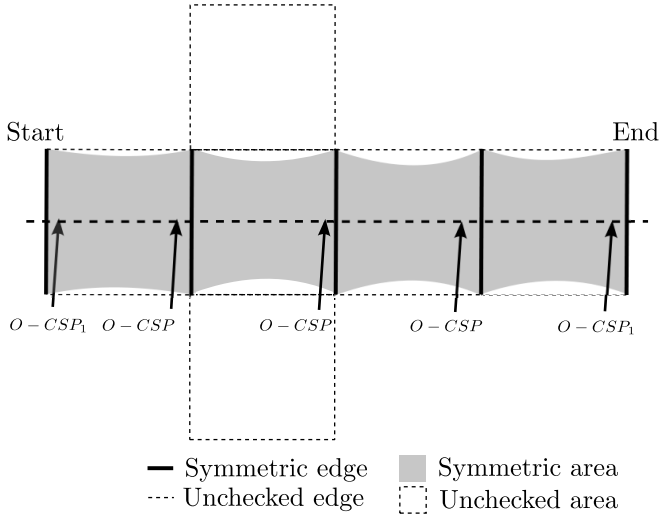


Figure 6.14: The unfolded drawing of Figure 6.8a and e after 1st propagation.

not been taken here since the propagation process has been described only for faces bounded by a single loop. The configuration of faces with multiple loops is addressed at Section 6.7.3.

The principle of the 'two sides' propagation process compares the relative position of a couple of surfaces with respect to a CSP chain. Indeed, this appears to be a new category of CSPs.

6.7.2 Introducing the SS-CSP category

The 'two sides' propagation process compares two faces F_1 and F_2 of M_{MAX} that don't share any other edge or vertex of M_{MAX}: i.e. there is no B-Rep adjacency relationship between F_1 and F_2 (see Figure 6.15). This is new compared to:

- O-CSPs: two faces share a common edge;
- BS-CSPs: two faces share a common edge that lies in the symmetry plane;
- LB-CSPs: three faces share two common edges that are symmetrically set with respect to the symmetry plane of the reference face;
- LS-CSPs: at least two loops of a face share a common symmetry plane that belongs to one of the symmetry planes of the external loop of this face.

Because intersection curves between the reference surfaces of F_1 and F_2 always exist in the existing categories, the Surfaces Symmetry candidate symmetry plane

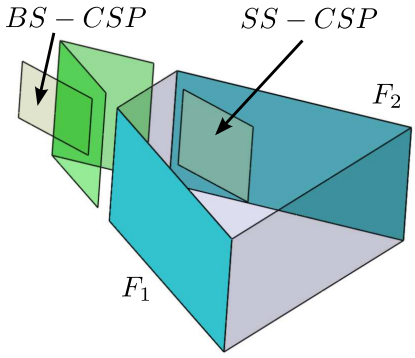


Figure 6.15: An example of SS-CSP illustrating the difference of interaction between the surfaces F_1 and F_2 compared to a BS-CSP.

(SS-CSP) characterizes configurations where the reference surfaces of F_1 and F_2 may not intersect at all. This is particularly true for spheres and tori. Anyhow, if they intersect their intersection is not part of M_{MAX} so that F_1 and F_2 do not share any edge or vertex in M_{MAX}.

Indeed, the 'two sides' propagation process checks that the surfaces on both sides of a CSP chain share the same SS-CSP, which is defined by the CSP chain.

Now, considering the geometric constraints producing these SS-CSPs, they are very close to those related to BS-CSPs. The conditions for BS-CSPs must, and effectively, incorporate the existence of an intersection curve whereas this condition is no longer necessary for SS-CSPs. SS-CSPs must also incorporate configuration where the reference surfaces of F_1 and F_2 coincide exactly, i.e. the intersection between F_1 and F_2 is a surface.

6.7.3 Processing faces with multiple loops

The generation of CSP chains and the 'two sides' propagation process introduced in this section strongly rely on adjacency relations available in the hypergraphs. However, the generation of LS-CSPs is a first example where no adjacency relations are available between the loops forming the boundary of the face where the LS-CSPs are generated.

Indeed, this configuration is fairly similar to generation of LS-CSPs. The initial configuration expresses the lack of connectivity between loops but the existence of an LS-CSP structures the loops. It becomes possible to identify:

- The edge E_α that is symmetric to a reference edge E_β of the same loop if the LS-CSP halves this loop;
- The loop Γ_α that is symmetric to a reference one Γ_β if the LS-CSP separates

them without intersecting them. Within Γ_α it becomes also possible to identify the edge E_δ that is symmetric to a reference edge $E_\eta \in \Gamma_\beta$.

These are indeed adjacency relations. Therefore, when the 'two sides' propagation process encounters faces with multiple loops, it can take advantage of these new adjacency relations generated together with the LS-CSPs. This propagation process over couples of faces F_1 and $F_2$with multiple loops can be summarized as follows, once the reference surfaces of F_1 and F_2 have been checked against the current SS-CSP:

- If F_1 and F_2 have a different number of loops, the propagation stops;
- If F_1 and F_2 have the same number of loops, the lists of LS-CSPs of F_1 and F_2 is searched for any couple of planes (Π_{LS1}, Π_{LS2}) that are symmetrically located with respect to the current SS-CSP (see Figure 6.16). If (Π_{LS1}, Π_{LS2}) exists, the 'adjacencies' between Π_{LS1} and the internal loops of F_1 can be used to carry on the 'two sides' propagation process. Because Π_{LS2} plays the same role for F_2, it can be used with its 'adjacencies' to locate the loops and the entities within these loops that must be symmetric to F_1. During this phase, 'two sides' propagation process applies as described at the beginning of this section. This propagation may fails and the purpose stays the coverage of the whole faces of M_{MAX}. An example of this propagation is given in Figure 6.17;
- If F_1 and F_2 have the same number of loops and the lists of LS-CSPs of F_1 and F_2 don't produce any couple of planes (Π_{LS1}, Π_{LS2}) symmetrically located with respect to the current SS-CSP. In this case, the use of an LS-CSP to cover the internal loops of F_1 and F_2 is not possible. It becomes necessary to generate the connections between the loops inside each face and across each face to find homologous faces if any. These connections are generated on a combinatorial basis since there is no property similar to the existence of CSPs attached to the external loop of face F_1 or F_2 and used to reduce the number of iterations.

The above analysis of configurations shows that the 'two sides' propagation process reduces to propagation through faces. This means that its complexity is linear with respect to the number of faces of M_{MAX}. It has to be pointed out that M_{MAX} being a 2-manifold model, the Euler theorem applies and shows that the propagation process through faces is most efficient than its counter part based on edges.

Complementarity, the analysis of the 'two sides' propagation process operates mostly in constant time when moving from one face to another since adjacency relations are available most of the time and if not available, connections between faces can be obtained for faces with multiple loops when they contain LS-CSPs. It is only in the configuration of faces with multiple loops without LS-CSP that the propagation mechanism becomes quadratic for some entities. This last configuration is really a worst case.

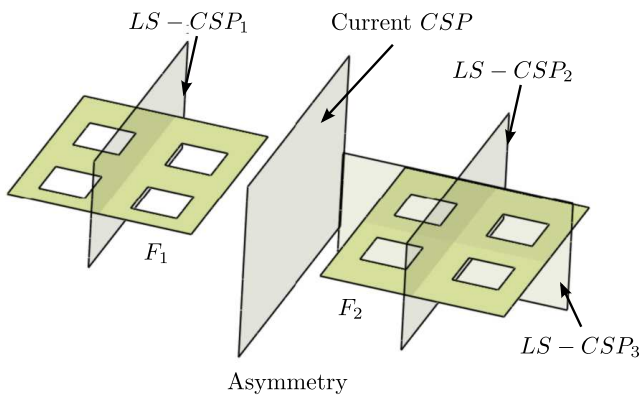


Figure 6.16: The use of LS-CSPs as a reference to structure faces with multiple loops and the symmetric condition to satisfy in the 'two sides' propagation process.

Coming back on the propagation of first level, a similar analysis can be conducted since propagating over a loop or crossing a vertex is achieved in constant time. Again, some restricted number of configurations can produce a quadratic complexity. This process is linear with respect to the number of faces and vertices.

Overall, the conquer phase can be regarded as mostly linear with respect to the faces and vertices of M_{MAX}. The divide phase indicated a similar behavior, which leads to a global behavior for the analysis of M_{MAX} with respect to a CSP that is a process of mostly linear complexity with respect to entities defining M_{MAX}.

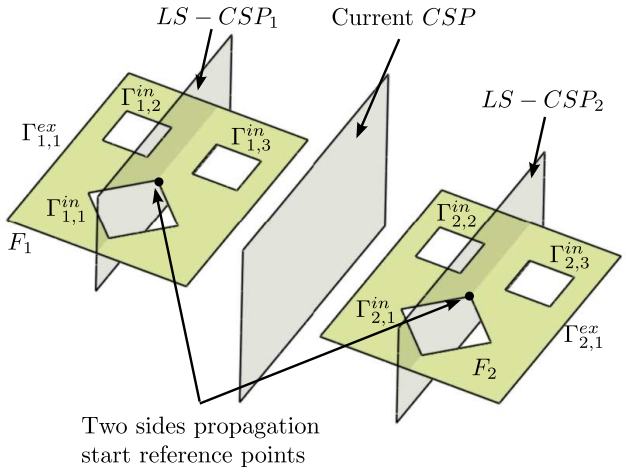


Figure 6.17: An example of 'two sides' propagation process highlighting the use of the 'adjacencies' obtained from the LS-CSPs.

6.8 Highlighting asymmetry in M_{MAX}

The propagation process is the conquer phase of the symmetry analysis of M_{MAX}. Whether the first level propagation process or the 'two sides' propagation is under focus, their basic principle is close from each other. They start from any CSP or CSA belonging to an edge, a vertex or a face or they start from a CSP chain and extend it through the adjacencies available in the hypergraphs.

If these propagation mechanisms succeed in finding a CSP or CSA or another SS-CSP coinciding with the reference plane, Π, the propagation goes on. Whenever the propagation cannot find a CSP or CSA or another SS-CSP coinciding with Π, the propagation stops and shows that Π cannot get the status of GSP or GSA. Throughout the description of the propagation phase, the generation of CSP chains as well as the 'two sides' propagation have been presented to define the maximal extent of each of them: CSP chains are merged even though they are not closed and the 'two sides' propagation has been stated with a stop criterion that is the complete coverage of M_{MAX} faces.

Indeed, if the focus of the symmetry analysis of M_{MAX} is placed on the identification of GSPs or GSA, every time the propagation cannot find a new CSP or CSA or another SS-CSP, the conclusion is clear for the corresponding reference plane, it cannot be a GSP or a GSA. Then, the conquer can stop and move on with the next reference plane. Such a conquer process does not change the algorithm complexity but the propagation will terminate much earlier and this can significantly speed up the generation of GSPs and GSA. This behavior matches the first objective stated at

Chapter 2.

When the first level propagation spreads until both sides of a CSP chain cannot propagate and then, merging the CSP chains whose reference planes coincide, is able to divide the intersection between M_{MAX} and the corresponding reference plane Π is two areas: one area symmetric with respect to Π and another one that is asymmetric. It has to be noticed that the symmetric and asymmetry parts are precisely defined with respect to the faces, edges and vertices of M_{MAX} since the CSPs and CSAs are initiated at every entity of M_{MAX}. From there, the generation of the maximal CSP chains defines the continuous arc of the intersection $\Pi \cap M_{MAX}$. Then, merging all the CSP chains coinciding with Π generates all the arcs belonging to $\Pi \cap M_{MAX}$, which corresponds to the symmetric area of $\Pi \cap M_{MAX}$. This symmetric area, if formed by a set of arcs, cannot contained isolated points because the only CSPs attached to vertices are LB-CSPs but their existence derives from the symmetric configuration of some of its adjacent faces. Hence, if an LB-CSP exists, there is a non null arc belonging to $\Pi \cap M_{MAX}$.

Now, when considering the 'two sides' propagation starting from the previous CSP chains, using the entire coverage of the faces of M_{MAX} ensures that the faces can be categorized into symmetric or asymmetric with respect to Π. Whatever, the shape of the symmetric areas, they are obtained from this propagation process. However, classification of the faces does not include their boundaries. Section 6.2 has shown that the symmetry area of an O-CSP can be an unbounded domain. Consequently, the 'two sides' propagation does not provide a precise answer to the second objective. This is also rooted in the fact that the 'two sides' propagation uses some of adjacencies between faces, i.e. not all the edges of M_{MAX} take part to the propagation process. Hence, the categorization of edges into symmetric or asymmetric is incomplete.

A simple asymmetric case (see Figure 6.18) illustrates the prominence of symmetry information about face boundaries to define, as clearly as possible the symmetric/asymmetric areas.

To reach the second objective, complementary treatments are mandatory that focus on the symmetry properties of edges and some complement about SS-CSPs. The edges of M_{MAX} that have not been visited during the conquer phase must be visited to classify them. Section 6.2 shows that the symmetry status of edges involves two couples of faces in the 'two sides' propagation. This forms the basis of the treatment needed. However, if the two couples of adjacent faces are already symmetric with respect to Π, the two common edges are also symmetric and can be categorized as such even though they were not visited. The categorization of edges requires a real treatment only for edges located at the boundary of symmetric areas.

Another treatment is related to SS-CSPs. At present, the symmetric / asymmetric areas that can be identified are attached to the SS-CSPs that coincide with the planes obtained from the CSP chains. To provide a complete answer to the second objective,

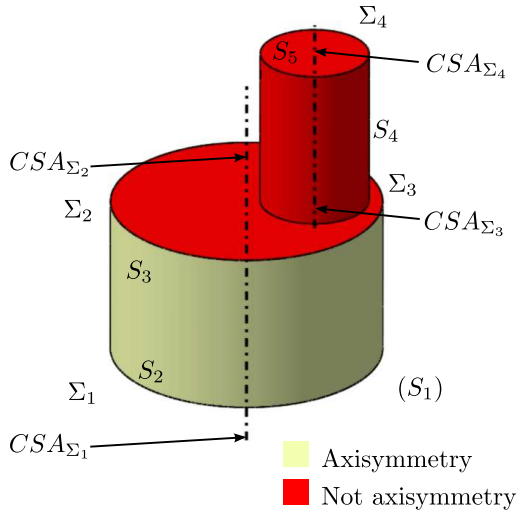


Figure 6.18: The area with red color is not axisymmetric when the propagation is attached to $CSA_{\Sigma 1}$. S_3 attached to $CSA_{\Sigma 2}$ is axisymmetric with respect to $CSA_{\Sigma 1}$ but it is asymmetric when attached to $CSA_{\Sigma 3}$, hence the prominence of more precise boundary status.

it is mandatory to consider the configurations where there exist a SS-CSP attached to one couple of surfaces, at least, but this SS-CSP does not coincide with any CSP chain (see Figure 6.19).

In this case, the goal is to find two sets of faces, not adjacent to each other, whose intrinsic parameters are identical on a two by two basis. These two sets must share the same SS-CSP and this one should not coincide with the planes already defined by the CSP chains. Because of time, this matter has not been investigated further.

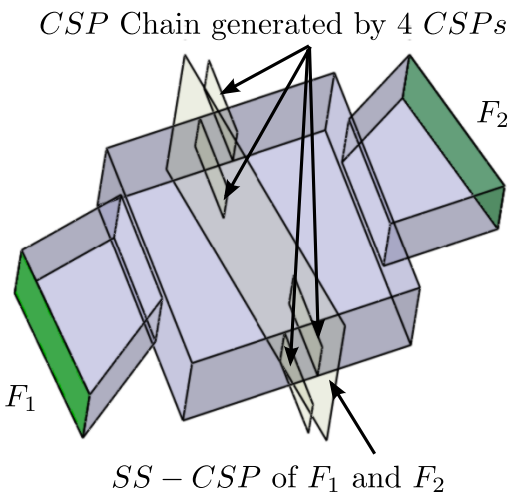


Figure 6.19: An example where the SS-CSP does not coincide with the available CSP chains.

6.9 Conclusion

CSPs and CSAs generation is part of the divide process. The propagation is a conquer process that has been decomposed into two levels. The first one is based on the necessary intersection between GSPs or GSA and a volume object, it creates CSP chains and propagates them. Because it starts from the O-CSPs, LB-CSPs and BS-CSPs that enumerates all the possible interactions between GSPs, GSA and M_{MAX}, it is a necessary and sufficient process to identify all the GSPs, GSA of M_{MAX}.

The second one propagates the symmetry property over each side of CSPs, it is the 'two sides' propagation starting from the CSP chains obtained from the first level. Because the results of the first level enumerate all the possible CSPs, the GSPs obtained after this propagation process are effectively all the GSPs of M_{MAX}.

Axisymmetry propagation has been set as part of the first level since CSA propagation is a unidirectional process. As a result, all the global symmetry properties as well as all the local ones leading or not to the global ones are also available to characterize M_{MAX}. The corresponding conquer process appears to be a process of linear complexity with respect to the faces, edges and vertices of M_{MAX}. The topology of M_{MAX} evolving linearly with respect to these entities, any volume defined from the reference surfaces can be processed with a linear complexity. The only treatment that does fit into this schema is the particular case of some faces with multiple loops where quadratic complexity may be necessary.

It has been demonstrated also that this divide and conquer process can nearly reach the second objective with the algorithmic complexity since this process can define the

symmetric area of M_{MAX} for any CSP chain, hence the corresponding asymmetry over M_{MAX}. However, the introduction and analysis of the new CSP category, the SS-CSP, shows that not all the possible symmetric / asymmetric areas over M_{MAX} can be obtained from the divide and conquer algorithm. To reach entirely the second objective a complementary process needs to be set up.

The purpose is now to illustrate and to evaluate the efficiency of the proposed approach through various examples.

Chapter 7

The examples and applications for reflective symmetry detection

After the detailed description of the divide and conquer algorithm, the whole algorithm of reflective symmetry analysis is available and justified. In this chapter, several examples are presented to illustrate the process of reflective symmetry analysis and to evaluate it. From the examples, the algorithm achieves the global symmetry plane detection, axisymmetry detection and highlight most of the asymmetric areas. The models selected are all manifold and without complex tangent configurations. The more examples and details left to Appendix C.

7.1 Introduction

The software environments for testing the symmetry detection algorithm have been created on two systems. The first one is developed using the Macro facility of CATIA V5 R19 under Windows operating system, the status of computer is Inter(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz and with 3.24G memory. The model input is created within CATIA or it can be imported in that environment as a STEP file. Because CATIA B-Rep data structures are not opened at the macro level, all the informations about faces, edges, vertices as well as their adjacency relationships cannot be accessed. The preparation phase with simple treatments such as surface type recognition and surfaces intrinsic parameters extraction, etc., can be performed using CATIA macros using the CATIA 'measure' function or through new functions dedicated to the symmetry analysis process.

Because the direct access to B-Rep datastructure is not possible, the conquer phase of the symmetry analysis could not be coded with the schema described in the previous chapters. Hence, the time complexity of this implementation has not been studied because it is no relevant. Furthermore, the CATIA Macro execution environment is much slower than the same functions directly implemented in the modeling kernel, which makes time measurements non representative of the algorithm efficiency.

Indeed, the major interest of CATIA Macros, is a rather easy access to the history tree of an object, which a good environment to illustrate the possible reorganizations

in a history tree when symmetry properties have been identified.

The hypergraphs construction, the divide phase with the CSPs collection have been implemented with algorithm variants because of the limitations set by the available macros and the lack of access to the B-Rep datastructure of the object. Because CATIA surface selection behaves nearly like the identification of maximal surfaces when an object has been entirely generated within CATIA and because of the lack of access to the B-Rep datastructure, it has not possible to implement the surface preparation process with all the merging configurations described, i.e. it is not possible to process all surface connections with the CATIA Macro environment.

The software system uses OpenCascade CAD software library available on Linux and Windows. It runs on a labtop with Intel(R) Core(TM)2 CPU T7200 @ 2.00GHz and 2G memory. From this library, a dedicated application software has been developed that perform the symmetry analysis of an object. It reads STEP files directly as input and follows the processes described in the previous chapters. The object is described in the STEP both from a geometric and topological point of view.

The examples and figures that will be shown in the next sections were generated on both systems. However, the time statistics are created on the second software system only: OpenCascade, for the reasons mentioned above about the limitations of CATIA Macro execution environment.

7.2 Global symmetry detection

The reflective symmetry detection algorithm starts with a STEP file input. The STEP file has been exported from a CAD software. The first example is a bearing holder illustrated at Figure 7.1)a that represents the initial B-Rep model. The first work is to generate the initial hypergraph and at same time to read the surface parameters. Figure 7.2 shows the three initial hypergraphs. There are 186 reference surfaces and 446 curves, hence as many faces and edges. Next, comes the processes for maximal faces generation and maximal edges generation. The initial hypergraphs change to Figure 7.3 and the model M_{MAX} contains now 142 faces and 273 edges. It has 63 planes, 61 cylinders, 2 cones, 3 spheres and 13 tori. Figure 7.1)b, c, d, e show the different maximal faces.

From the hypergraphs, loop structures are extracted. Then, the divide phase starts with the collection of the O-CSPs, BS-CSPs, LB-CSPs and CSAs. Figure 7.4a gives the view with all 436 CSPs that will be used during the conquer phase. The set of CSPs decomposes into 252 O-CSP, 68 BS-CSPs, 6 LB-CSPs, 33 LS-CSPs and 77 CSAs. The LS-CSPs generated during the first level propagation process are also part of this figure. Figure 7.4b though f give a representation of each category of CSP.

The first level propagation process starts from one CSP within the group of O-

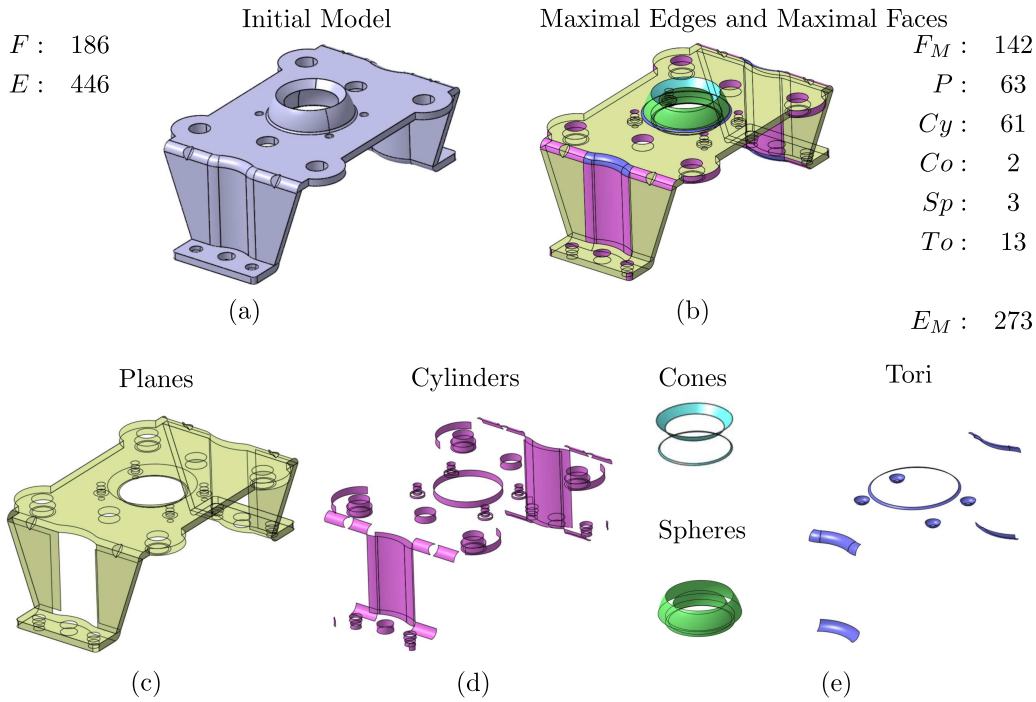


Figure 7.1: A reflective symmetry detection example: (a) is the initial model; (b) colored view highlighting the different categories of reference surfaces; (c)-(e) view of reference surfaces by category.

CSP, BS-CSP and LB-CSP. It merges all coinciding CSPs and CSAs and should cover all faces which intersect with these CSPs (CSAs). Figure 7.5a shows a result of this propagation propagation process for one O-CSP. The red surfaces are all valid, i.e. symmetric with respect to this CSP, based on their intrinsic parameters. Except for some edges not taken into account during this propagation process, all areas of the red surfaces are symmetric with respect to the current CSP. After this first level propagation has ended, the second level propagation starts, i.e. 'two sides' propagation process. It validates the symmetry properties of the object extending at the left and the right of the CSPs resulting from the first level. Figure 7.5b illustrates this process. In fact, the initial CSP is a real symmetry plane. So, the second level propagation process covers successfully the rest of faces of the object and this CSP changes its status to a GSP. Figure 7.5c is the result of the symmetry detection process. There are two global symmetry planes.

The total time cost of the symmetry detection of this example running on the OpenCascade CAD environment is 384ms. Except the hypergraph creation and loops preparation, the time for symmetry detection, i.e. the divide and conquer phases, is about 200ms.

The complexity of the algorithm is not easy concluded by a simple function. It can be divided into different process:

- Preparation process covering: hypergraph generation, maximal faces generation, maximal edges generation, loops extraction and loops status analysis;
- Divide phase: generation of the four categories of CSPs and CSAs;
- Conquer phase: propagation processes of first and second levels.

For a manifold model input, this model contains n_F faces, n_E edges and n_V vertices. To generate the initial hypergraphs is a mapping process and has to go through all the elements of the model. So, the complexity of this initialization is $O(n_F + n_E + n_V)$. The maximal faces generation checks whether two adjacent faces through a common edge are homologous or not, which is based on the number of edges. Similarly, maximal edges generation is based on vertices. The maximal faces generation using the regular vertex split operator needs to scan all vertices of the model and it is linear process with respect to the number of vertices. However, it has to be noticed that this process incorporate some complexity since it to scan the edges and faces around the reference vertex. The corresponding complexity has not been studied. Then, the loop extraction process needs to cover all the faces. The detail of this process uses the help of hypergraphs: they provide the adjacent edges and vertices which are already linked as loops. Right now, the previous processes cover all the elements of the B-Rep and these processes differ in time depending on the category of entity addressed. In a first place, this complexity is $O(n_E) + O(n_V) + O(n_F)$. Regarding the loop status analysis, in worst case, the algorithm needs to analyze all the edges in order to get their status with respect to their two adjacent faces. This process can be represented with complexity of $O(n_E)$. Finally, summing up all the preparation processes applied to the input model, the complexity can be represented by $O(n_F + n_E + n_V)$, as a first approach since the processing time vary for the faces, vertices and edges.

Then, it is the divide phase with the CSP generation process. From the algorithm description, O-CSP and BS-CSP (including CSA) generation varies linearly with respect to n_E. In the worst case, there will be one O-CSP and one BS-CSP generated for each edge. So, the number of CSPs is $2n_E$. The LB-CSP generation needs to process all the vertices of the model. The complexity is $O(n_V)$. However, the number of LB-CSPs is hard to estimate. To describe precisely, this number is linked to the number of faces around each vertex. Here, a parameter p is added to describe the maximum number of adjacent faces around a vertex. So, pn_V of LB-CSP are generated in the worst case. The number of LS-CSPs depends on the number of CSPs attached to the external loops of faces. If the external loop is axisymmetric or if all the loops are assigned an unknown status, the number of LS-CSPs depends on the number of loops. A sphere model with many circular holes can be considered as the worst case. All

loops on this surface are of type unknown. In this case, generating the LS-CSPs is a combinatorial process. The corresponding complexity is $O(n_E^2)$ and the number of LS-CSP is $(0.5n_E^2 - 0.5n_E)$ when all edges are loop edges. Except some special and very symmetric models, from the experience of testing different mechanical models, the most populated category of CSPs is the O-CSP and then it is the BS-CSP.

The last part is conquer phase using the propagation processes. After covering all the faces of the model, the propagation process stops and the current CSP is upgraded to global symmetry plane. So, the complexity of this process for one CSP is $O(n_F)$. In the worst case, each CSP is going to through this propagation process. Consequently, if there are n_{CSP} CSPs, the total complexity is $O(n_{CSP}n_F)$. However, there are several aspects of the algorithm to take into account that significantly reduces this evaluation. During the first level propagation process, when a CSP coincides with a reference one, the new one is removed from the list of CSPs, so lists of CSPs decrease during the propagation process. Then, the number of resulting CSP chains is much smaller than the initial number of CSPs. Finally, the second level propagation process takes place after having merged the CSP chains and only the CSP chains containing loops can generate GSPs. So, the real time cost is far less then $O(n_{CSP}n_F)$.

The complexity of the whole algorithm is hard to estimate, but the example shows that in the OpenCascade CAD environment, it is a very quick process. Figures 7.6 and 7.10 give another global symmetry detection example. Its total time cost is 211ms.

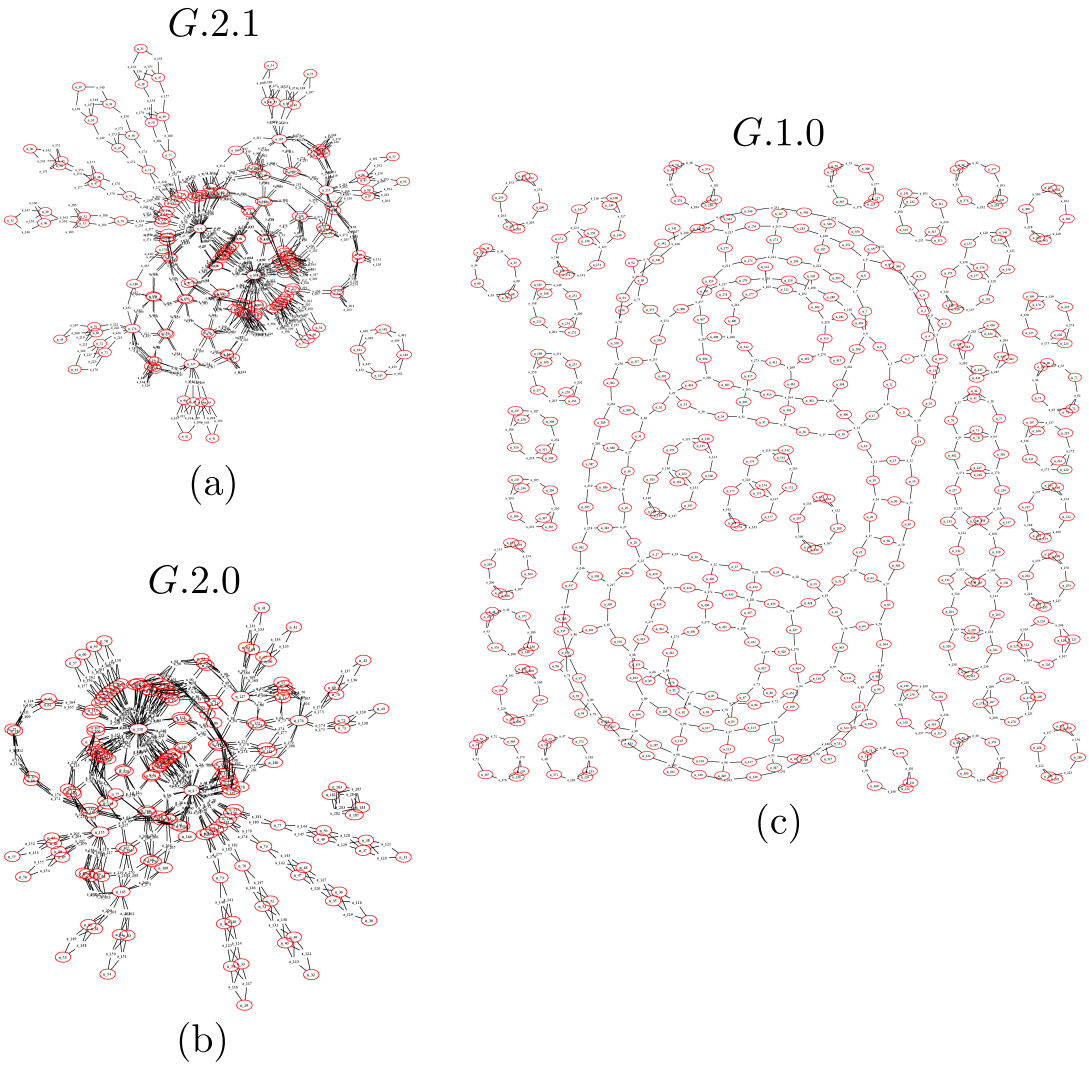


Figure 7.2: The initial hypergraph of the Figure 7.1.

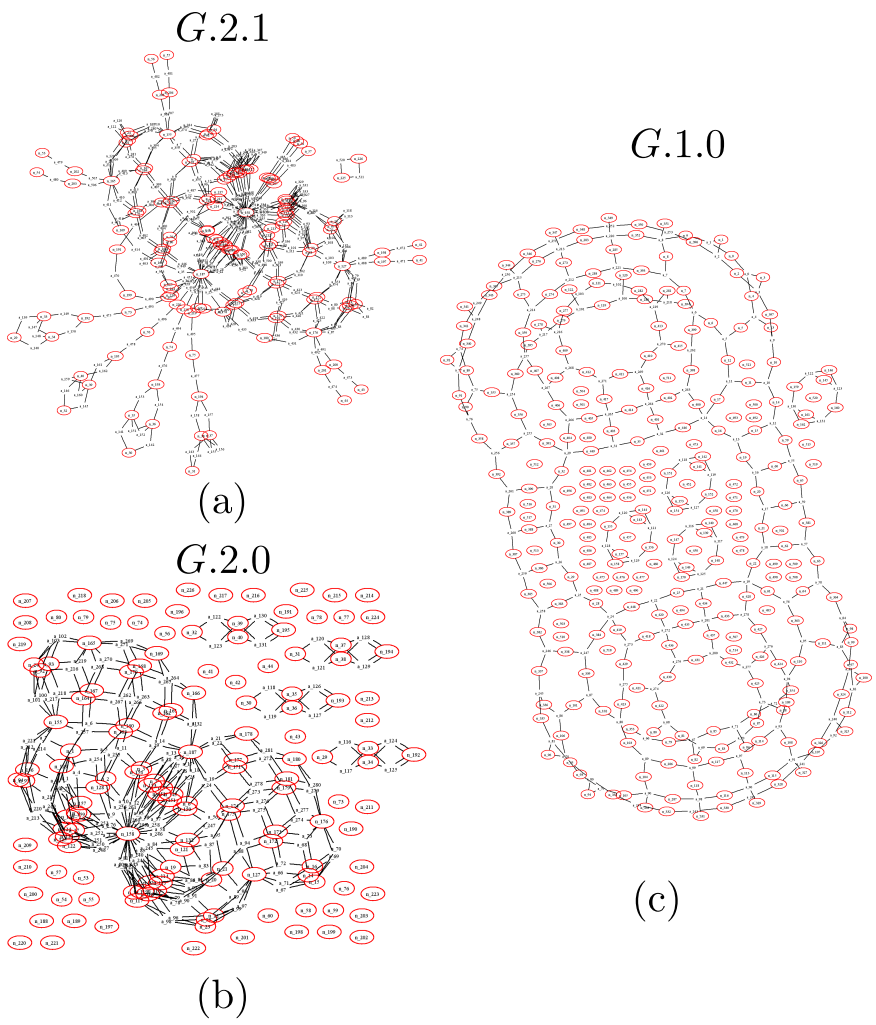


Figure 7.3: The hypergraph after maximal faces and maximal edges generation of the Figure 7.1.

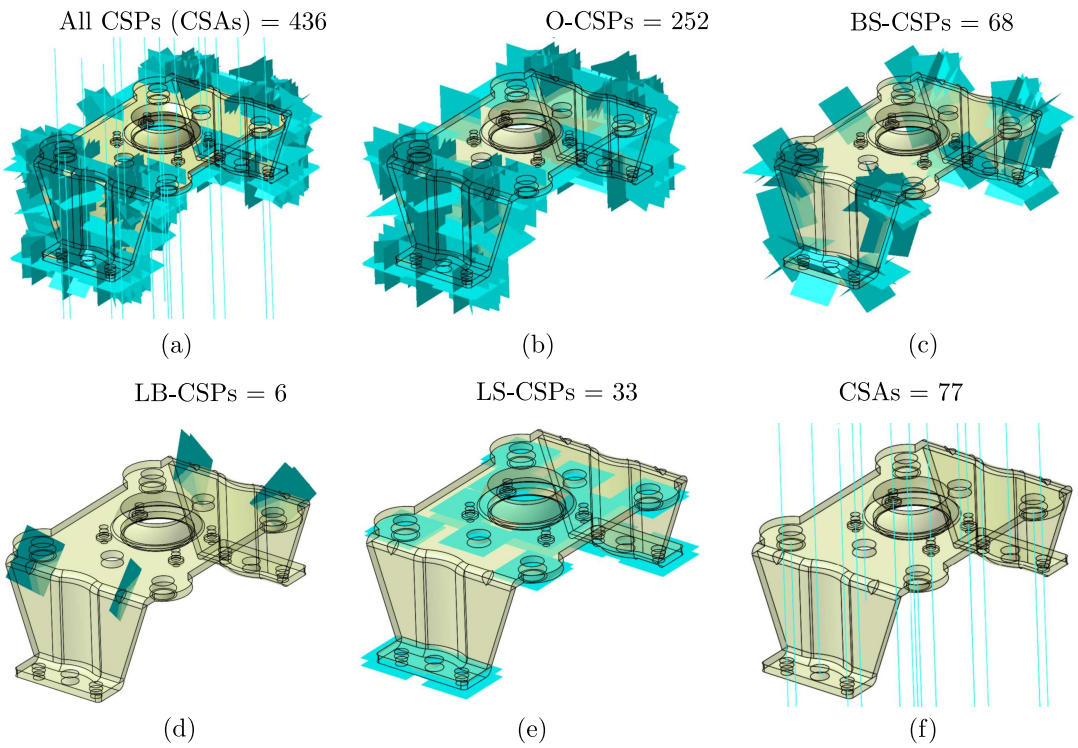


Figure 7.4: The set of CSPs of the example in Figure 7.1.

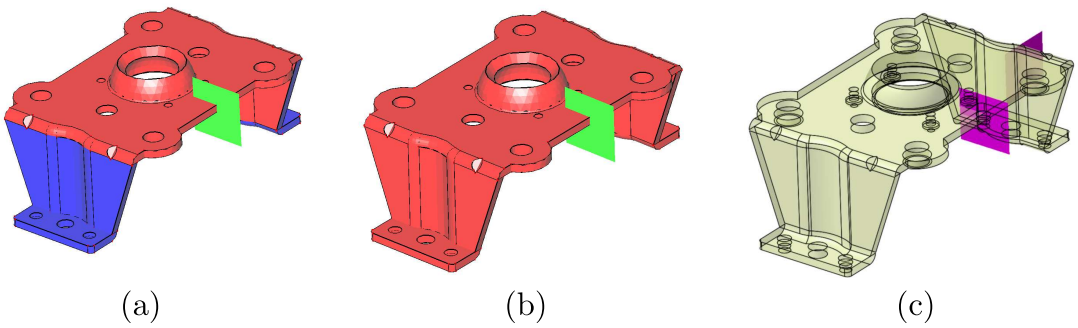


Figure 7.5: The conquer phase applied to the example of Figure 7.1: (a) first level propagation process, (b) second level propagation, (c) the result about global symmetry planes.

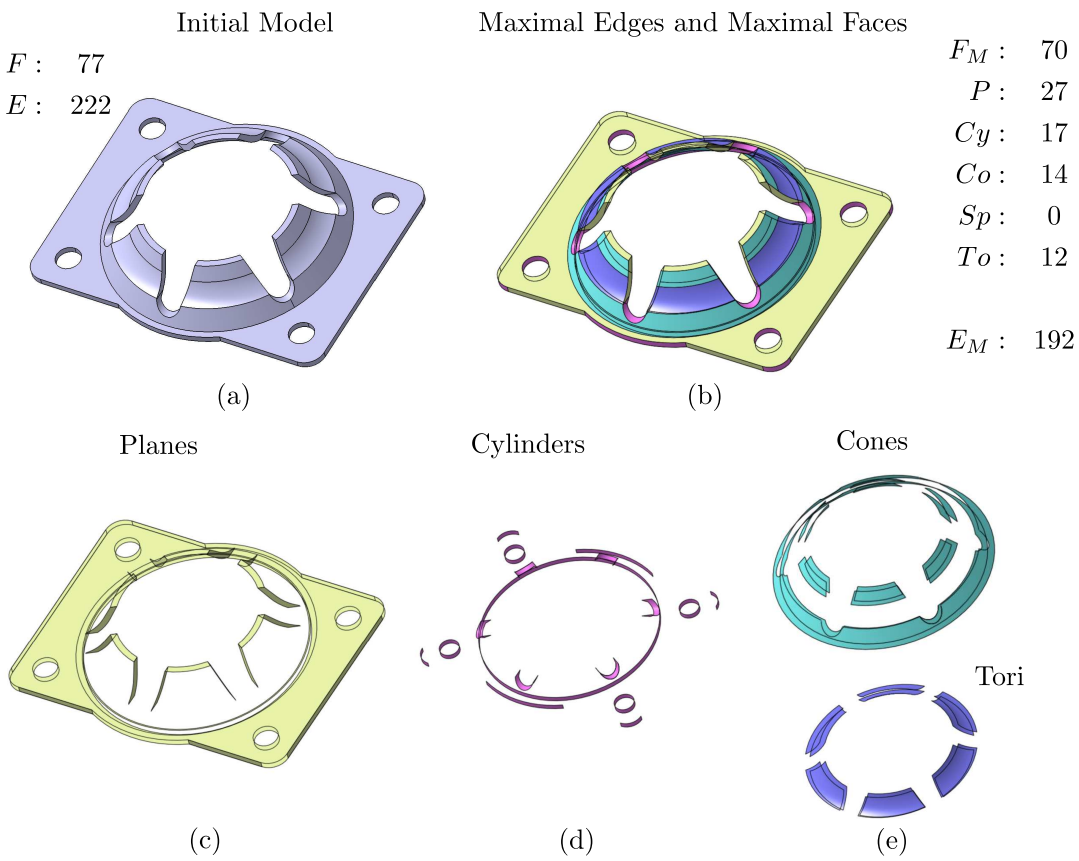


Figure 7.6: A reflective symmetry detection example: (a) is the initial model with the statistics about the faces, edges and vertices; (b)- (e) the views of different reference surfaces.

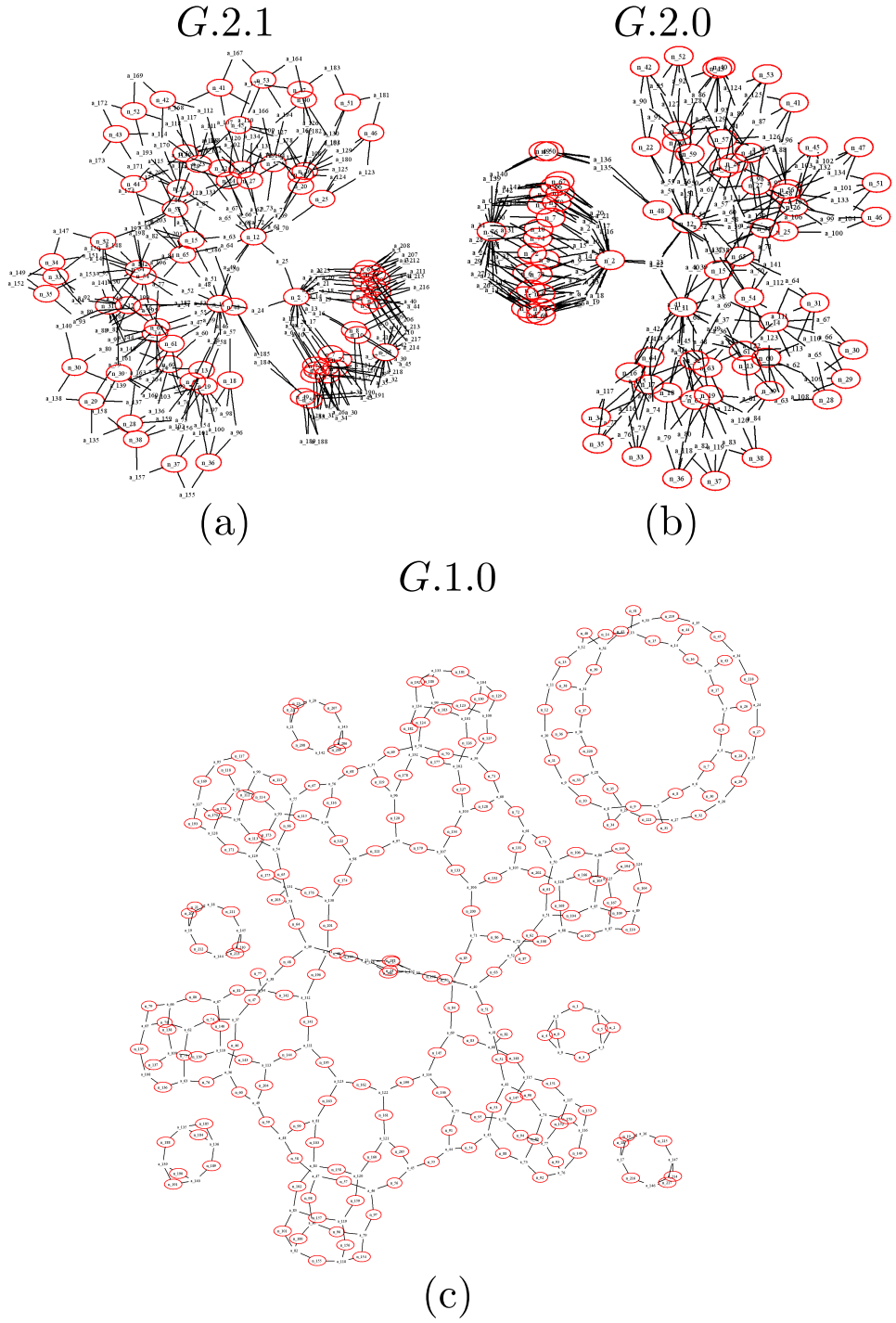


Figure 7.7: The initial hypergraph of the Figure 7.6.

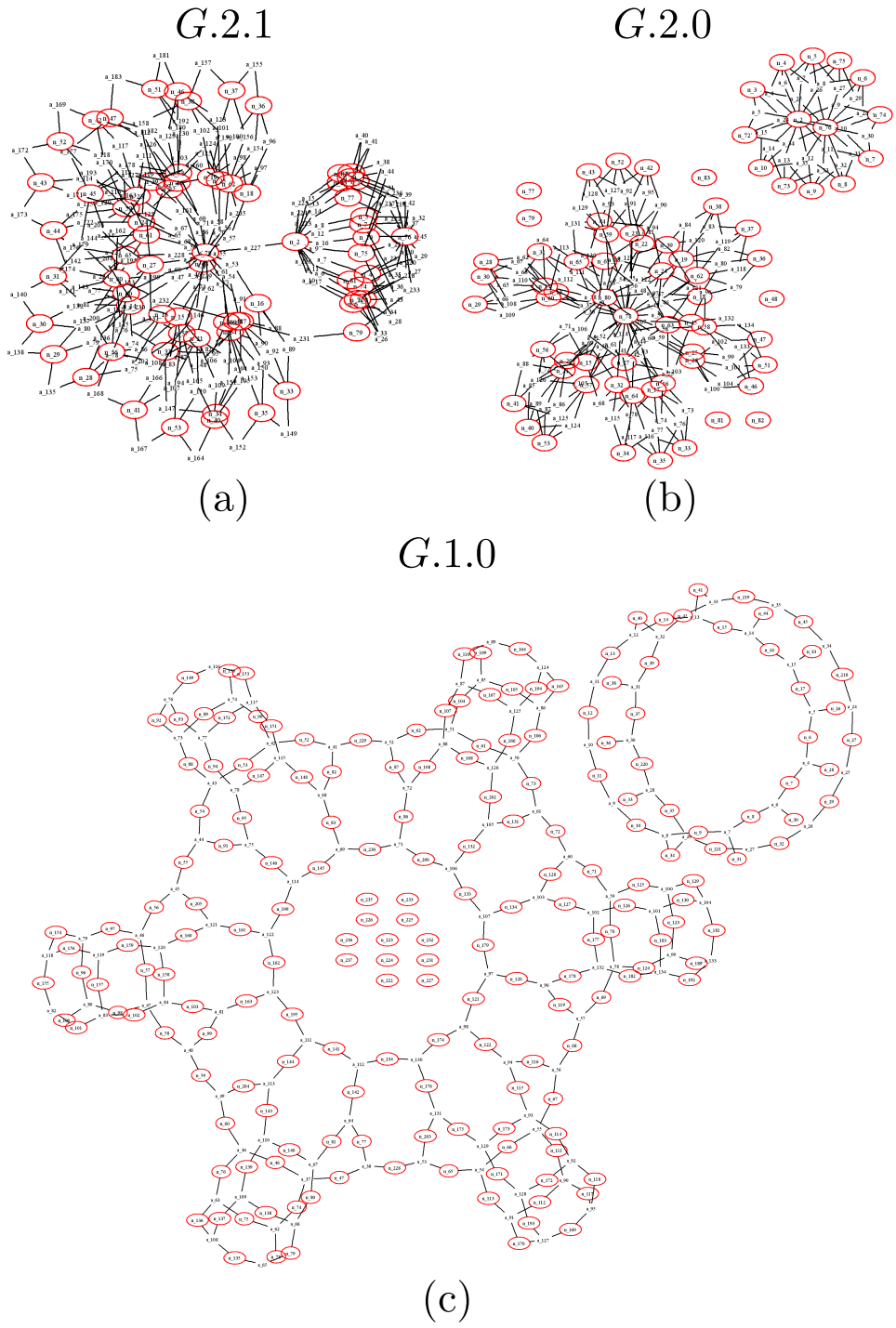


Figure 7.8: The hypergraph after maximal faces and maximal edges generation of the Figure 7.6.

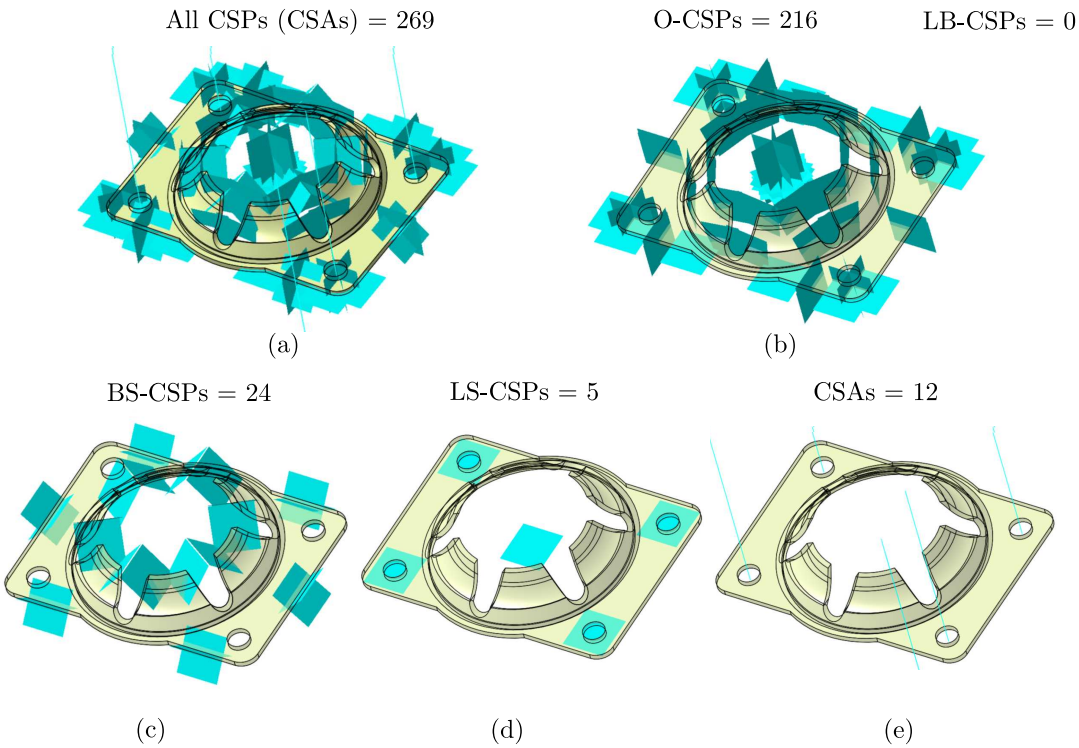


Figure 7.9: The set of CSPs of the example in Figure 7.6.

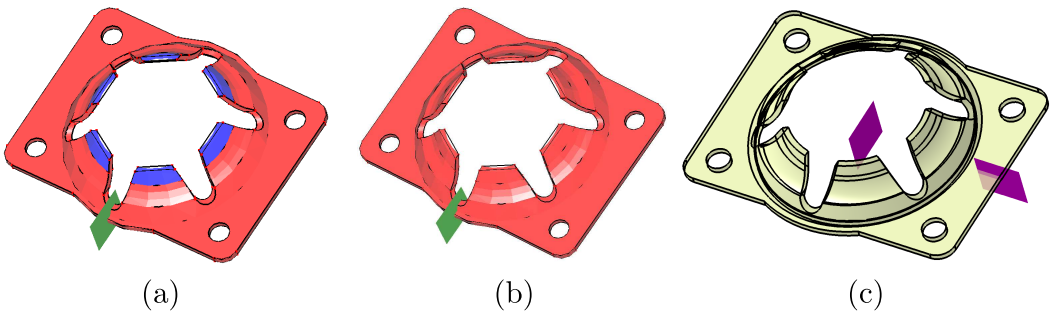


Figure 7.10: The propagation process and the symmetry detection results of the model in Figure 7.1. (a) is the first level propagation of one CSP, (b) is the result of the second level propagation, (c) illustrates the global symmetry planes.

7.3 Highlighting asymmetry and modification suggestions

During the propagation process, when it stops because an asymmetric configuration is encountered, some asymmetric faces, and possibly edges, are found. Figure 7.11-7.14 gives an example of asymmetric area related to a CSP that is highlighted. The axis should be the location of symmetry planes, obviously. But after the symmetry detection phase, there is no global symmetry plane nor global symmetry axis. Then, selecting interactively a CSP chain which is possibly a contribution to a global symmetry plane, the algorithm can highlight where the propagation stops (see Figure 7.13a). Figure 7.14a gives all asymmetric areas after merging the coinciding CSPs. Using this information, the user can find that the asymmetry comes from the teeth (see Figure 7.14b). After rotating the teeth slightly, two symmetry planes appear (see Figure 7.14c). If needed, the details at the bottom of the model can be removed and the model will have 24 symmetry planes.

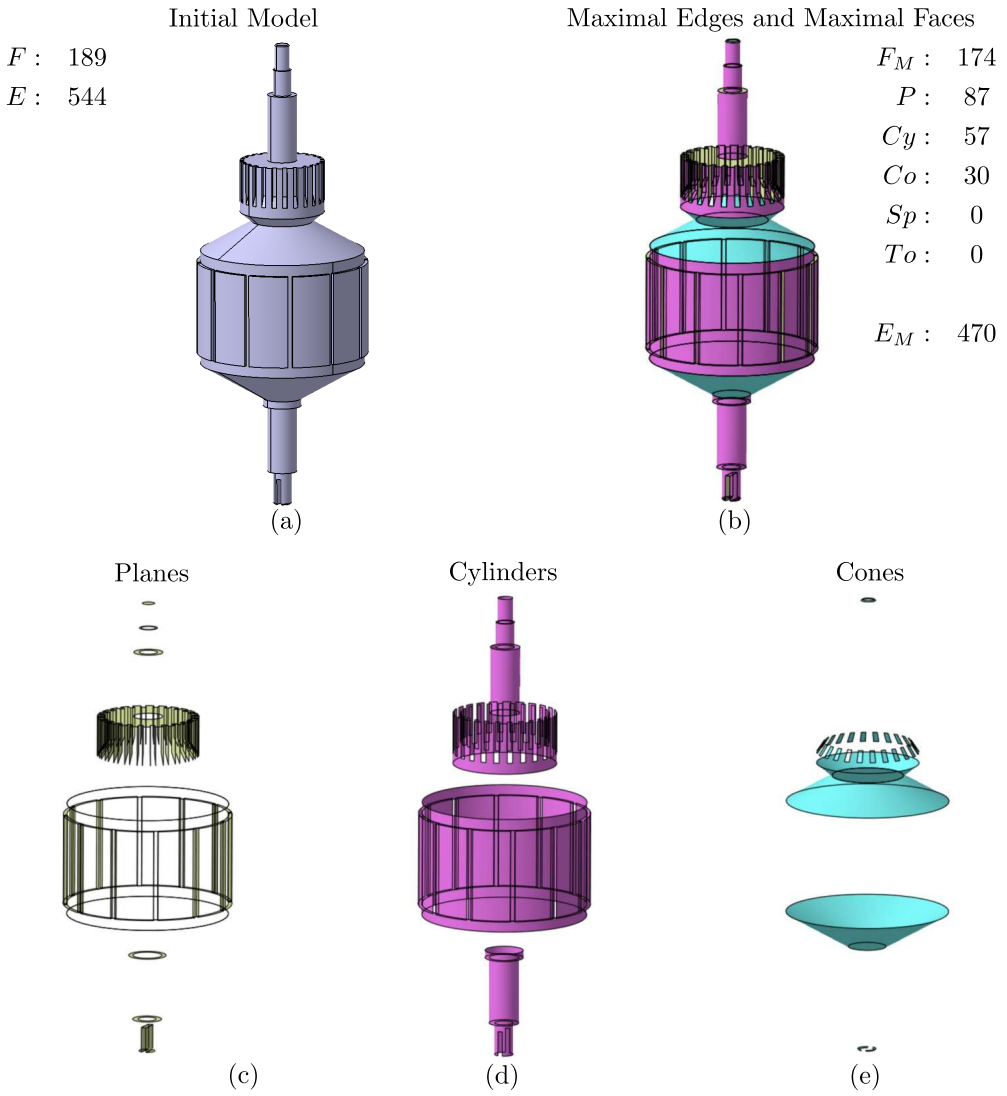


Figure 7.11: A reflective symmetry detection example applied to a rotor: (a) is the initial model; (b)- (e) the views of the different reference surfaces.

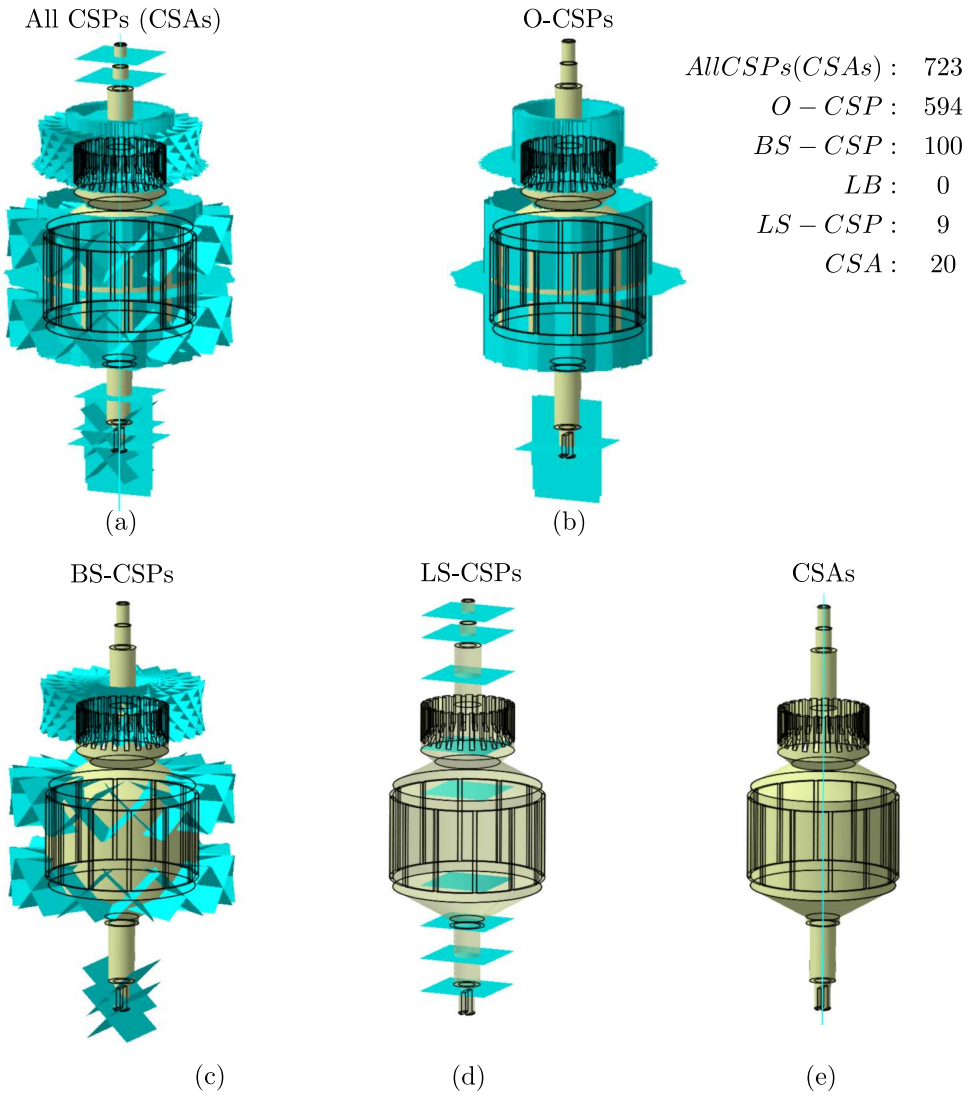
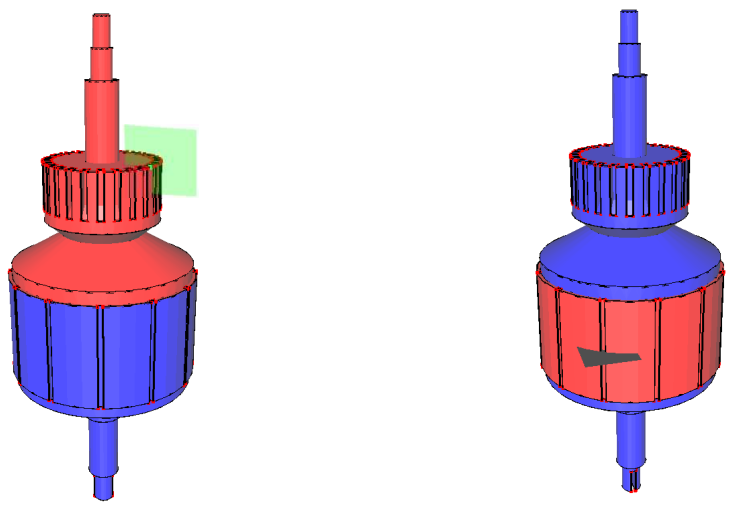


Figure 7.12: The set of CSPs of the example in Figure 7.11.



(a) (b)

Figure 7.13: The propagation process can't cover all faces. The red part is the symmetric one, the blue part represents the asymmetric one: (a) the location where the first level propagation process stops with a vertical CSP, (b) the location where the first level propagation stops with an horizontal CSP.

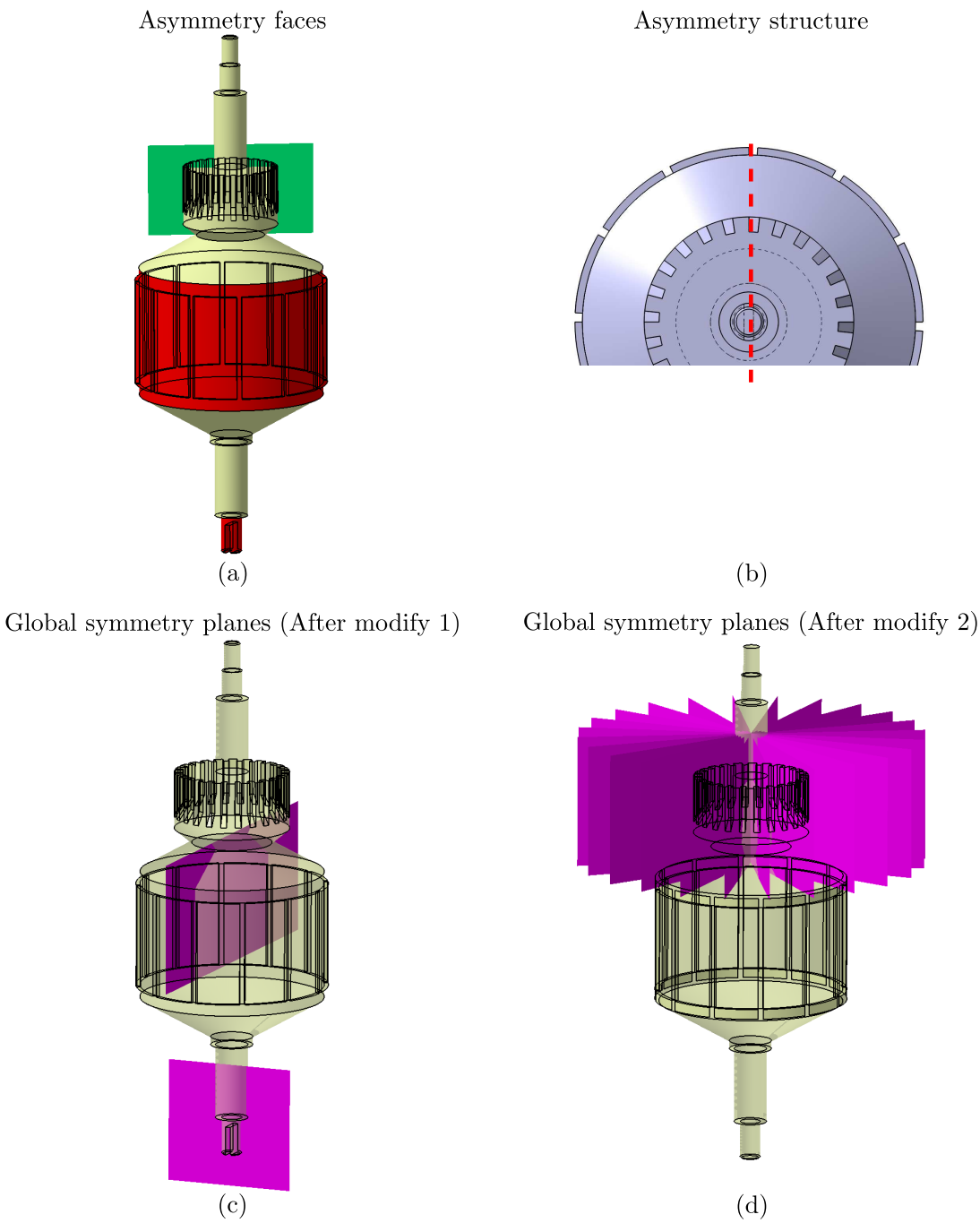


Figure 7.14: The reasons explaining the asymmetry and the results after modifications: (a) asymmetric area represented in red, (b) the tooth location is the asymmetry reason, (c) the global symmetry planes obtained after rotating the teeth slightly, (d) when ignoring the detail at the bottom, more symmetry planes are found.

7.4 Axisymmetry modification under suggestions derived from asymmetry identification

In this example, the initial model has only one symmetry plane. Figure 7.15a, b, c shows the initial B-Rep model, maximal boundary model and all CSPs. After the propagation process, one global symmetry plane is found (see Figure 7.15d). With respect to the CSA selected interactively in Figure 7.15e, the red surfaces are highlighted, because they are not axisymmetric.

Figure 7.16 gives the result of after ignoring some details on the model. There are more global symmetry planes cerated but the object is not axisymmetric yet. Finally, after removing the holes, the model is axisymmetric, which is shown in Figure 7.17.

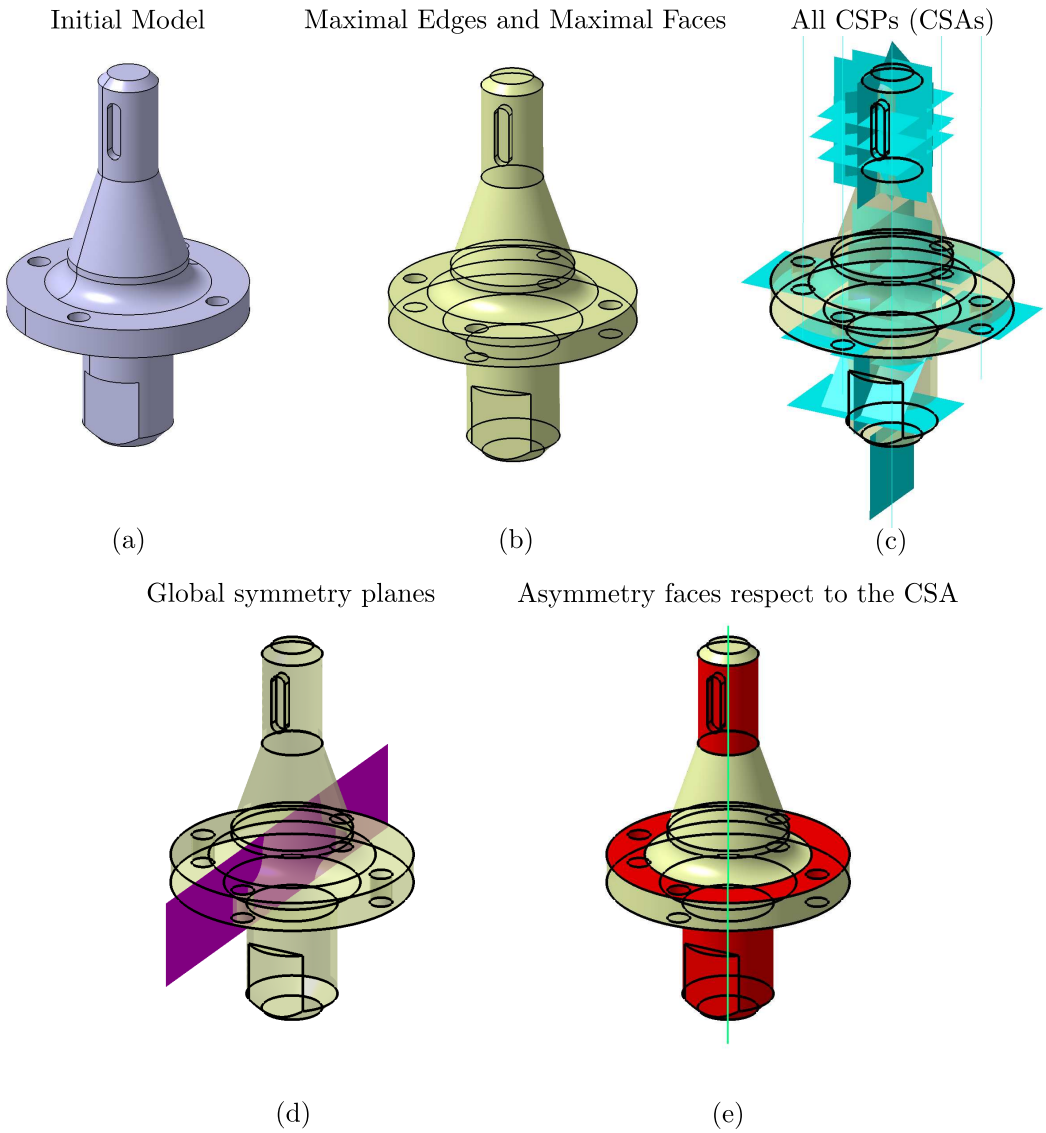


Figure 7.15: A reflective symmetry detection example: (a) is the initial model; (b) represents the maximal faces and maximal edges; (c) shows all the CSPs (CSAs); (d) is the result of the symmetry detection: one global symmetry plane.

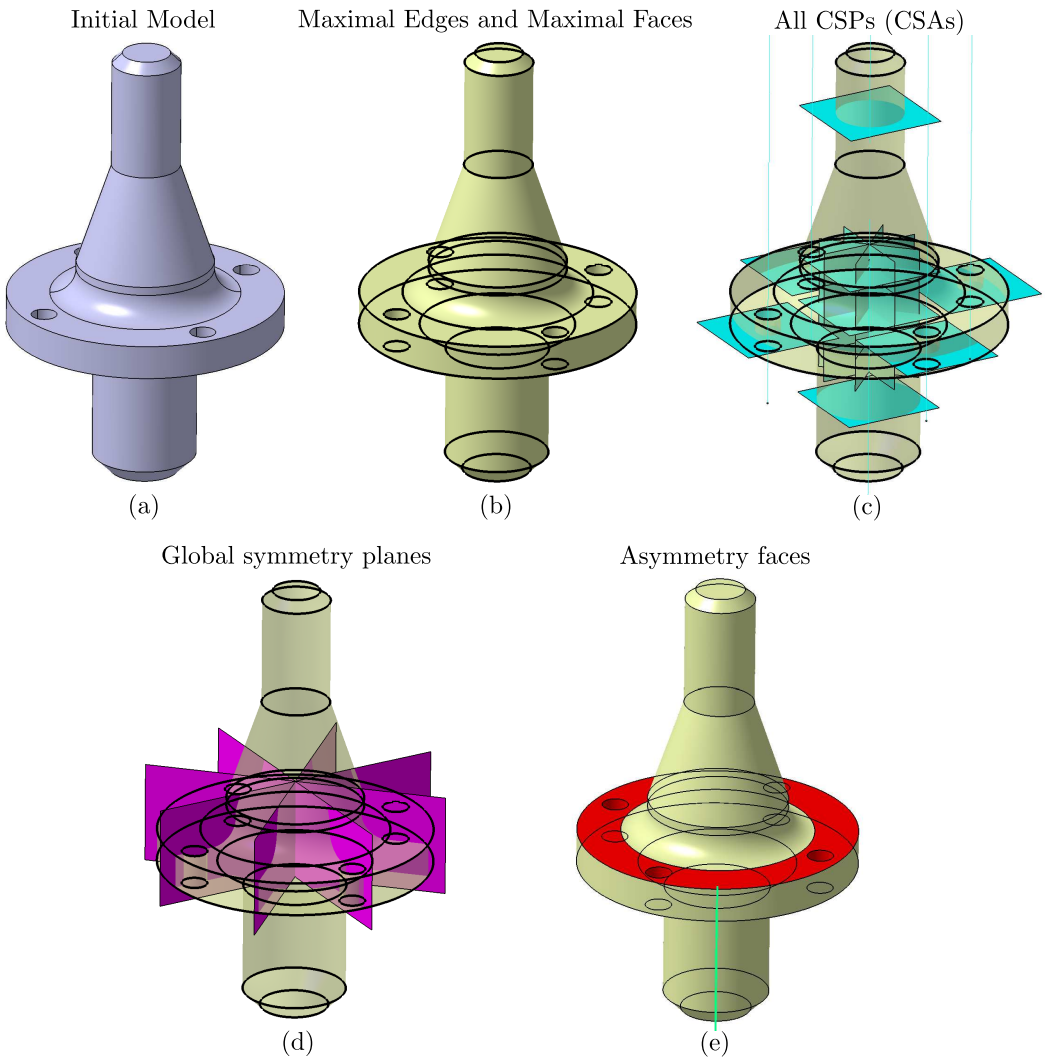


Figure 7.16: After a modification, the model in Figure 7.15 becomes more symmetric: (a) is the initial model; (b) represents the maximal faces and maximal edges; (c) shows all the CSPs (CSAs) generated; (d) is the result of the symmetry detection: there are more global symmetry planes; (e) represents the non axisymmetric faces with regard to the CSA.

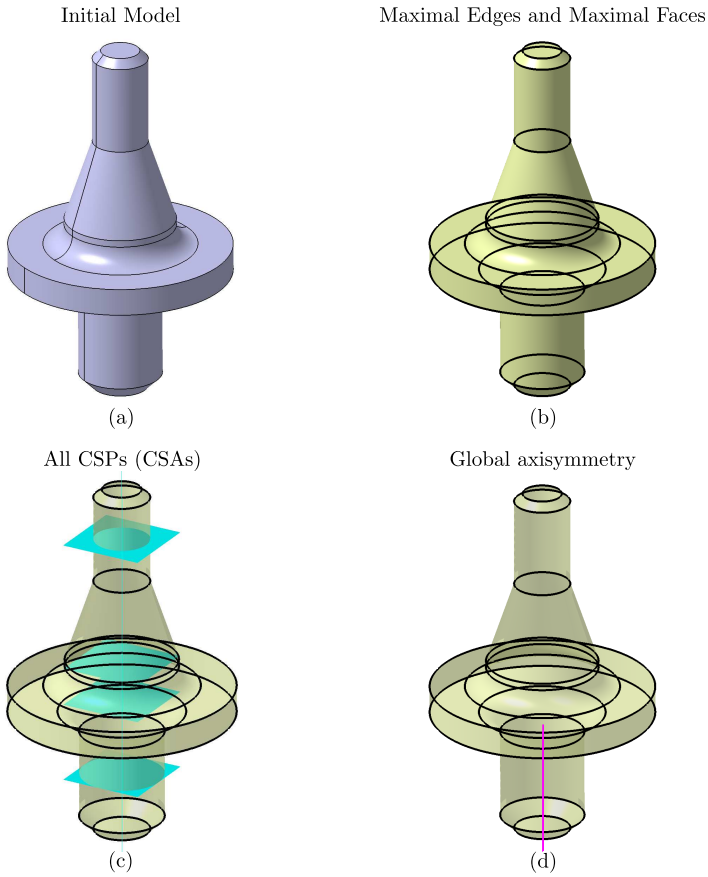


Figure 7.17: With further modifications with the removal of the through holes, the model in Figure 7.15 becomes axisymmetric: (a) is the initial model; (b) represents the maximal faces and maximal edges; (c) shows all the CSPs (CSAs) generated; (d) is the result of symmetry detection: the axisymmetry.

7.5 Conclusion

The models shown in this chapter prove the validation of the reflective symmetry analysis. The goals of detection of global reflective symmetry planes, global axisymmetry and asymmetric areas, are achieved. With the help of the asymmetry areas highlighted, a user can find the right symmetry planes and also know where to modify the input model to remove some asymmetry. The algorithm has been developed with variants in CATIA Macro execution environment under Windows OS and as an OpenCascade CAD application under Linux or Windows OS. The complexity for each major step has been discussed. With input models having hundreds of elements model, the time cost has a magnitude of a few tens of second. Compared to the other symmetry detection methods, this algorithm is fast and this makes applicable in an interactive context for various PDP applications. The examples proposed show that asymmetric areas can be useful to operate model transformations and they form a good basis to set up an algorithm using them assist the user in performing these modifications.

Chapter 8

Conclusions and perspectives

Conclusion

The present work has been motivated by the needs of improving PDPs because symmetries are widely used in many simulations processes. Shape analysis can help the user understand digital models more easily and can initiate suggestions for model modifications and shape simplifications. Indeed, reflective symmetry is a main property of many man made products and is ubiquitous in product digital models and, as examples, it can be used in model storage and retrieval, model simplification for FEA as well as assembly planning processes.

The thesis reports a reflective symmetry analysis algorithm. The algorithm is valid on B-Rep CAD models which are two manifolds objects whose boundary is obtained through the combination of five reference surfaces: planes, cylinders, cones, spheres and tori. The symmetry analysis process is of type divide-and-conquer. In order to stay consistent with the symmetry properties extracted, the B-Rep model input is divided into maximal surfaces, maximal edges and vertices. As a contribution to encode their adjacency relationships, these three categories of entities are structured with hypergraph data structures and connected to the surface intrinsic parameters. Hypergraphs form a topological description of the object boundary and stay connected to its B-Rep topological datastructure, which is kept unchanged. These are the preliminaries of the proposed symmetry analysis process and, during this preliminary phase, all the boundary singularities are processed. Then, the five reference surfaces form the initial infinite point sets and, with their associated symmetry properties, they can initiate the divide phase. The combinations of couples of these surfaces provide the symmetry constraints to generate the CSPs forming the target of the divide phase. These CSPs are attached to the edges, vertices and faces of the object boundary. Five categories of CSPs: O-CSP, BS-CSP, LB-CSP, LS-CSP, SS-CSP and a CSA have been defined, which contain all the local symmetry possibilities of the object. Using propagation processes, which describe the conquer phase, local symmetry properties are expanded through the adjacency relationships encoded by the hypergraphs. The conquer phase subdivides into two complementary propagation processes. The first one, merges coincident CSPs/CSAs together to form CSP chains representing the intersections between symmetry planes and the object boundary. Then, the second one carries on the expansion of the symmetric areas on both sides of these CSP chains,

forming SS-CSPs, until all surfaces are covered without the occurrence of asymmetry. This leads to the desired GSPs and axisymmetry. This divide and conquer approach does not only provide global symmetry properties but also the extent of CSPs/CSAs over the object boundary.

Compared to other symmetry detection methods, the divide-and-conquer method proposed has the following advantages:

- A B-Rep CAD model forms input model and is addressed as an infinitive point set. Compared to the finite point sets representations which can only give approximate symmetry planes, the CAD model covers all the details and singularities of the corresponding shape. Because the surface locations and their intrinsic parameters are the only geometric information processed, the tolerance of the symmetry detection can be set to that of the geometric modeler. The symmetry properties obtained are precise results because all the singularities of the object boundary have been encoded in the hypergraph datastructures. Choosing the STEP format as reference data structure, which is an ISO file format, the algorithm can be widely used during a PDP;

- The hypergraphs datastructure provides the object boundary with a topological representation intrinsic to the object shape and consistent with the symmetry properties addressed. Compared to classical B-Rep datastructures, loop edges with one or no vertex is a major difference. Consequently, hypergraphs support well the major processes of this symmetry analysis algorithm starting from the maximal faces and edges generation (the core infinite point sets), the CSPs collection and propagation processes;

- Maximal faces and maximal edges representation is needed that form the core point sets of the divide and conquer process. The input B-Rep model often breaks surfaces into pieces due to the modeling process, topological requirements, etc. and also the symmetry properties. This limits other methods based on classical B-Rep model datastructures. Referring to maximal faces and edges is independent from the modeling process, the surface parameterization, embedding and singularities. This preparation phase ensures that all the CSPs attached to the surfaces can reflect the inherent symmetry properties of the object. Tangent configurations in the B-Rep model form singularities that can be processed with the help of the hypergraphs. So, the regular and non-regular points are defined to support the face merging process, which can contribute to the consistent representation of the object boundary;

- The CSPs may not be only included in the global symmetry planes, but also contribute to partial symmetry properties. CSPs are attached to the vertices, edges or faces of the object boundary and they cover all the meaningful rela-

tionships between these entities. Consequently, it is a local symmetry property with respect to its original entities.

Even though there is a combinatorial complexity appearing in the CSP generation process with regard to the categories of reference surfaces, a closer look is needed there. Related to infinite point sets, curvature distribution, singularities of the embedding, behavior at infinity, topological features of the corresponding surfaces are among the core mathematical issues that can be faced. In addition, combining surfaces exhibiting at least one the previous features can hardly avoid some combinatorial processing. Now, it happens that the reference surfaces addressed cover a large spectrum of the above features. The above features are hardly taken into account in many approaches. Discrete approaches [44] with discrete curvatures are note able to distinguish ombilic points from the apex of a cone as one among other observations related to these approaches.

Indeed, the five reference surfaces combine open surfaces (plane, cylinder and cone) and closed ones (sphere and torus), surfaces with null curvature (plane) and constant one (sphere) without unique directions of curvature, surface with singularity (cone), which creates specific treatments anyway.

Coming back to the CSPs, each one represents a basic symmetry property. After the conquer phase, these partial symmetry planes cover an area as large as possible until asymmetry occurs. This propagation process produces the symmetry analysis of the object. So, after the detection of the global symmetry properties, all partial symmetry properties are also, which is a good basis for shape analysis and is obtained at a low complexity. Few other approaches can provide equivalent information;

- Referring to parametric surfaces makes the symmetry detection process works on a precise and global basis. Tate's [60][58] uses such an approach. Her method compares surface loops using some global geometric properties like area, etc., which don't exist in the input model and need to be computed through accuracy thresholds. Surface intrinsic parameters are available and stored in the B-Rep model input. The algorithm presented here only reads these and use them to identify the symmetry properties. Once the parameters of two surfaces characterize a symmetric configuration, the infinite points formed by the corresponding surfaces are symmetric. The symmetric layout of surfaces propagates to their intersection curves without requiring a complex analysis of the trimming curves defining a B-Rep CAD model. All these features significantly improves the robustness of the approach compared to Tate's one;

- Axisymmetry is accessible. Axisymmetry is a configuration with an infinite number of symmetry planes. With the finite points sets representations and their approximate results, a precise axisymmetry cannot be generated. The proposed approach incorporates symmetry axes right from the divide phase and

the global approach set up with infinite points sets helps extracting precisely and easily the axisymmetry properties at all levels of the divide-and-conquer process;

- The time cost of the algorithm is low. Using surfaces makes the symmetry detection operates using infinite points set without any discretization. Even the edges of the B-Rep model are not approximated but processed globally through the parameters of their adjacent surfaces. As a complement, hypergraphs provide all the consistent adjacency relationships. The preparation phase and divide-and-conquer method make the algorithm fairly less combinatorial than the current approaches with a nearly entirely linear complexity and keep the whole process at low cost.

The examples illustrated in this thesis demonstrate the feasibility of our approach. The CATIA macros developed prove that the approach is valid for industrial CAD software and can be readily incorporated as a new function.

Perspectives

Regarding further research, the divide and conquer symmetry analysis has some more potentialities, among which:

- The extension to multiple volumes and assembly models. At present, the input model is only one volume. Symmetry properties exist not only for a single component, but also for groups of objects, including assembly models. In fact, the second level propagation process does not require, with the SS-CSPs, that the left and right faces belong to same volume. If a CSP attached to two volumes is generated, the propagation process can be applied over these two volumes. To implement the algorithm on assembly models, a topological relationship between its volumes needs to be created;

- To include new categories of reference surfaces. Currently, the five categories of reference surfaces is a strict constraint of the algorithm and real engineering components cannot be covered entirely. In prior research, surfaces of revolution, ruled surfaces, blends and free-form surfaces are categories that could be added into our approach. Especially NURBS curves and surfaces that can be used arbitrary surface generation;

- To detect rotational symmetries. Currently, the approach can detect reflective symmetries only. Rotational symmetry is also an important shape property. The criteria used in our approach is purely focusing on reflective symmetry. There is a need to define the concept of Candidate Rotational Symmetry Axis (CRSA) at

a similar level as the CSPs, CSAs so that it can take part to the divide phase and incorporated in the propagation processes. Another question is how to extract these CRSAs from M_{MAX} and check that the concept of maximal faces and edges is adequate with respect to this concept;

- Non-manifold model processing. Non-manifold models are also useful for geometric modeling and PDPs. For example, many simulation models for FEA are simplified as planes, segments, etc. The hypergraphs form a powerful tool which can represent not only manifold models, but also non-manifold ones. So the objective is to adapt the current approach to non-manifold models where the edge/face adjacency relationship can vary from one edge to the other inside the input model. A global approach in this context is indeed a challenge;

- Valid for mesh models. Mesh models are a special category of B-Reps, which has a wide range of application in PDPs. A mesh generation process breaks the surface representation into many pieces forming the mesh cells. In some cases, a mesh model is macroscopically symmetric, but it is not true when comparing each cell. The current divide-and-conquer symmetry analysis algorithm is not producing the desired in this case., So whether to use tolerance control or rebuilding some initial surfaces, there are some issues to be solved to process efficiently mesh models.

Appendix A

Combinatorial analysis of the combination of reference surfaces producing CSPs

Plane/Sphere (P, Sp)**:** Γ_{P-Sp} reduces to circles and defines E. All configurations are axisymmetric with O_s the center of Sp such that $O_s \subset \Pi$ and $\Pi \perp P$.

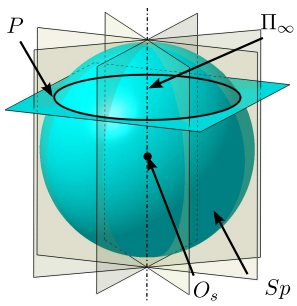


Figure A.1: Intersection between a plane P and a sphere Sp.

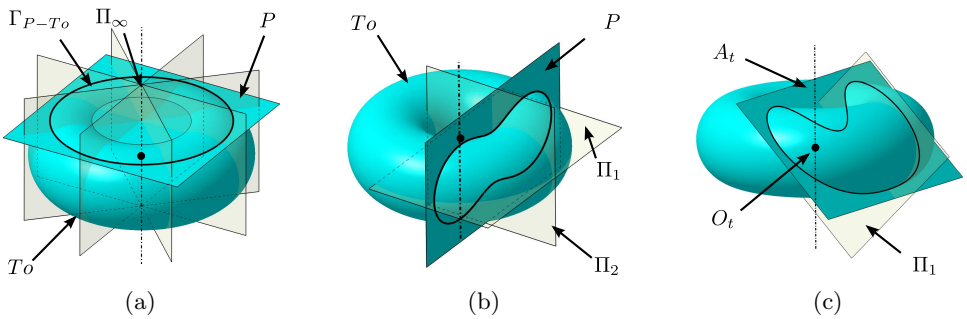


Figure A.2: (a) P orthogonal to A_t, (b) P is parallel to A_t, (c) P is neither orthogonal nor parallel to A_t.

Plane/Torus (P, To)**:** Symmetry properties of To are similar to those of Cy (see Figure A.2) and summarized in Table A.1. A difference holds for orthogonal symmetry plane Π_1 that necessarily contains A_t, the torus axis. If P has an arbitrary

(F, F_a)	(P, To)	(P, To)	(P, To)
Geometric constraint	$P \perp A_t$	$P \parallel A_t$	P not $\perp A_t$, P not $\parallel A_t$
Π	∞ axisymmetry	Π_1: $A_t \subset \Pi_1$	Π_1: $A_t \subset \Pi_1$
		Π_2: $A_t \perp \Pi_2$, $O_t \subset \Pi_2$	

Table A.1: Configurations of symmetry planes for P and To.

orientation (see Figure A.2c) with respect to To, the intersection curve can reduce to a curve with singularity when P has a point tangent to To and here, Π_1 contains this singularity. In this configuration, Π_1 is no longer an O-CSP but the *two loops* forming this configuration produce two distinct maximal edges. These maximal edges end up with one vertex only since these loop edges characterize a crossing configuration around the vertex, hence Π_1 is not lost but appears as an LB-CSP.

Cylinder/Cylinder $(Cy1, Cy2)$**:** Based on the symmetry properties of Cy_i alone, the space location of symmetry planes reduces to three configurations (see Figure A.3, Table A.2):

- Cy_1 and Cy_2 are orthogonal. In Figure A.3a and d, the maximal edge E changes from one component to two disconnected ones. In Figure A.3a, Π_1 and Π_2 are both O-CSPs whereas in d, Π_2 is no longer an O-CSP. Similarly to $(P,\ Cy)$, it is not lost but appears as an LS-CSP and evolves in an LB-CSP if Cy_1 and Cy_2 are tangent to each other. In addition, when A_{c1} and A_{c2} intersect (see Figure A.3e), another plane Π_3 appears. That plane originates from the two components defining the intersection with E_1 and E_2. It is not mentioned in Table A.2 since it never belongs to the O-CSP category, rather it is an LS-CSP evolving into a BS-CSP when Cy_1 and Cy_2 have the same diameter (see section 5.6);

- Cy_1 and Cy_2 are parallel to each other (see Figure A.3c). The resulting intersection is similar to the $(P,\ Cy)$ where Cy is parallel to P, which explains the reduction of the intersection to one straight line only;

- Cy_1 and Cy_2 are neither parallel nor orthogonal to each other but A_{c1} and A_{c2} intersect (see Figure A.3b). Only one symmetry plane exists, Π_1. The limiting configuration where Cy_1 and Cy_2 have the same radius, has no influence over the status of Π_1.

Cylinder/Cone (Cy, Co)**:** Between Cy and Co, symmetry properties mainly rely on their axes A_{cy}, A_{co} relative locations (see Figure A.4, Table A.3). Intersection configurations producing either one or two disconnected components do not generate any particular symmetry plane.

(F, F_a)	(Cy_1, Cy_2)	(Cy_1, Cy_2)	(Cy_1, Cy_2)
Geometric constraint	$A_{c1} \perp A_{c2}$,	$A_{c1} \cap A_{c2} = pt$,	$A_{c1} \parallel A_{c2}$
Π	Π_1: $A_{c1} \subset \Pi_1$, $A_{c2} \perp \Pi_1$	Π_1: $A_{c1}, A_{c2} \subset \Pi_1$	Π_1: $A_{c1}, A_{c2} \perp \Pi_1$
	Π_2: $A_{c1} \subset \Pi_2$, $A_{c2} \perp \Pi_2$,		

Table A.2: Configurations of symmetry planes for Cy_1 and Cy_2.

Configurations where Co becomes tangent to Cy do not alter the status of the symmetry planes nor their number (see Figure A.4a, b, c). Coinciding axes produce an axisymmetric configuration (see Figure A.4d).

(F, F_a)	(Cy, Co)	(Cy, Co)
Geometric constraint	$A_{cy} \perp A_{co}$, Not co-planar	$A_{cy} \cap A_{co} = pt$
Π	Π_1: $A_{co} \subset \Pi$	Π_1: $A_{cy}, A_{co} \subset \Pi_1$,

(F, F_a)	(Cy, Co)	(Cy, Co)
Geometric constraint	$A_{cy} \cap A_{co} = pt$, $A_{cy} \perp A_{co}$	$A_{cy} = A_{co}$
Π	Π_1: $A_{cy}, A_{co} \subset \Pi_1$	∞: axisymmetry
	Π_2: $A_{co} \subset \Pi_2$, $A_{cy} \perp \Pi_2$	

Table A.3: Four configurations of symmetry planes for Cy and Co, first subset.

Cylinder/Sphere (Cy, Sp)**:** The relative locations of Cy and Sp end up with two configurations (see Figure A.5a, b and Table A.4) whether the center O_s lies on Cy axis A_c or not. Having O_s on A_c produces axisymmetry as well as an intersection with two disconnected components and a symmetry plane of type LS-CSP, which is not part of the current analysis.

(F, F_{aij})	(Cy, Sp)	(Cy, Sp)
Geometric constraint	$O_s \neq A_c$	$O_s \subset A_c$
Π	Π_1: $A_c \subset \Pi_1, O_s \subset \Pi_1$,	∞: axisymmetry
	Π_2: $A_c \perp \Pi_2$, $O_s \subset \Pi_2$	

Table A.4: Two configurations of symmetry planes for (Cy, Sp).

Cylinder/Torus (Cy, To)**:** Depending on the location of Cy axis, A_c, with respect to the axis A_t and center O_t of To, five categories of symmetry plane configurations emerge (see Figure A.6, Table A.5):

- A_t is parallel to A_c (see Figure A.6b). If the intersection curve contains only one component, Π_1 is an O-CSP. However, if To is tangent to Cy Π_1 changes

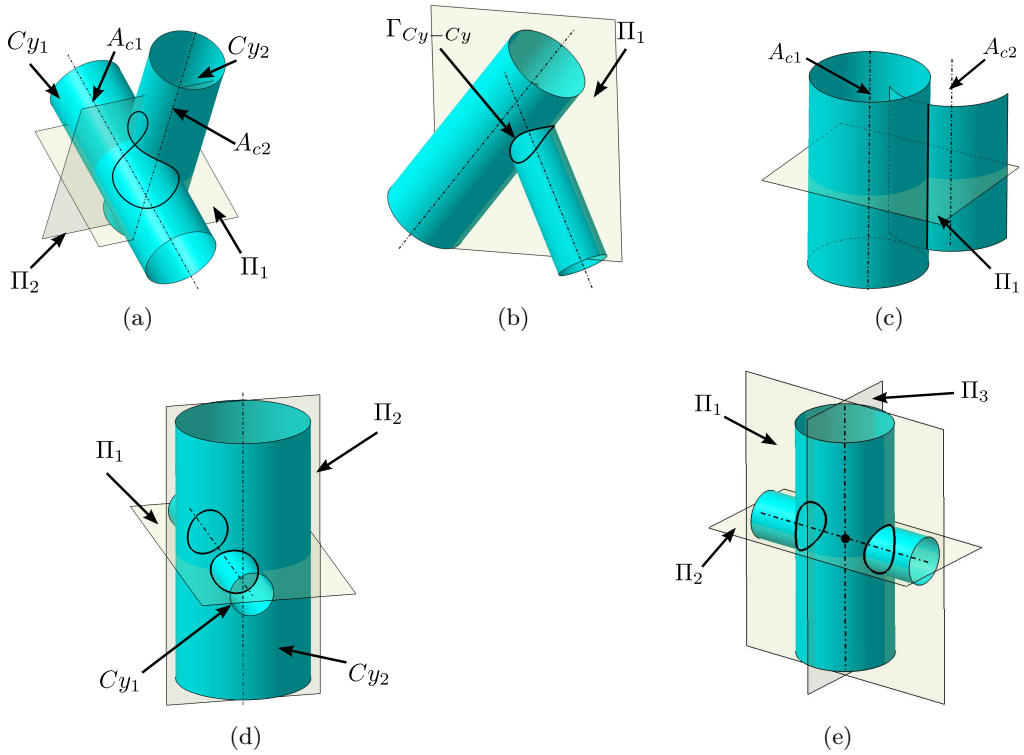


Figure A.3: Categories of symmetry plane distribution between intersecting cylinders Cy_1 and Cy_2.

category to LB-CSP. Now, if their relative position generates an intersection with two disconnected components, Π_1 still exists but as an LS-CSP;

- A_t is orthogonal to A_c (see Figure A.6c) and they are crossing each other. The same analysis process as above applies now with Π_2;
- A_t is orthogonal to A_c (see Figure A.6d) but they are not crossing each other. The same analysis process as above applies with Π_2;
- A_t is neither orthogonal nor parallel to A_c but A_t intersects with A_c (see Figure A.6a). Π_1 is the only O-CSP when the intersection reduces to one maximal edge E. Π_1 evolves into an LB-CSP when To is tangent to Cy or to an LS-CSP when their intersection produces two disconnected components;
- A_t coincides with A_c and produces an axisymmetric configuration (see Figure A.6e).

Cone/Cone ($Co1, Co2$)**:** Symmetry properties of Co generate two configurations close to that of Cy and Sp. If the two axes A_{c1}, A_{c2} coincide, Γ_{Co-Co} is a circle, hence

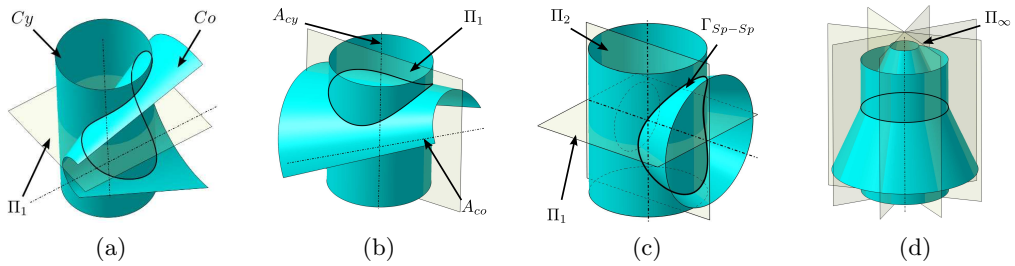


Figure A.4: Symmetry properties between a cylinder and a cone.

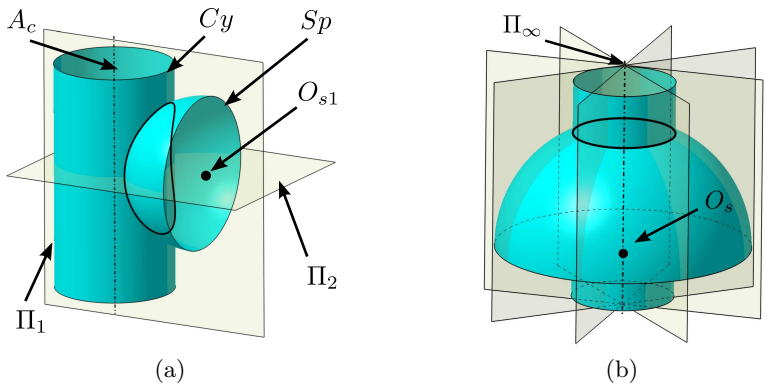


Figure A.5: Symmetry properties between (Cy, Sp).

the axisymmetry (see Figure A.7b). The other configuration produces a symmetry plane Π_1 when A_{c1} and A_{c2} intersect. The O-CSP thus obtained is stable under Coi dimensional parameter variations whether the intersection between $Co1$ and $Co2$ produces one loop or two or a singular configuration where the loops touch each other.

Cone/Sphere (Co, Sp)**:** It is also close to the (Cy, Sp) configurations with the difference that $A_c \neq O_s$ generates no more than one symmetry plane Π_1 (see Figure A.8 and Table A.7).

Cone/Torus (Co, To)**:** The different axes locations synthesize with five categories of symmetry planes (see Table A.8 and Figure A.9), rather close to Cy and To ones:

- A_t is parallel to A_c (see Figure A.9b). If the intersection curve contains only one component, Π_1 is an O-CSP. However, if To is tangent to Co, Π_1 changes category to LB-CSP. Now, if their relative position generates an intersection with two disconnected components, Π_1 still exists but as an LS-CSP;
- A_t is orthogonal to A_c (see Figure A.9c) and they are crossing each other. The same analysis process as above applies with Π_1. Whatever the relative position of Co with respect to To, Π_2 stays an O-CSP;

(F, F_a)	(Cy, To)	(Cy, To)	(Cy, To)
Geometric constraint	$A_c \cap A_t = pt$	$A_c \parallel A_t$	$A_c \perp A_t$, $A_c \cap A_t = pt$
Π	Π_1: $(A_c, A_t) \subset \Pi_1$	Π_1: $(A_c, A_t) \subset \Pi_1$	Π_1: $A_c \perp \Pi_1$, $A_t \subset \Pi_1$
		Π_2: $O_t \subset \Pi_2$, $(A_c, A_t) \perp \Pi_2$	Π_2: $A_t \perp \Pi_2$, $A_c \subset \Pi_2$

(F, F_a)	(Cy, To)	(Cy, To)
Geometric constraint	$A_c \perp A_t$, $A_c \cap A_t = \phi$	$A_c = A_t$
Π	Π_1: $A_t \perp \Pi_1$, $A_c \subset \Pi_1$	∞: axisymmetry
	Π_2: $A_c \perp \Pi_2$, $A_t \subset \Pi_2$	

Table A.5: Five configurations of symmetry planes for Cy and To.

(F, F_a)	(Co, Co)	(Co, Co)
Geometric constraint	$A_{c1} \cap A_{c2} = pt$	$A_{c1} = A_{c2}$
Π	Π_1: $A_{c1}, A_{c2} \subset \Pi_1$	∞: axisymmetry

Table A.6: Two configurations of symmetry planes for $(Co1, Co2)$.

- A_t is orthogonal to A_c (see Figure A.9d) but they are not crossing each other. The same analysis process as above applies with Π_1;
- A_t is neither orthogonal nor parallel to A_c but A_t intersects with A_c (see Figure A.9a). Π_1 is the only O-CSP when the intersection reduces to one maximal edge E. Π_1 evolves into an LB-CSP when To is tangent to Co or an LS-CSP when their intersection produces two disconnected components;
- A_t coincides with A_c and produces an axisymmetric configuration (see Figure A.9e).

Sphere/Sphere $(Sp1, Sp2)$**:** It reduces to only one axisymmetric configuration whose axis is $O_{s1}O_{s2}$ (see Figure A.10 and Table A.9).

Sphere/Torus (Sp, To)**:** Three categories appear (see Figure A.11 and Table A.10) among which the coincidence of O_s with A_t produces the axisymmetry configuration (see Figure A.11c).

Whatever the relative position of O_s with respect to A_t, there exists a symmetry plane Π_1 containing these entities (see Figure A.11a). When the intersection curve generates only one maximal edge E, Π_1 is an O-CSP, which transforms into an LB-CSP or LS-CSP when two loops appear progressively without missing any symmetry plane. If A_t is orthogonal to O_tO_s, two symmetry planes appear where Π_1 plays

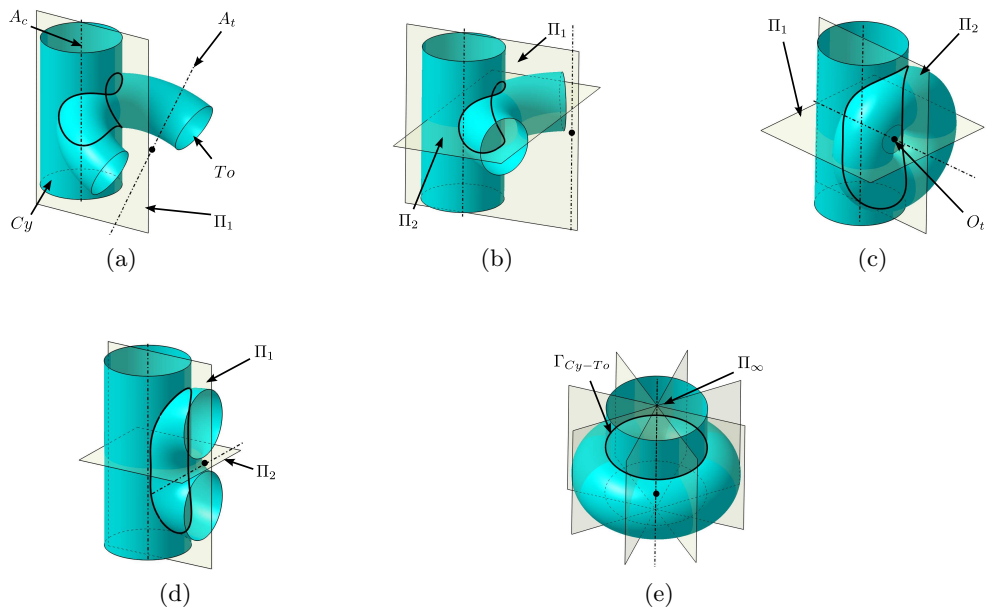


Figure A.6: Symmetry properties between Cy and To.

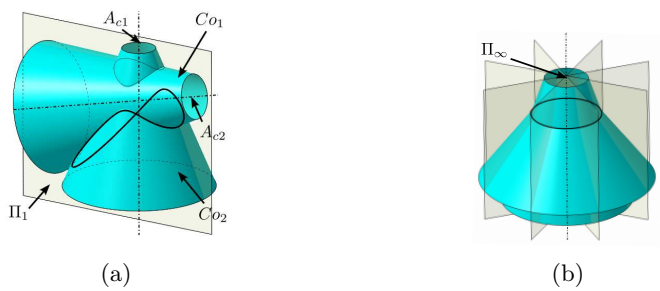
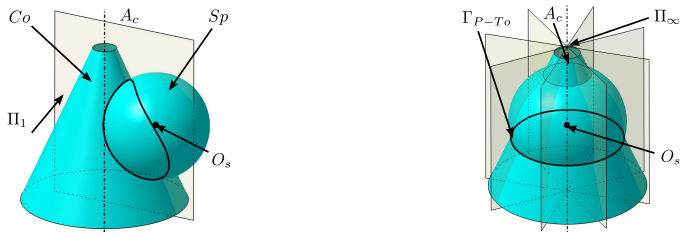


Figure A.7: Symmetry properties between $Co1$ and Co_2.

the same role as previously and Π_2 stays an O-CSP under the transformation of the intersection curve.

Torus/Torus $(To1, To2)$**:** $(To1, To2)$ ends up with six configurations (see also Table A.11):

- In Figure A.12a, b and c, the axes A_{t1} and A_{t2} are co-planar, which forms a sub-category containing two configurations. Whether A_{t1} and A_{t2} intersect (see Figure A.12a) or are parallel to each other with centers $O_{t1}O_{t2}$ not orthogonal to A_{ti} (see Figure A.12b), the symmetry plane Π_1 thus produced behaves similarly. Π_1 defines an O-CSP when there is a single intersection curve E, otherwise it is not lost but evolves toward an LS-CSP through an LB-CSP. Figure A.12c depicts the configuration where $O_{t1}O_{t2}$ is orthogonal to A_{ti}. This configuration



(a) (b)

Figure A.8: Symmetry properties between Co and Sp.

(F, F_a)	(Co, Sp)	(Co, Sp)
Geometric constraint	$O_s \neq A_c$	$O_s \subset A_c$
Π	Π_1: $A_c \subset \Pi_1, O_s \subset \Pi_1$,	∞: axisymmetry

Table A.7: Two configurations of symmetry planes for (Co, Sp).

adds Π_2, which is a stable O-CSP;

- Figure A.12d illustrates a configuration where A_{t1} and A_{t2} are orthogonal to each other, with $O_{t1}O_{t2}$ orthogonal to either A_{t1} or A_{t2} and $||O_{t1}O_{t2}|| > 0$. The resulting symmetry plane Π_1 behaves identically to the previous configurations;
- In addition to the previous constraints, when $O_{t1}O_{t2}$ is orthogonal to A_{t1} and A_{t2} both and $||O_{t1}O_{t2}|| > 0$, Π_2 emerges (see Figure A.12e). Likewise Π_2 in the configuration where A_{t1} and A_{t2} are parallel to each other, Π_2 is a stable O-CSP;
- When A_{t1} coincides with A_{t2}, the configuration becomes axisymmetric (see Figure A.12f);
- Finally, A_{t1} is orthogonal to A_{t2} and O_{t1} coincides with O_{t2} (see Figure A.12g). This configuration necessarily produces two curves or four ones and E is one of them, which means that Π_1 is necessarily an LS-CSP, i.e. one of the three symmetry planes appearing in this configuration belongs to the LS-CSP category. Then, Π_2 and Π_3 are O-CSPs but either Π_2 or Π_3 can evolve toward the LS-CSP category while the other plane is stable in the O-CSP category.

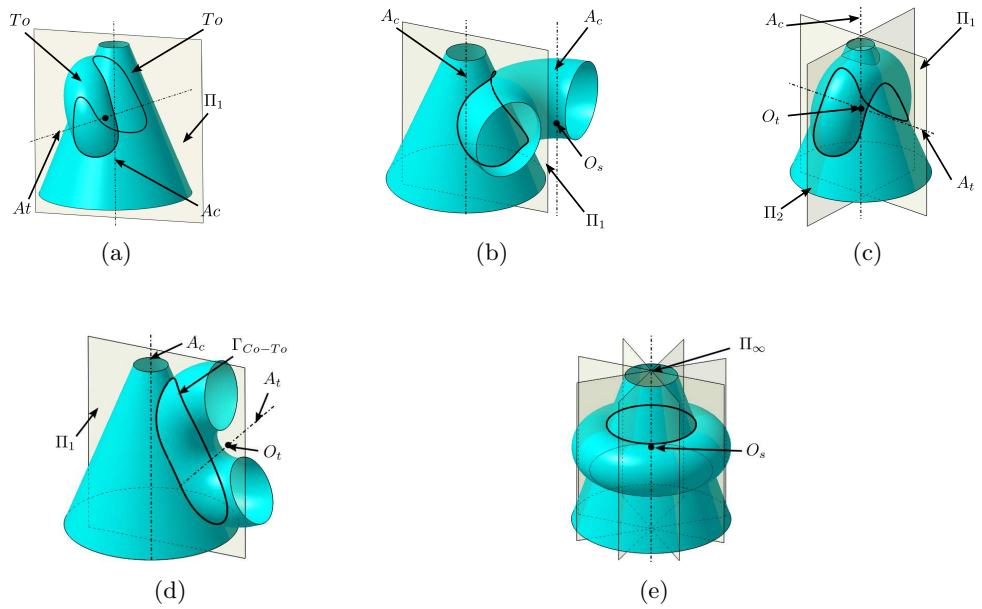


Figure A.9: Symmetry properties between Co andTo.

(F, F_a)	(Co, To)	(Co, To)	(Co, To)
Geometric constraint	$A_c \cap A_t = pt$	$A_c \parallel A_t$	$A_c \perp A_t$, $A_c \cap A_t = pt$
Π	Π_1: $(A_c, A_t) \subset \Pi_1$	Π_1: $(A_c, A_t) \subset \Pi_1$	Π_1: $A_c \subset \Pi_1$, $A_t \subset \Pi_1$
			Π_2: $A_t \perp \Pi_2$, $A_c \subset \Pi_2$

(F, F_a)	(Co, To)	(Co, To)
Geometric constraint	$A_c \perp A_t$, $A_c \cap A_t = \phi$	$A_c = A_t$
Π	Π_1: $A_t \perp \Pi_1$, $A_c \subset \Pi_1$	∞: axisymmetry

Table A.8: Five configurations of symmetry planes for Co and To.

(F, F_a)	(Sp, Sp)
Geometric constraint	$\forall$ (O_{s1}, O_{s2})
Π	∞: axisymmetry

Table A.9: Configurations of symmetry planes for Sp and Sp.

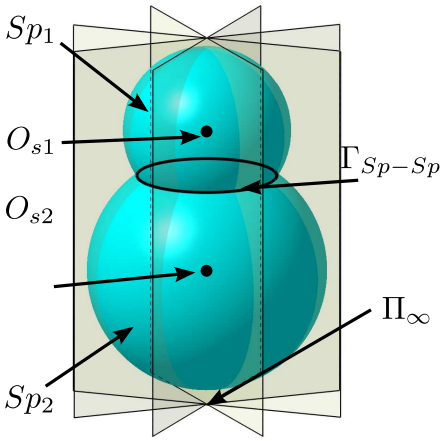
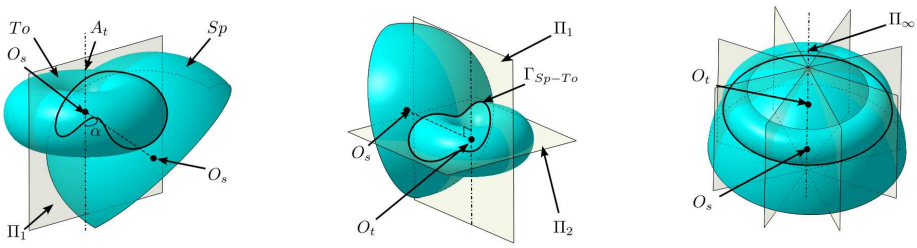


Figure A.10: Intersection and symmetry properties between $(Sp1, Sp2)$.

(F, F_a)	(To, Sp)	(To, Sp)	(To, Sp)
Geometric constraint	$A_t \neq O_s$	$A_t \perp O_s O_t$	$A_t = O_s$
Π	Π_1: $A_t, O_s \subset \Pi_1$	Π_1: $(A_t, O_s) \subset \Pi_1$	∞: axisymmetry
		Π_2: $A_t \perp \Pi_2$, $O_t \subset \Pi_2$	

Table A.10: Configurations of symmetry planes for Sp and To.



(a) (b) (c)

Figure A.11: Symmetry plane configurations of Sp and To.

(F, F_a)	(To_1, To_2)	(To_1, To_2)	(To_1, To_2)
Geometric constraint	$A_{t1} \cap A_{t2} = pt$ $O_{t1}O_{t2}$ not $\perp A_{t1}$	$A_{t1} \parallel A_{t2}$, $O_{t1}O_{t2} \perp A_{t1}$	$A_{t1} \perp A_{t2}$, $O_{t1} = O_{t2}$
Π	Π_1:$(A_{t1}, A_{t2}) \subset \Pi_1$	Π_1:$(A_{t1}, A_{t2}) \subset \Pi_1$, Π_2:$A_{t1} \perp \Pi_2$, $A_{t2} \perp \Pi_2$	Π_1:$(A_{t1}, A_{t2}) \subset \Pi_1$ Π_2:$A_{t1} \subset \Pi_2$, $A_{t2} \perp \Pi_2$, Π_3:$A_{t2} \subset \Pi_3$, $A_{t1} \perp \Pi_3$

(F, F_a)	(To_1, To_2)	(To_1, To_2)	(To_1, To_2)
Geometric constraint	$A_{t1} \perp A_{t2}$, $O_{t1}O_{t2} \perp A_{t1}$	$A_{t1} \perp A_{t2}$, $O_{t1}O_{t2} \perp A_{t1}$, $O_{t1}O_{t2} \perp A_{t2}$	$A_{t1} = A_{t2}$
Π	Π_1:$A_{t1} \perp \Pi_1$, $O_{t1} \subset \Pi_1$	Π_1:$A_{t1} \perp \Pi_1$, $O_{t1} \subset \Pi_1$ Π_2:$A_{t2} \perp \Pi_2$, $O_{t2} \subset \Pi_2$	∞:axisymmetry

Table A.11: Symmetry plane configurations for two tori.

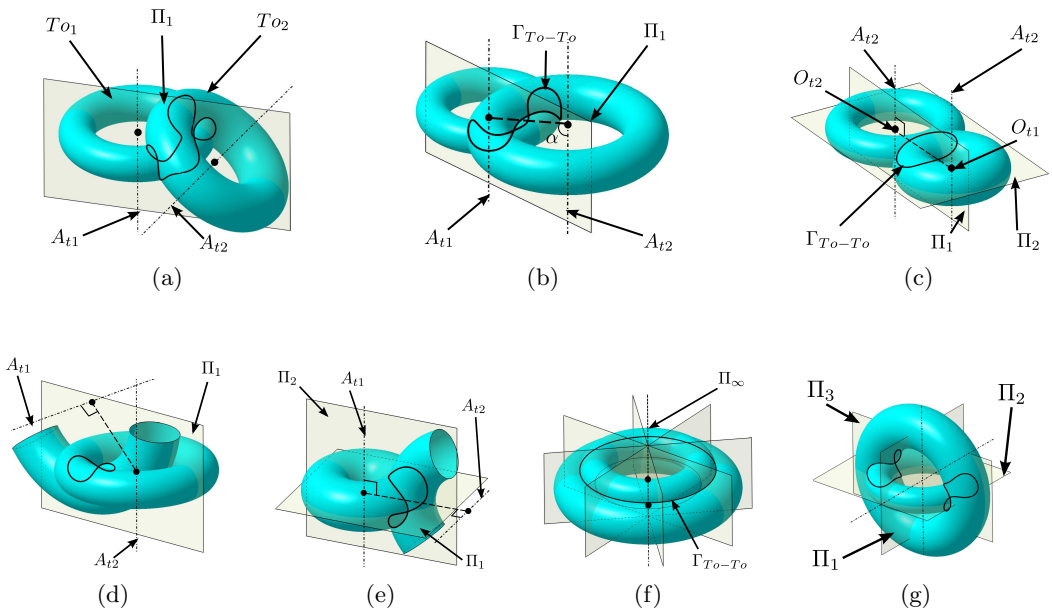


Figure A.12: Symmetry plane configurations of $(To1, To2)$.

Appendix B

Details about the loop status analysis

B.1 Cylinder and cone

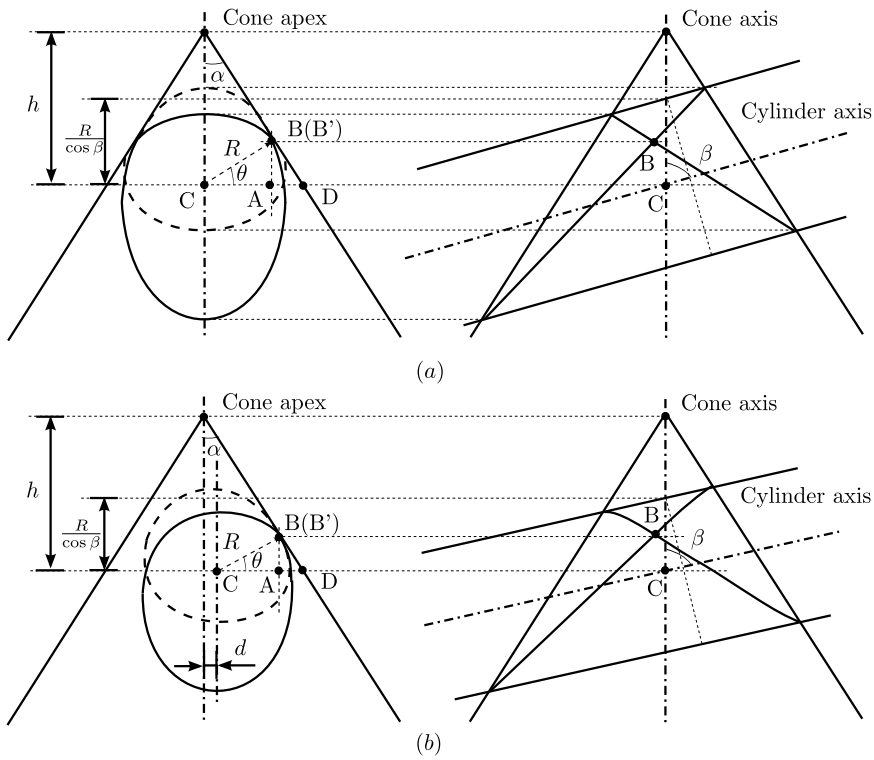


Figure B.1: The tangent loop status generated by (Cy, Co).

Between cylinder and cone, the intersection loop types are divided by the tangent configurations. The Figure B.1a gives the tangent configuration and the parameters where the cone is described by apex, axis and the angle α; the cylinder is described by its axis and radius R. At first, the condition is that axis of cylinder and axis of cone are co-planar and their angle is β. The propose is to get the location of point C when the cylinder is tangent to the cone. Here C is the intersection point between the two axes, which is described by h an CD.

$$\begin{aligned}
\mathrm{AB} &= R\frac{\sin\theta}{\cos\beta} && \text{(B.1)}\\
\mathrm{AD} &= \mathrm{AB}\tan\alpha && \text{(B.2)}\\
\mathrm{CD} &= \mathrm{CA} + \mathrm{AD} \\
&= R\cos\theta + R\frac{\sin\theta}{\cos\beta}\tan\alpha && \text{(B.3)}
\end{aligned}$$

With the tangent condition, B and B' coincide:

$$\mathrm{CD} = \frac{R}{\cos\theta} \qquad \text{(B.4)}$$

$$\begin{aligned}
R\sin\theta\cos\theta\frac{\tan\alpha}{\cos\beta} + R(\cos^2\theta - 1) &= 0 && \text{(B.5)}\\
\sin\theta\cos\theta\frac{\tan\alpha}{\cos\beta} - \sin^2\theta &= 0 \qquad \because \theta \neq 0, \therefore \sin\theta \neq 0 \\
\cos\theta\frac{\tan\alpha}{\cos\beta} &= \sin\theta \\
\tan\theta &= \frac{\tan\alpha}{\cos\beta} && \text{(B.6)}
\end{aligned}$$

The result of the tangent condition is:

$$\begin{aligned}
h\tan\alpha &= \mathrm{CA} + \mathrm{AD} = \mathrm{CD} \\
&= \left[R\cos(\arctan)\frac{\tan\alpha}{\cos\beta} + R\frac{\tan\alpha}{\cos\beta}\sin(\arctan(\frac{\tan\alpha}{\cos\beta}))\right] && \text{(B.7)}
\end{aligned}$$

When $h\tan\alpha$ is bigger or smaller then the condition, it will give a different loop type for the initial cone and the cylinder.

With a more general configuration, the axes of cylinder and cone are not co-planar but at a distance d (see Figure B.1b). The condition is:

$$h\tan\alpha = \mathrm{CD} + d \qquad \text{(B.8)}$$

B.2 Cylinder and torus

Figure B.2a gives the illustration that Cy axis is orthogonal and intersecting with To axis: $A_C \perp A_T$ and $A_C \cap A_T \neq \phi$. In this case:

$$\begin{aligned}
d &= \left\|\overrightarrow{O_T A_C}\right\| && \text{(B.9)}\\
(R_c + r)^2 &= R^2 + d^2 && \text{(B.10)}
\end{aligned}$$

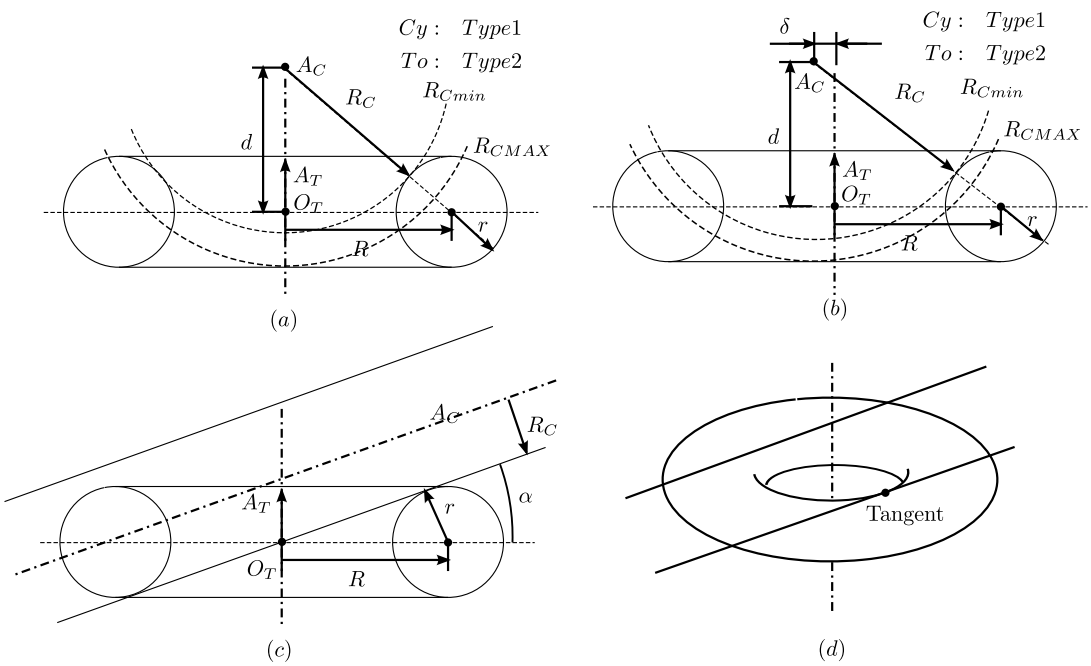


Figure B.2: The intersection loop status generated by (Cy, To).

If the intersection loop is Type 1 for cylinder and type 2 for torus, it needs R_C within:

$$\sqrt{d^2 + R^2} - r < R_C < (d + r) \tag{B.11}$$

When $A_C \perp A_T$ and $A_C \cap A_T = \phi$ (see Figure B.2b), in order to get type 1 loop on cylinder and type 2 loop on torus, the constraint of cylinder radius is:

$$\text{in constraint}: \quad \delta < R \tag{B.12}$$

$$d^2 + (R + \delta^2) = (r + R_C)^2 \tag{B.13}$$

$$\delta \leqslant \sqrt{(r + R_C)^2 - d^2} - R \tag{B.14}$$

$$\text{or} \quad R_C > \sqrt{d^2 + (R + \delta)^2} - r \tag{B.15}$$

The next special configuration is that A_C and A_T are intersecting but not orthogonal. The angle (A_C, A_T) is defined as $\frac{\Pi}{2} - \alpha$, $\alpha \in (0, \Pi)$. The extreme values of α are tangent configurations shown by Figure B.2c:

$$\alpha = \arcsin(\frac{r}{R}) \tag{B.16}$$

Only giving the range of α is not enough to get the intersection loop type. Different intrinsic parameters of torus and cylinder can make cylinder tangent to torus "inside the hole" (see Figure B.2d). The rest of the analysis aims at characterizing the tangent condition. At a tangent point, the normals pointing to torus and cylinder are defined

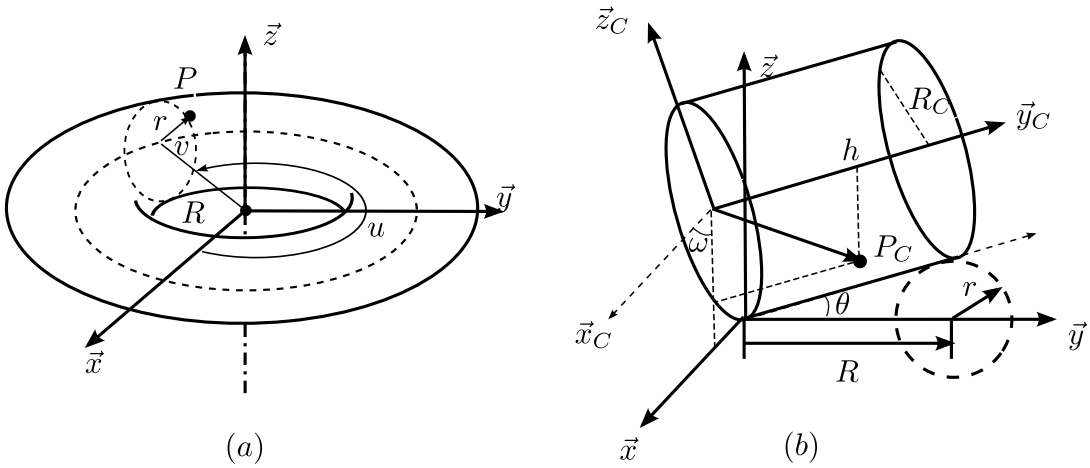


Figure B.3: The intersection loop status generated by (Cy, To): (a) is the parameter representation of torus; (b) is the parameter representation of cylinder with reference frame rotation.

as $\vec{n}$ and $\vec{n}_C$, respectively. Because it is a tangent configuration, $\vec{n}_C \cdot \vec{n} = \pm 1$. A torus is described by intrinsic parameters (R, r) and parameters (u, v), the center is located at the origin of the reference frame. The point P on the torus is:

$$\vec{P} = \begin{cases} x = (R + r\cos v)\cos u \\ y = (R + r\cos v)\sin u \\ z = r\sin v \end{cases} \tag{B.17}$$

The derivatives respect to (u, v) are

$$\frac{\partial \vec{P}}{\partial u} = \begin{cases} -(R + r\cos v)\sin u \\ (R + r\cos v)\cos u \\ 0 \end{cases} \quad \frac{\partial \vec{P}}{\partial v} = \begin{cases} -r\sin v\cos u \\ -r\sin v\sin u \\ r\cos v \end{cases} \tag{B.18}$$

$$\begin{aligned} \frac{\partial \vec{P}}{\partial u} \times \frac{\partial \vec{P}}{\partial v} &= \begin{cases} (R + r\cos v)\cos u \; r\cos v \\ (R + r\cos v)\sin u \; r\cos v \\ (R + r\cos v)\sin u \; r\sin v\sin u + (R + r\cos v)\cos u \; r\sin v\cos u \end{cases} \\ &= \begin{cases} r(R + r\cos v)\cos u\cos v \\ r(R + r\cos v)\sin u\cos v \\ r(R + r\cos v)\sin v \end{cases} \end{aligned} \tag{B.19}$$

$$\left\| \frac{\partial \vec{P}}{\partial u} \times \frac{\partial \vec{P}}{\partial v} \right\| = r(R + r\cos v) \tag{B.20}$$

$$\vec{n}(u, v) = \frac{\frac{\partial \vec{P}}{\partial u} \times \frac{\partial \vec{P}}{\partial v}}{\left\| \frac{\partial \vec{P}}{\partial u} \times \frac{\partial \vec{P}}{\partial v} \right\|} = \begin{cases} \cos u\cos v \\ \sin u\cos v \\ \sin v \end{cases} \tag{B.21}$$

The second derivatives of (u, v) are:

$$\frac{\partial^2 \vec{P}}{\partial u^2} = \begin{cases} -(R + r\cos v)\cos u \\ -(R + r\cos v)\sin u \\ 0 \end{cases} \tag{B.22}$$

$$\frac{\partial^2 \vec{P}}{\partial v^2} = \begin{cases} -r\cos v\cos u \\ -r\cos v\sin u \\ -r\sin v \end{cases} \tag{B.23}$$

$$\frac{\partial^2 \vec{P}}{\partial u \partial v} = \begin{cases} r\sin v\sin u \\ -r\sin v\cos u \\ 0 \end{cases} \tag{B.24}$$

The Gaussian curvature:

$$K = \frac{LN - M^2}{EG - F^2} \tag{B.25}$$

In the Equation B.25:

$$\begin{aligned} L &= \frac{\partial^2 \vec{P}}{\partial u^2} \cdot \vec{n} && \text{(B.26)} \\ &= -(R + r\cos v)\cos^2 u\cos v - (R + r\cos v)\sin^2 u\cos v \\ &= -(R + r\cos v)\cos v \\ N &= \frac{\partial^2 \vec{P}}{\partial v^2} \cdot \vec{n} && \text{(B.27)} \\ &= -r\cos^2 v\cos^2 u - r\cos^2 v\sin^2 u - r\sin^2 v \\ &= -r \\ M &= \frac{\partial^2 \vec{P}}{\partial u \partial v} \cdot \vec{n} && \text{(B.28)} \\ &= r\sin v\sin u\cos u\cos v - r\sin v\cos u\sin u\cos v \\ &= 0 \end{aligned}$$

$$\begin{aligned} E &= \frac{\partial \vec{P}}{\partial u} \cdot \frac{\partial \vec{P}}{\partial u} && \text{(B.29)} \\ &= (R + r\cos v)^2\sin^2 u + (R + r\cos v)^2\cos^2 u \\ &= (R + r\cos v)^2 \\ G &= \frac{\partial \vec{P}}{\partial v} \cdot \frac{\partial \vec{P}}{\partial v} && \text{(B.30)} \\ &= r^2\sin^2 v\cos^2 u + r^2\sin^2 u\sin^2 v + r^2\cos^2 v \\ &= r^2 \\ F &= \frac{\partial \vec{P}}{\partial u} \cdot \frac{\partial \vec{P}}{\partial v} && \text{(B.31)} \\ &= r(R + r\cos v)\sin u\sin v\cos u - r(R + r\cos v)\cos u\sin v\sin u \\ &= 0 \end{aligned}$$

and then, here it is:

$$EG - F^2 = r^2(R + r\cos v)^2 \tag{B.32}$$

Finally, K is:

$$K = \frac{-(R + r\cos v)\cos v(-r) - 0}{r^2(R + r\cos v)^2} = \frac{\cos v(R + r\cos v)}{(R + r\cos v)^2 r^2} = \frac{\cos v}{r(R + r\cos v)} \tag{B.33}$$

The representation of cylinder is illustrated in the Figure B.3(b). The reference frame of cylinder is the $\vec{x}$ rotation result of reference frame torus which rotates θ. The rotation matrix is:

$$R_{\vec{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{B.34}$$

In the cylinder reference frame, the cylinder is represented by intrinsic radius R_C and parameters (h, ω):

$$\vec{P}_C = \begin{cases} x_C = R_C\cos\omega \\ y_C = h \\ z_C = R_C\sin\omega \end{cases} \tag{B.35}$$

So, put the cylinder into the torus reference frame produces:

$$\vec{P}_C = \begin{cases} x = R_C\cos\omega + \delta \\ y = h\cos\theta - R_C\sin\theta\sin\omega \\ z = h\sin\theta + R_C\cos\theta\sin\omega + R_C \end{cases} \tag{B.36}$$

Here δ is the distance between the axes cylinder and torus which is only a $\vec{x}$ parameter.

With the same process calculation, the $\vec{n}_c$ of cylinder is:

$$\frac{\partial\vec{P}_C}{\partial\omega} = \begin{cases} -R_C\sin\omega \\ -R_C\sin\theta\cos\omega \\ R_C\cos\theta\cos\omega \end{cases} \tag{B.37}$$

$$\frac{\partial\vec{P}_C}{\partial h} = \begin{cases} 0 \\ \cos\theta \\ \sin\theta \end{cases} \tag{B.38}$$

$$\frac{\partial \vec{P}_C}{\partial \omega} \times \frac{\partial \vec{P}_C}{\partial h} = \begin{cases} -R_C \cos\omega \\ R_C \sin\omega \sin\theta \\ -R_C \sin\omega \cos\theta \end{cases} \tag{B.39}$$

$$\vec{n}_C = \frac{\frac{\partial \vec{P}_C}{\partial \omega} \times \frac{\partial \vec{P}_C}{\partial h}}{\left\| \frac{\partial \vec{P}_C}{\partial \omega} \times \frac{\partial \vec{P}_C}{\partial h} \right\|} = \begin{cases} -\cos\omega \\ \sin\omega \sin\theta \\ -\sin\omega \cos\theta \end{cases} \tag{B.40}$$

$$\begin{aligned} \vec{n}_C \cdot \vec{n} &= -\cos\omega \cos u \cos v + \sin\omega \sin\theta \sin u \cos v - \sin\omega \cos\theta \sin v \\ &= \pm 1 \end{aligned} \tag{B.41}$$

Another condition is $\vec{n}_c \times \vec{n} = 0$:

$$\vec{n}_c \times \vec{n} = \begin{bmatrix} -\cos\omega \\ \sin\omega \sin\theta \\ \sin\omega \cos\theta \end{bmatrix} \times \begin{bmatrix} \cos\omega \\ \sin u \cos v \\ \sin v \end{bmatrix} \tag{B.42}$$

$$= \begin{cases} \sin\omega \sin\theta \sin v - \sin u \cos v \sin\omega \cos\theta \\ \sin\omega \cos\theta \cos u \cos v + \cos\omega \sin v \\ -\cos\omega \sin u \cos v - \sin\omega \sin\theta \cos u \cos v \end{cases} \tag{B.43}$$

$$= \begin{cases} \sin\omega(\sin\theta \sin v - \sin u \cos v \cos\theta) & = 0 \\ \sin\omega \cos\theta \cos u \cos v + \cos\omega \sin v & = 0 \\ -\cos v(\sin u \cos\omega + \sin\omega \sin\theta \cos u) & = 0 \end{cases} \tag{B.44}$$

Within the equation system, there are (u, v, ω, h) 4 unknowns. The Equation B.41 and B.44 give 4 equations. The tangent configuration can be calculated by these 4 equations but equation system is non linear and requires a numerical approach. This is a much slower process than the processing configurations through analytical treatments. It has also to be considered that curvature compatibility equations based on the Gaussian and principal curvatures need to be added to characterize the loop behavior around the tangent configurations.

B.3 Cone and cone

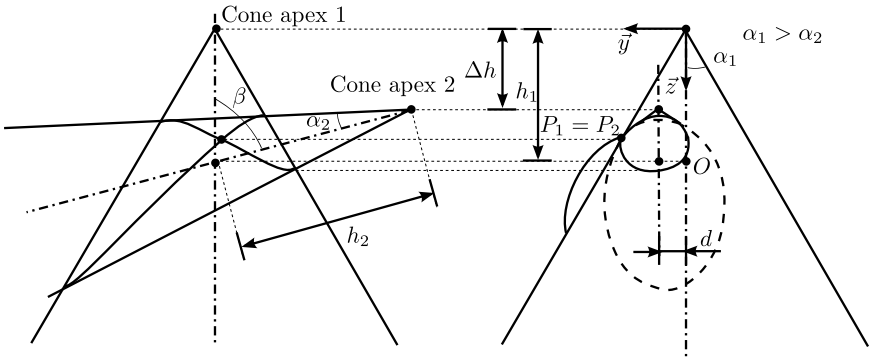


Figure B.4: The tangent loop status generated by (Co, Co).

The analysis of arbitrary configuration between (Co, Co) is similar to the process of (Cy,To). The Figure B.4 gives the condition and the parameters of two cones. The point O is the original point of the reference frame. The axis of cone 1 coincides with the axis $\vec{z}$.

The parameter equation of cone 1 is:

$$\vec{P_1} = \begin{cases} x_1 = h_1 \tan\alpha_1 \cos u_1 \\ y_1 = h_1 \tan\alpha_1 \sin u_1 \\ z_1 = h_1 \qquad h_1 \geqslant 0 \end{cases} \tag{B.45}$$

The derivatives respect to parameters (h_1, u_1) are:

$$\frac{\partial \vec{P_1}}{\partial h_1} = \begin{cases} \tan\alpha_1 \cos u_1 \\ \tan\alpha_1 \sin u_1 \\ 1 \end{cases} \tag{B.46}$$

$$\frac{\partial \vec{P_1}}{\partial u_1} = \begin{cases} -h_1 \tan\alpha_1 \sin u_1 \\ h_1 \tan\alpha_1 \cos u_1 \\ 0 \end{cases} \tag{B.47}$$

and then:

$$\frac{\partial \vec{P_1}}{\partial h_1} \times \frac{\partial \vec{P_1}}{\partial u_1} = \begin{cases} -h_1 \tan\alpha_1 \cos u_1 \\ h_1 \tan\alpha_1 \sin u_1 \\ h_1 \tan^2\alpha_1 \cos^2 u_1 + h_1 \tan^2\alpha_1 \sin^2 u_1 = h_1 \tan^2\alpha_1 \end{cases} \tag{B.48}$$

For the cone 2, with the frame transform matrix, the parameter equation is:

$$\vec{P}_2 = \begin{cases} x_2 = h_2 \tan\alpha_2 \cos u_2 \sin\beta + \Delta h \\ y_2 = h_2 \tan\alpha_2 \sin u_2 + \Delta h + d \\ z_2 = h_2 \cos\beta + \Delta h + h_2 \tan\alpha_2 \cos u_2 \cos\beta \qquad h_2 \geqslant 0 \end{cases} \tag{B.49}$$

The derivatives respect to (h_2, u_2), the result are:

$$\frac{\partial \vec{P}_2}{\partial h_2} = \begin{cases} \tan\alpha_2 \cos u_2 \sin\beta \\ \tan\alpha_2 \sin u_2 \\ \cos\beta + \tan\alpha_2 \cos u_2 \cos\beta \end{cases} \tag{B.50}$$

$$\frac{\partial \vec{P}_2}{\partial u_2} = \begin{cases} -h_2 \tan\alpha_2 \sin u_2 \sin\beta \\ \quad h_2 \tan\alpha_2 \cos u_2 \\ -h_2 \tan\alpha_2 \sin u_2 \cos\beta \end{cases} \tag{B.51}$$

So, there is:

$$\begin{aligned} & \frac{\partial \vec{P}_2}{\partial h_2} \times \frac{\partial \vec{P}_2}{\partial u_2} \\ = & \begin{cases} -h_2 \tan^2\alpha_2 \sin^2 u_2 \cos\beta - h_2 \tan\alpha_2 \cos u_2 (\cos\beta + \tan\alpha_2 \cos u_2 \cos\beta) \\ \\ h_2 \tan^2\alpha_2 \sin u_2 \cos u_2 \sin\beta \cos\beta + \\ h_2 \tan\alpha_2 \sin u_2 \sin\beta (\cos\beta + \tan\alpha_2 \cos u_2 \cos\beta) \\ \\ h_2 \sin\beta \tan^2\alpha_2 \cos^2 u_2 + h_2 \sin\beta \tan^2\alpha_2 \sin^2 u_2 = h_2 \sin\beta \tan^2\alpha_2 \end{cases} \\ = & \begin{cases} -h_2 (\tan^2\alpha_2 \cos\beta + \tan\alpha_2 \cos u_2 \cos\beta) = -h_2 \tan\alpha_2 \cos\beta (\tan\alpha_2 + \cos u_2) \\ h_2 \tan\alpha_2 \sin\beta \cos\beta (\sin u_2 + 2 \tan\alpha_2 \sin u_2 \cos u_2) \\ h_2 \tan^2\alpha_2 \sin\beta \end{cases} \end{aligned} \tag{B.52}$$

The tangent condition makes the cross product between the normals coming from two faces equal to $\vec{0}$:

$$\left(\frac{\partial \vec{P}_1}{\partial h_1} \times \frac{\partial \vec{P}_1}{\partial u_1}\right) \times \left(\frac{\partial \vec{P}_2}{\partial h_2} \times \frac{\partial \vec{P}_2}{\partial u_2}\right) \tag{B.53}$$

$$= \begin{cases} -h_1 \tan\alpha_1 \cos u_1 \\ h_1 \tan\alpha_1 \sin u_1 \\ h_1 \tan^2\alpha_1 \end{cases} \times \begin{cases} -h_2 \tan\alpha_2 \cos\beta(\tan\alpha_2 + \cos u_2) \\ h_2 \tan\alpha_2 \sin\beta \cos\beta \\ (\sin u_2 + 2\tan\alpha_2 \sin u_2 \cos u_2) \\ h_2 \tan^2\alpha_2 \sin\beta \end{cases} \tag{B.54}$$

$$= \begin{cases} h_1 \tan\alpha_1 \sin u_1 h_2 \tan^2\alpha_2 \sin\beta - \\ h_1 \tan^2\alpha_1 h_2 \tan\alpha_2 \sin\beta \cos\beta(\sin u_2 + 2\tan\alpha_2 \sin u_2 \cos u_2) \\ \\ h_1 \tan\alpha_1 \cos u_1 h_2 \tan^2\alpha_2 \sin\beta + \\ h_1 \tan^2\alpha_1 h_2 \tan\alpha_2 \cos\beta(\tan\alpha_2 + \cos u_2) \\ \\ -h_1 \tan\alpha_1 \cos u_1 h_2 \tan\alpha_2 \sin\beta \cos\beta(\sin u_2 + 2\tan\alpha_2 \sin u_2 \cos u_2) - \\ h_1 \tan\alpha_1 \sin u_1 h_2 \tan\alpha_2 \cos\beta(\tan\alpha_2 + \cos u_2) \end{cases}$$

$$= \vec{0}$$

3 equations are generated:

$$(1) \quad \tan\alpha_2 \sin u_1 - \tan\alpha_1 \cos\beta(\sin u_2 + 2\tan\alpha_2 \sin u_2 \cos u_2) = 0 \tag{B.55}$$

$$(2) \quad \tan\alpha_2 \sin\beta \cos u_1 + \tan\alpha_1 \cos\beta(\tan\alpha_2 + \cos u_2) = 0 \tag{B.56}$$

$$(3) - \sin\beta \cos u_1(\sin u_2 + 2\tan\alpha_2 \sin u_2 \cos u_2) - \sin u_1(\tan\alpha_2 + \cos u_2) = 0 \tag{B.57}$$

The Equation B.57 is not independent. The Equation B.55 and B.56 can be transformed as follows:

$$(1) \sin u_1 = \frac{\tan\alpha_1}{\tan\alpha_2} \cos\beta(\sin u_2 + 2\tan\alpha_2 \sin u_2 \cos u_2) \tag{B.58}$$

$$(2) \sin\beta \cos u_1 = -\frac{\tan\alpha_1}{\tan\alpha_2} \cos\beta(\tan\alpha_2 + \cos u_2) \tag{B.59}$$

Equation B.58 is combined with Equation B.59:

$$\tan u_1 = -\sin\beta \frac{\sin u_2 + 2\tan\alpha_2 \sin u_2 \cos u_2}{\tan\alpha_2 + \cos u_2} \tag{B.60}$$

$$= -A_{(u_2)} \tag{B.61}$$

It is a result independent with (h_1, h_2) represented by the parameter $A_{(u_2)}$.

Because on the tangent condition, the tangent point is on both surfaces. So there is a common result between parameters (u_1, h_1) and (u_2, h_2). From the $z_1 = z_2$ of Equations B.45 and B.49, it comes:

$$h_1 = h_2 \cos\beta(1 + \tan\alpha_2 \cos u_2) + \Delta h \tag{B.62}$$

Then $x_1 = x_2$ which is in the Equation B.60 and B.67 is substituted by the Equation B.62:

$$\begin{aligned} h_2 \tan\alpha_1 \cos u_1 \cos\beta(1 + \tan\alpha_2 \cos u_2) + \Delta h \tan\alpha_1 \cos u_1 \\ = h_2 \tan\alpha_2 \cos u_2 \sin\beta + \Delta h \end{aligned} \tag{B.63}$$

$$\begin{aligned} h_2 \left[\tan\alpha_1 \cos u_1 \cos\beta(1 + \tan\alpha_2 \cos u_2) - \tan\alpha_2 \cos u_2 \sin\beta\right] \\ = \Delta h(1 - \tan\alpha_1 \cos u_1) \end{aligned} \tag{B.64}$$

$$h_2 = \frac{\Delta h(1 - \tan\alpha_1 \cos u_1)}{\left[\tan\alpha_1 \cos u_1 \cos\beta(1 + \tan\alpha_2 \cos u_2) - \tan\alpha_2 \cos u_2 \sin\beta\right]} \tag{B.65}$$

Put h_1 in Equation B.62 and h_2 in Equation B.65 into y, here is:

$$\begin{aligned} &\left[\frac{\Delta h(1 - \tan\alpha_1 \cos u_1) \cos\beta(1 + \tan\alpha_2 \cos u_2)}{\tan\alpha_1 \cos u_1 \cos\beta(1 + \tan\alpha_2 \cos u_2) - \tan\alpha_2 \cos u_2 \sin\beta} + \Delta h\right] \tan\alpha_1 \sin u_1 \\ =&\frac{\Delta h(1 - \tan\alpha_1 \cos u_1) \tan\alpha_2 \sin u_2}{\tan\alpha_1 \cos u_1 \cos\beta(1 + \tan\alpha_2 \cos u_2) - \tan\alpha_2 \cos u_2 \sin\beta} + \Delta h + d \end{aligned} \tag{B.66}$$

On the other hand, $y_1 = y_2$ substituted by $x_1 = x_2$, here is:

$$\tan u_1 = \frac{h_2 \tan\alpha_2 \sin u_2 + \Delta h + d}{h_2 \tan\alpha_2 \cos u_2 \sin\beta + \Delta h} = B_{(h_2, u_2)} \tag{B.67}$$

Combine the Equation B.60 and B.67, the result is:

$$h_2 \tan\alpha_2 \sin u_2 + \Delta h + d = -A(h_2 \tan\alpha_2 \cos u_2 \sin\beta + \Delta h) \tag{B.68}$$

$$h_2 \tan\alpha_2 (\sin u_2 + A \cos u_2 \sin\beta) = -A\Delta h - \Delta h - d \tag{B.69}$$

$$\begin{aligned} h_2 &= -\frac{A(1 + \Delta h) + d}{\tan\alpha_2(\sin u_2 + A \cos u_2 \sin\beta)} \\ &= -C_{(u_2)} \end{aligned} \tag{B.70}$$

Transform the Equation B.67, here is:

$$\begin{aligned} &\text{let}: \quad t = \tan\frac{u_1}{2} \\ &\tan u_1 = \frac{2t}{1 - t} = B_{(h_2, u_2)} \end{aligned} \tag{B.71}$$

So:

$$B_{(h_2,u_2)}t^2 + 2t - B_{(h_2,u_2)} = 0 \tag{B.72}$$

$$\begin{aligned} t &= \frac{-2 \pm 2\sqrt{1 - B^2_{(h_2,u_2)}}}{2B_{(h_2,u_2)}} \\ &= \frac{-1 \pm \sqrt{1 - B^2_{(h_2,u_2)}}}{B_{(h_2,u_2)}} \end{aligned} \tag{B.73}$$

From $x_1 = x_2$ of the Equation B.45 and B.49, here is:

$$h_1 \tan\alpha_1 \cos u_1 = h_2 \tan\alpha_2 \cos u_2 \sin\beta + \Delta h \tag{B.74}$$

Replacing h_1 by the Equation B.62, and using t to represent $\cos u_1$, the result is:

$$[h_2 \cos\beta\,(1 + \tan\alpha_2 \cos u_2) + \Delta h] \tan\alpha_1 \left(\frac{1 - t^2}{1 + t^2}\right) = h_2 \tan\alpha_2 \cos u_2 \sin\beta + \Delta h \tag{B.75}$$

Here t is represented by $B_{(h_2,u_2)}$. Then replacing h_2 by $C_{(u_2)}$ represented in the Equation B.70, the Equation B.75 has only one unknown u_2. It is a non linear equation. So, it needs numerical processing.

B.4 Cone and torus, torus and torus

The intersection analysis between (Co, To) and (To, To) are similar with the previous works, such as the analysis of (Cy, To) or (Co, To). Finally, the result are equation systems of high order which needs special mathematical method, such as numerical analysis, to give the solution. The purpose of these combinatorial analyses is to get the type of the loop without vertex and to reduce the number of candidate symmetry planes to process during the LS-CSP generation process. There is no need to spend much more time on solving non-linear equations. Consequently, the configurations which can give loop type directly are discussed in the chapter 5. The analysis of the other configurations are listed in this appendix. But within the symmetry analysis algorithm, when generating the LS-CSP, it shows that the configurations listed here may not supersede the combinatorial process where loops are processed combinatorially. A more detailed analysis would be needed to evaluate the efficiency of each approach.

Appendix C

More examples and statistic

The purpose of this appendix is to illustrate the behavior of the algorithm as implemented using the OpenCascade software library through a larger of example shapes and to evaluate the correlation with the complexity analysis performed in the main part of the manuscript. This appendix addresses also the behavior of the algorithm with respect to accuracy issues.

First of all, let us go through the range of test models used for the performance evaluation of the algorithm. Figures C.2, C.3, C.4 describe the shape of each test model and its corresponding identifier that will be used in the performance charts to identify each model.

As a parameter to characterize the complexity of a model, the sum of faces, edges and vertices is used throughout this performance analysis. This is consistent with the complexity analysis conducted at each step of the algorithm since faces, edges and vertices of the B-Rep model input where the essentially parameters of these analyses. However, using the sum of these parameters assumes that the time required for each elementary operation applied to either of these entities is nearly identical. This similitude has been observed precisely because of the variability of each configuration where the operations are conducted. The description of the detailed content of each operator does not show large differences of elementary operations between them, which is a first justification of the chosen assumption.

Figure C.1 shows the number of Face+Edge+Vertex of the different models. Their variation in complexity is a compromise between their symmetry properties and their functional meaning as components of digital mock-ups of industrial products. Figure C.5 and Table C.1 show the relationship between the total elements number and the total time cost of the symmetry analysis, i.e. determining global as well as local symmetries. Under the above assumptions, it shows that except the models $M04$, $M16$ and $M28$, generally, the time cost is evolving rather linearly. Regarding these three models, they all look similar but they aren't. $M16$ distinguishes from the others in the facts that it contains a face bounded by 14 loops, which can generate a fair amount of LS-CSPs, and the processing time is based on a basic and purely combinatorial implementation of the LS-CSPs generation. Additionally, all three models contain faces with loops containing a large amount of edges. The version of the algorithm scanning all the edges of a face during the first level propagation, it has a significant impact over the efficiency of this subset of the algorithm.

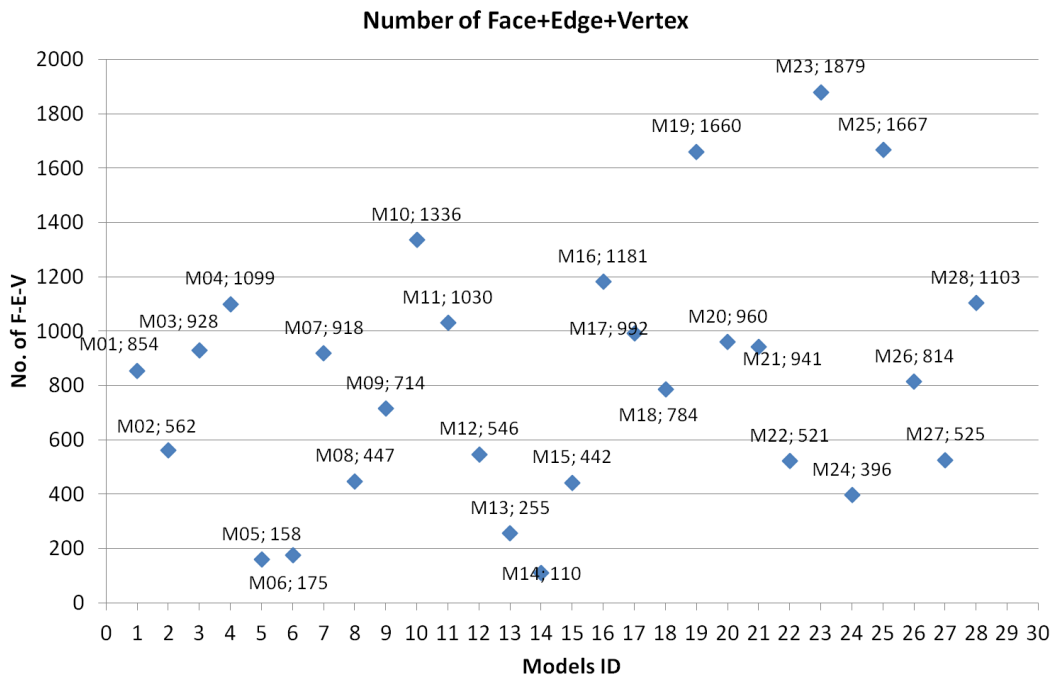


Figure C.1: The total number of Face+Edge+Vertex per test model Mxx.

Figures C.6, C.7, C.8 and Tables C.2, C.3, C.4 give the statistics about the time cost for different algorithm steps and their plots with respect to an increasing model complexity. By analyzing the different processes, the time cost of the 1st level propagation process is not linear. Models $M04$, $M16$ and $M28$, which are variants of one original model, cost more time than the others and don't follow the linear time evolution. Reasons explaining these differences have been stated previously. A complementary aspect amplifying this behavior can also originates from a common hypothesis of complexity analysis: the data access is assumed to be to constant and identical for all faces, edges and vertices. Indeed, the current implementation of the algorithm uses multimaps in order to simplify the development and maintenance compared to the use of pointers. However, this data access is no longer constant and can be comparable in amount to the operators of boundary preparation or first level propagation, which can have a significant influence on the corresponding task time.

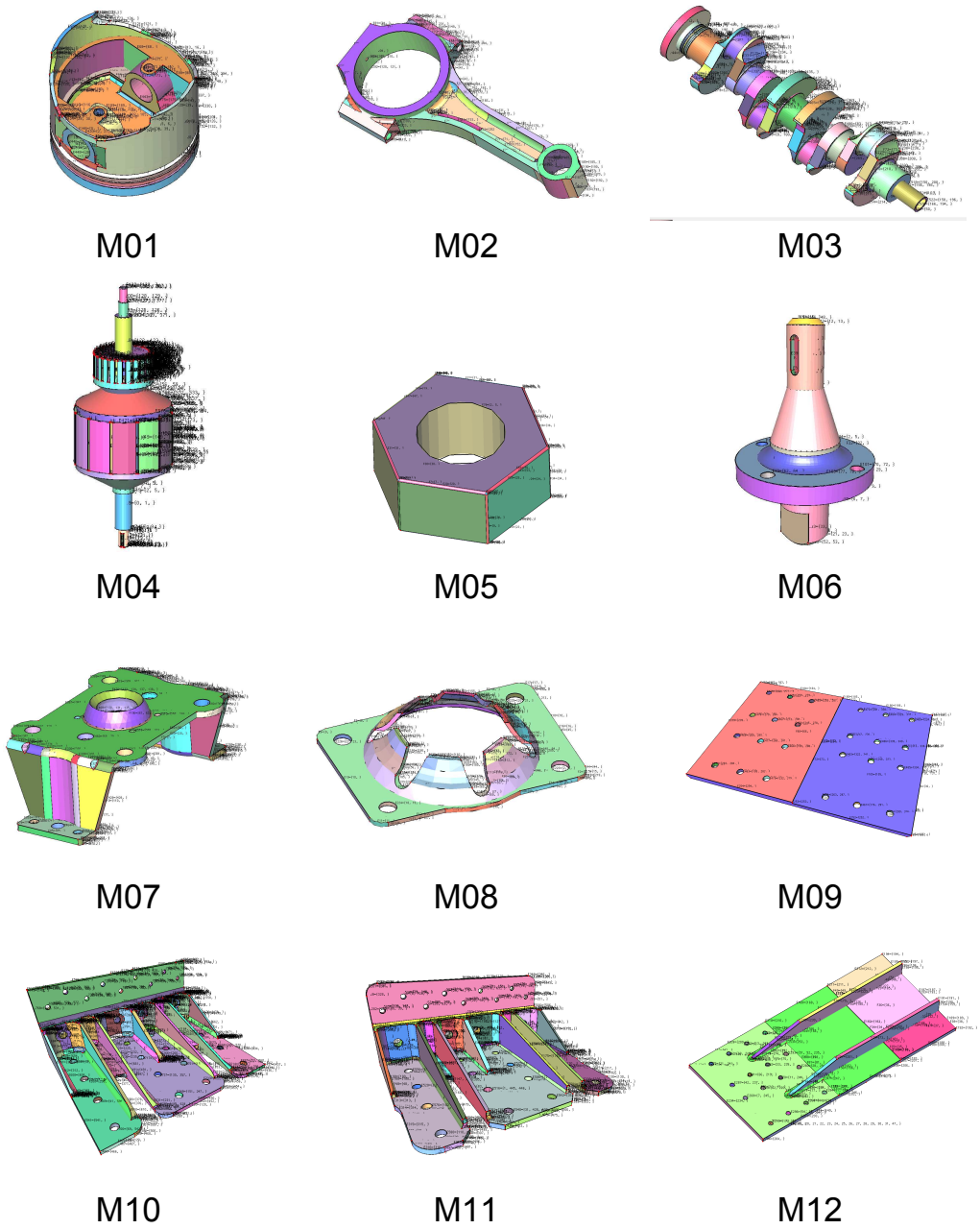


Figure C.2: The list of example models in the statistics. Subsection 1.

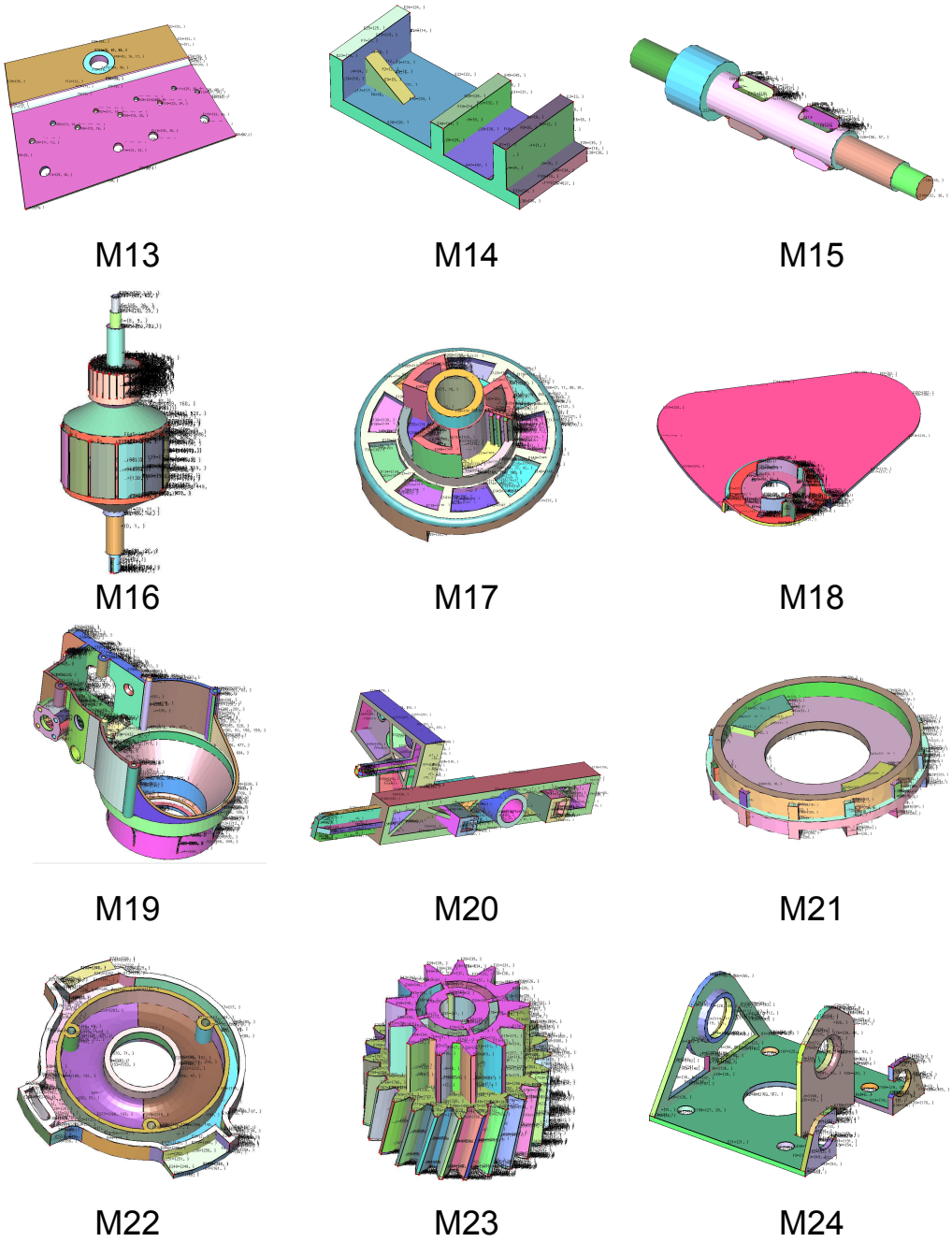


Figure C.3: The list of example models in the statistics. Subsection 2.

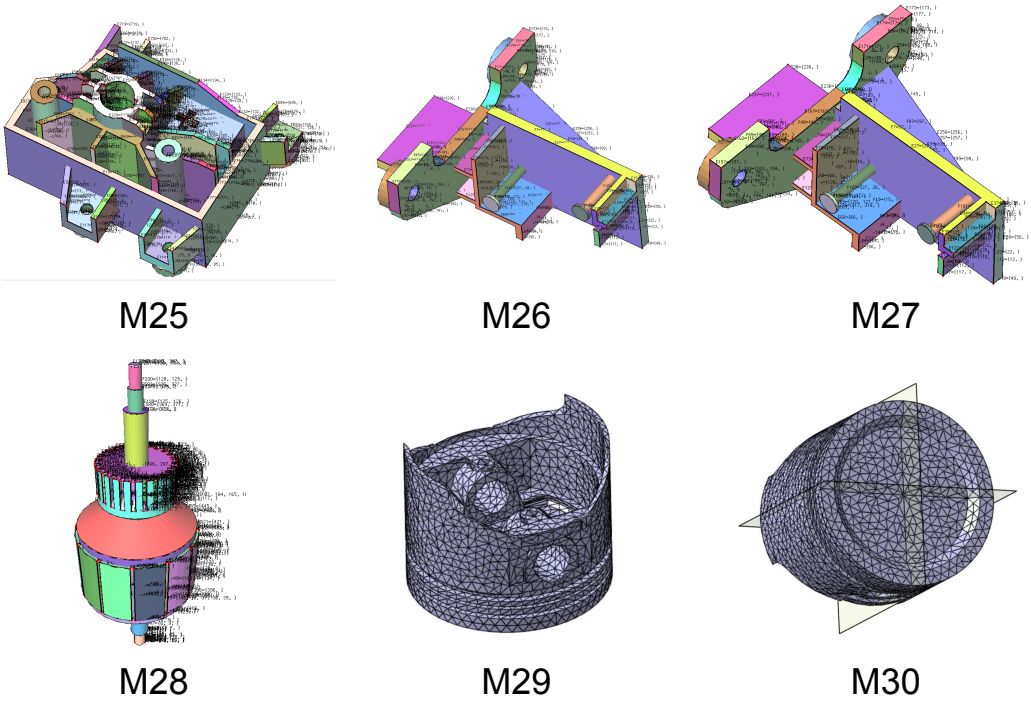


Figure C.4: The list of example models in the statistics. Subsection 3.

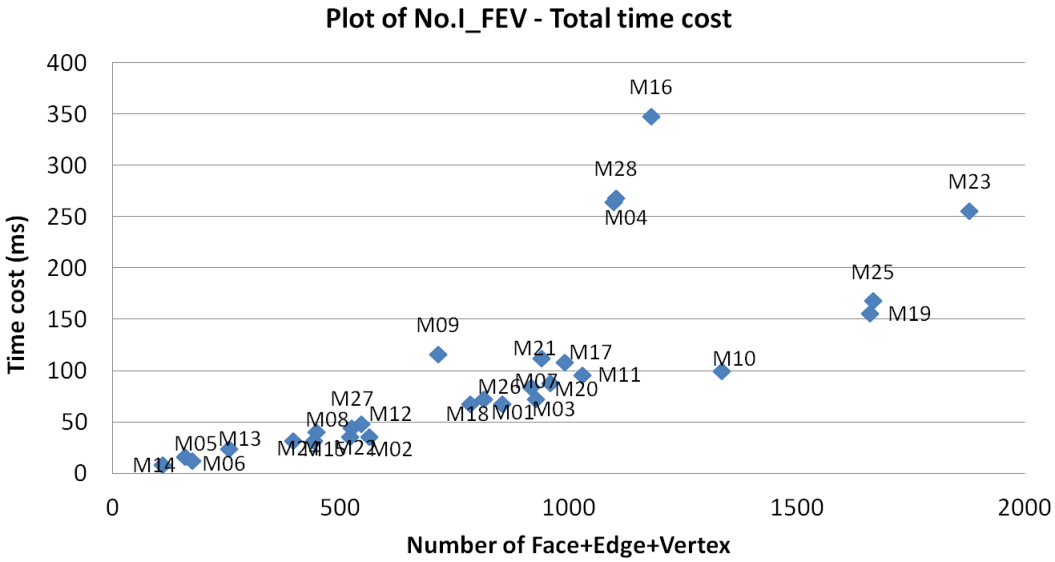


Figure C.5: The statistic of the total time cost with respect to the number of Face+Edge+Vertex, i.e. the model complexity.

	M01	M02	M03	M04	M05	M06	M07
No. of F-E-V	854	562	928	1099	158	175	918
Total time(ms)	68,005	36,002	72,003	264,016	16,001	12,001	84,006
	M08	M09	M10	M11	M12	M13	M14
No. of F-E-V	447	714	1336	1030	546	255	110
Total time(ms)	40,003	116,008	100,007	96,006	48,003	24,002	8,001
	M15	M16	M17	M18	M19	M20	M21
No. of F-E-V	442	1181	992	784	1660	960	941
Total time(ms)	32,002	348,022	108,007	68,004	156,011	88,006	112,006
	M22	M23	M24	M25	M26	M27	M28
No. of F-E-V	521	1879	396	1667	814	525	1103
Total time(ms)	36,002	256,016	32,002	168,01	72,003	44,003	268,017

Table C.1: The data used in Figure C.5, the total number of Face+Edge+Vertex and the total processing time for each model.

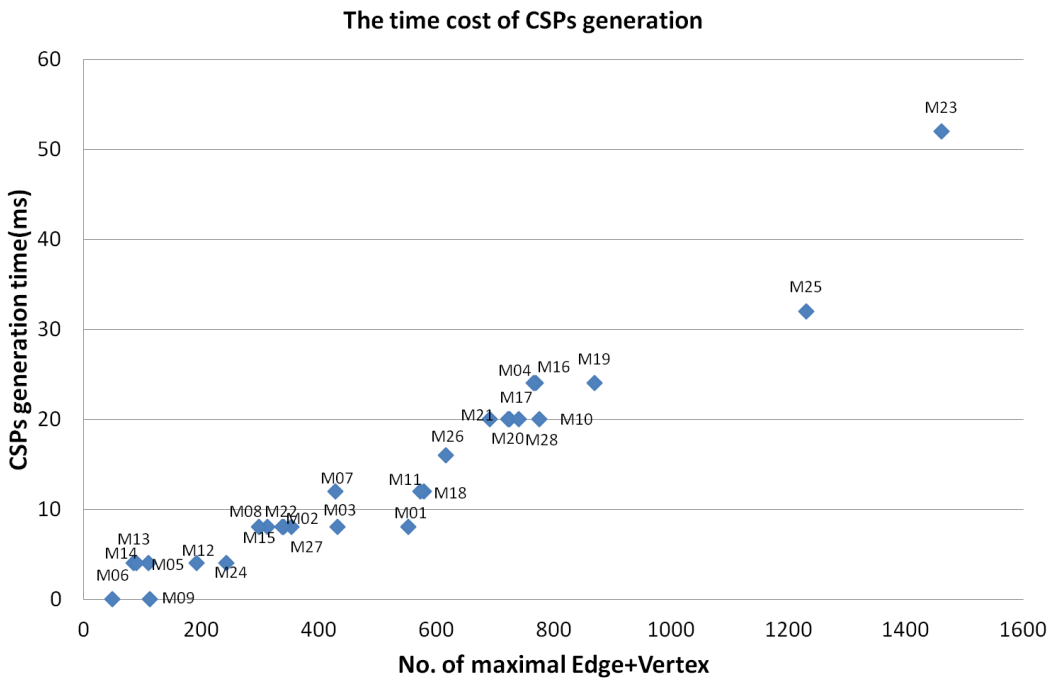


Figure C.6: The statistics of the CSPs generation step and its corresponding processing time.

	M01	M02	M03	M04	M05	M06	M07
No. Maximal E-V	553	354	432	770	110	49	428
CSPs generation time(ms)	8,001	8	8	24,002	4	0	12,001
	M08	M09	M10	M11	M12	M13	M14
No. Maximal E-V	312	112	775	579	192	85	90
CSPs generation time(ms)	8,001	0	20,002	12,001	4	4	4
	M15	M16	M17	M18	M19	M20	M21
No. Maximal E-V	298	766	725	573	869	723	691
CSPs generation time(ms)	8,001	24,002	20,001	12	24,002	20,001	20,001
	M22	M23	M24	M25	M26	M27	M28
No. Maximal E-V	340	1460	242	1230	616	338	741
CSPs generation time(ms)	8	52,003	4	32,002	16,001	8	20,001

Table C.2: The data of Figure C.6: the number of maximal Edge+Vertex and the processing time of CSPs generation.

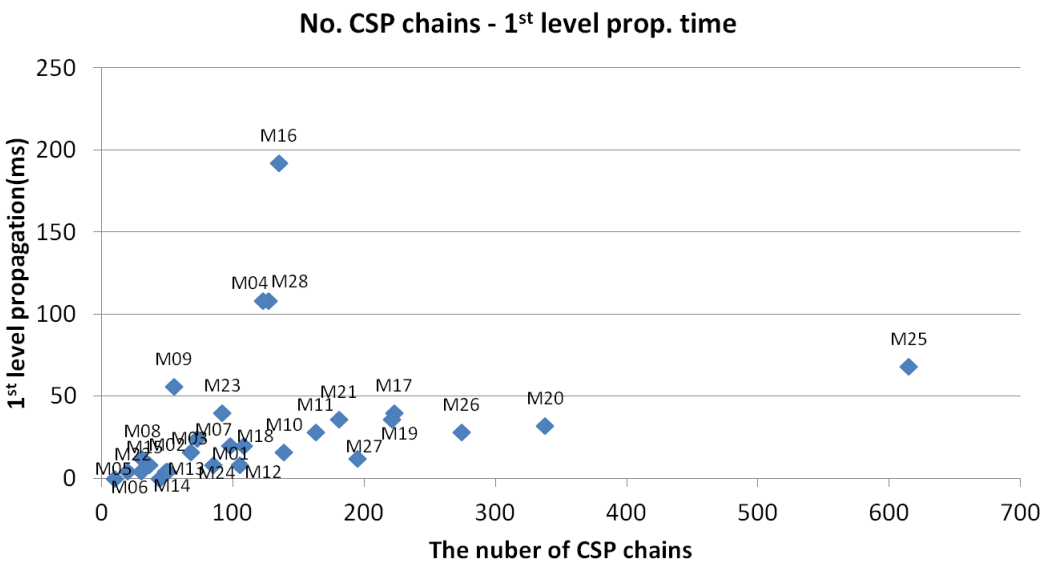


Figure C.7: The statistics of the 1st propagation process time with respect to the number of CSP chains generated.

	M01	M02	M03	M04	M05	M06	M07
No. CSP Chains	98	50	68	127	20	10	73
1st level prop. (ms)	20,001	4	16,001	108,006	4	0	24,002
	M08	M09	M10	M11	M12	M13	M14
No. CSP Chains	31	55	139	163	105	44	49
1st level prop. (ms)	12	56,003	16,001	28,002	8	0	4,001
	M15	M16	M17	M18	M19	M20	M21
No. CSP Chains	36	135	223	108	221	338	181
1st level prop. (ms)	8	192,012	40,003	20,002	36,002	32,002	36,002
	M22	M23	M24	M25	M26	M27	M28
No. CSP Chains	30	92	85	615	274	195	123
1st level prop. (ms)	4,001	40,003	8,001	68,005	28,001	12,001	108,007

Table C.3: The data of Figure C.7, the relationship between the number of CSP chains and 1st level propagation processing time.

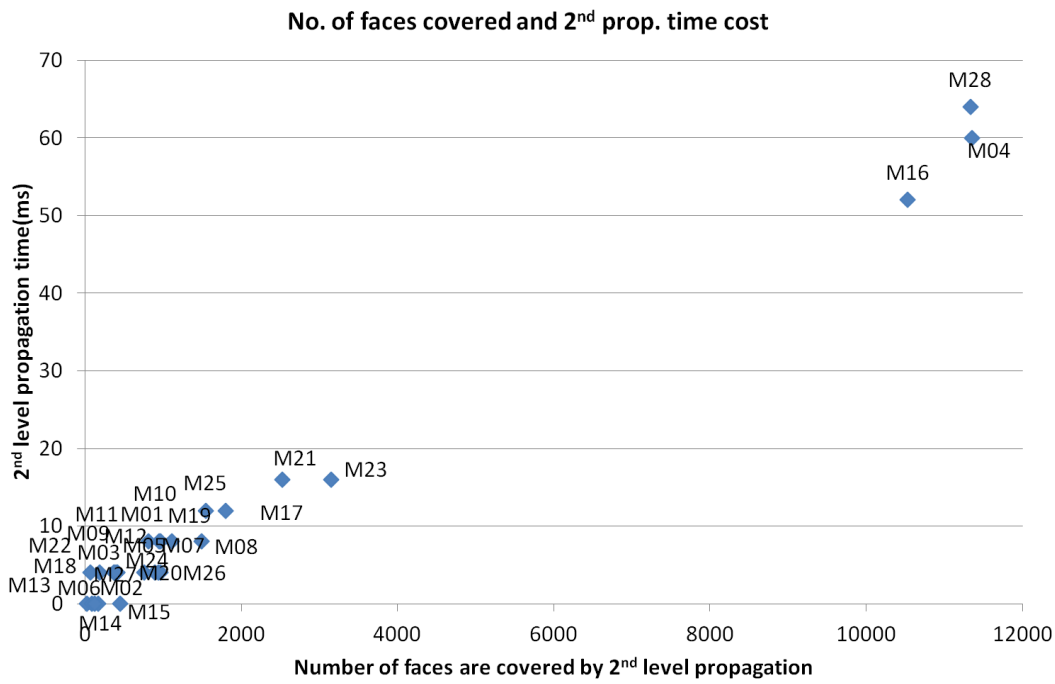


Figure C.8: The statistics of the 2nd propagation time with respect to the number of faces covered.

	M01	M02	M03	M04	M05	M06	M07
No. of face covered	944	170	180	11352	758	24	938
2nd level propagation(ms)	8,001	0	4	60,004	4	0	4
	M08	M09	M10	M11	M12	M13	M14
No. of face covered	1490	408	806	964	410	82	124
2nd level propagation(ms)	8,001	4,001	8	8	4,001	0	0
	M15	M16	M17	M18	M19	M20	M21
No. of face covered	444	10532	1800	376	1112	890	2526
2nd level propagation(ms)	0	52,003	12,001	4	8,001	4,001	16,001
	M22	M23	M24	M25	M26	M27	M28
No. of face covered	68	3144	362	1542	962	390	11330
2nd level propagation(ms)	4	16,001	4	12	4	4	64,004

Table C.4: The data of Figure C.8, the number of face covered and the 2nd propagation process time.

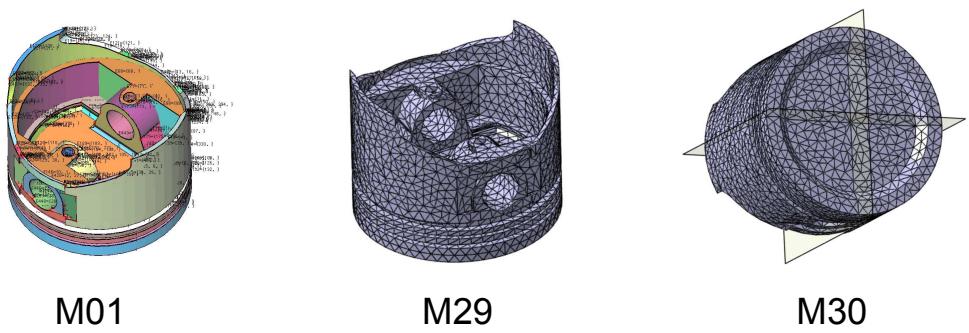


Figure C.9: The shape comparison of the different piston representations.

	M01	M29	M30
No. of Faces	146	7442	7524
No. of Edges	423	11163	11286
No. of Vertices	285	3711	3752
Total time coset(ms)	68,005	22316	22562
No. of maximal Face+Edge	553	9250	7200
CSPs generation time(ms)	8,001	444,028	436,027
No. CSP chains	98	6404	6353
1st level prop. (ms)	20,001	1788,11	1956,12
No. of face covered	944	86404	131280
2nd level propagation(ms)	8,001	632,04	980,062

Table C.5: The data corresponding to tests with the different piston representations.

Now, the purpose is to illustrate the behavior of the algorithm with respect to the accuracy of the input model. Throughout the performance tests performed, the accuracy parameter used in the algorithm is a distance of 0.001mm, which similar to the accuracy used in CATIA V5 to connect surface patches or perform other similar operations.

To this end, figure C.9 and Table C.5 highlight the comparison between different representations of the same object: a piston. As a smooth B-Rep model, $M01$ has less elements than models $M29$ and $M30$ that are faceted representations of the same object (see Table C.5). $M01$ has 2 symmetry planes. $M29$ is a faceted model of $M01$ that has been generated with a common mesh generator. As a result, the mesh topology and its vertices distribution are no longer precisely symmetric and the global symmetry planes are lost. $M30$ is also a faceted model of $M01$. This mesh has been generated by a mirror operation applied to one quarter of the faceted piston. So, it is symmetric with a high accuracy: the distance accuracy is of the order 10^{-8}mm for single precision arithmetic. Thus, the result gives 2 symmetry planes. This behavior shows that if the location of vertices in a faceted model is not monitored within the accuracy of the algorithm, symmetry properties will be lost between the smooth representation and the faceted one. Consequently, a very highly dense mesh would be necessary to obtain the same symmetry properties as the initial smooth model, which could not be processed interactively.

Benefiting from this example, however, it can be observed that the algorithm scales well with respect to the model complexity because the faceted models have been processed like smooth ones while their complexity reaches over 20 000 entities.

Also, perturbing the location of one arbitrary vertex of the symmetrically faceted model with a displacement of 0.001mm shows that the 2 symmetry planes are no longer found. Again, this test confirms the fact that the symmetry properties can be analyzed at the level of accuracy of the modeling kernel.

Bibliography

[1] C.W. Li A.K.C. Wong. Visual transformation of a 3-d object in attribute hypergraph representatioin. In *Model-Based Vision Proceedings of SPIE Volume: 1827*, pages 29–40. SPIE, 1993. 63

[2] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Levy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM Transactions on Graphics (SIGGRAPH conference proceedings)*, 22(3):485–493, Jul 2003. 26

[3] Helmut Alt, Kurt Mehlhorn, Hubert Wagener, and Emo Welzl. Congruence, similarity and symmetries of geometric objects. *Discrete Comput. Geom.*, 3(3):237–256, 1988. 21

[4] M. J. Atallah. On symmetry detection. *IEEE Trans. Comput.*, 34(7):663–666, 1985. 20

[5] Alexander Berner, Martin Bokeloh, Michael Wand, Andreas Schilling, and Hans-Peter Seidel. Generalized intrinsic symmetry detection. Research Report MPI-I-2009-4-005, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, August 2009. 23, 24

[6] Alexander Berner, Michael Wand, Niloy J. Mitra, Daniel Mewes, and Hans-Peter Seidel. Shape analysis with subspace symmetries. *Computer Graphics Forum*, 30(2), 2011. Proceedings Eurographics 2011. 23

[7] M. Bokeloh, A. Berner, M. Wand, H. P. Seidel, and A. Schilling. Symmetry Detection Using Feature Lines. *Computer Graphics Forum*, 28(2):697–706, 2009. 23

[8] Martin Bokeloh, Michael Wand, and Hans-Peter Seidel. A connection between partial symmetry and inverse procedural modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)*, 29(4):104,1–104,10, 2010. 23

[9] Peter Brass and Christian Knauer. Testing congruence and symmetry for general 3-dimensional objects. *Comput. Geom. Theory Appl.*, 27(1):3–11, 2004. 21

[10] David Cailliere, Florence Denis, Danielle Pele, and Atilla Baskurt. 3d mirror symmetry detection using hough transform. In *ICIP'08*, pages 1772–1775, 2008. 18, 28

[11] David Caillière, Florence Denis, Danielle Pelè, and Atilla Baskurt. Symmetry analysis with multiscale descriptor . In *SPIE Electronic Imaging - 3D Image Processing (3DIP) and Applications*, page 75260. SPIE, January 2010. 29

[12] L. Paul Chew, Michael T. Goodrich, Daniel P. Huttenlocher, Klara Kedem, Jon M. Kleinberg, and Dina Kravets. Geometric pattern matching under euclidean motion. *Comput. Geom. Theory Appl.*, 7:113–124, January 1997. 21

[13] Michal Choras and Tomasz Andrysiak. Symmetry-based salient points detection in face images. In *ICAISC*, pages 758–767, 2006. 8

[14] LARRY S. DAVIS. Understanding Shape: II. Symmetry. *Systems, Man and Cybernetics, IEEE Transactions on*, 7(3):204–212, 1977. 31

[15] Leila De Floriani and Bianca Falcidieno. A hierarchical boundary model for solid object representation. *ACM Trans. Graph.*, 7:42–60, January 1988. 63, 64

[16] Christian Dick, Joachim Georgii, and Rüdiger Westermann. A real-time multigrid finite hexahedra method for elasticity simulation using CUDA. *Simulation Modelling Practice and Theory*, 19(2):801–816, 2011. 19

[17] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15:11–15, January 1972. 28

[18] G. Foucault. *Adaptation de modèles CAO paramétrés en vue d'une analyse de comportement mcanique par éléments finis.* PhD thesis, INPG, France, Universit du Qubec, Canada, 2007. 97

[19] Gilles Foucault, Jean-Christophe Cuillière, Vincent François, Jean-Claude Léon, and Roland Maranzana. Adaptation of cad model topology for finite element analysis. *Comput. Aided Des.*, 40(2):176–196, 2008. 63, 64

[20] Cecil G. and Armstrong. Modelling requirements for finite-element analysis. *Computer-Aided Design*, 26(7):573 – 578, 1994. 32

[21] Dilay BİNGÜL Hacer Şengül AKAÇ. Hausdorff distance for shape matching. Technical report, May 2010. 21

[22] O. Hamri. *Method, models and tools for Finite Element model preparation integrated into a product development process.* PhD thesis, INPG, France, University of Genoa, Italy, 2006. 59

[23] Berthold K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(2):1671–1686, 1984. 29

[24] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:850–863, September 1993. 21

[25] Daniel P. Huttenlocher, Klara Kedem, and Jon M. Kleinberg. On dynamic voronoi diagrams and the minimum hausdorff distance for point sets under euclidean motion in the plane. In *Proceedings of the eighth annual symposium on Computational geometry*, SCG '92, pages 110–119, New York, NY, USA, 1992. ACM. 21

[26] Robert Iacob, Peter Mitrouchev, and Jean-Claude Léon. Contact identification for assemblydisassembly simulation with a haptic device. *The Visual Computer*, 24:973–979, 2008. 10.1007/s00371-008-0275-3. 39

[27] Sebastian Iwanowski. Testing approximate symmetry in the plane is np-hard. In *Proceedings on Mathematical Foundations of Computer Science 1989*, MFCS '89, pages 291–304, London, UK, 1989. Springer-Verlag. 21

[28] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 30

[29] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Symmetry descriptors and 3d shape matching. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 115–123, New York, NY, USA, 2004. ACM. 30

[30] Michael M. Kazhdan, Bernard Chazelle, David P. Dobkin, Adam Finkelstein, and Thomas A. Funkhouser. A reflective symmetry descriptor. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 642–656, London, UK, 2002. Springer-Verlag. 30

[31] Shankar Krishnan and Suresh Venkatasubramanian. Approximate shape matching and symmetry detection for 3d shapes with guaranteed error bounds. In Jun-Hai Yong, Michela Spagnuolo, and Wenping Wang, editors, *Shape Modeling International*, pages 44–51. IEEE Computer Society, 2009. 21

[32] SAIGAL R. KULKARNI P., DUTTA D. An investigation of techniques for asymmetry rectification. In *Trans ASME J Mech Des 1995*, volume 117, pages 620–626, 1995. 32

[33] Frank Langbein, Bruce I. Mills, A. Dave Marshall, and Ralph R. Martin. Finding approximate shape regularities in reverse engineered solid models bounded by simple surfaces. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, SMA '01, pages 206–215, New York, NY, USA, 2001. ACM. 22, 23

[34] Michael Leyton. *A Generative Theory of Shape*. Number LNCS 2145 in Lecture Notes in Computer Science. Springer-Verlag, 2001. 4, 7

[35] M. Li, F. C. Langbein, and R. R. Martin. Detecting approximate incomplete symmetries in discrete point sets. In *ACM Symp. Solid and Physical Modeling*, pages 335–340. ACM Siggraph, 2007. 22, 126

[36] Ming Li, Frank C. Langbein, and Ralph R. Martin. Detecting approximate symmetries of discrete point subsets. *Comput. Aided Des.*, 40:76–93, January 2008. 22, 24, 126

[37] Sheena X. Liu. Symmetry and asymmetry analysis and its implications to computer-aided diagnosis: A review of the literature. *Journal of Biomedical Informatics*, July 2009. 9

[38] Ralph Martin, Deba Dutta, R. R. Martin, Department Of Computing Mathematics, and D. Dutta. Tools for asymmetry rectification in shape design. *Journal of Systems Engineering*, 6:98–112, 1996. 10, 32

[39] Aurélien Martinet, Cyril Soler, Nicolas Holzschuch, and François X. Sillion. Accurately Detecting Symmetries of 3D Shapes. 07 2005. 30

[40] Aurélien Martinet, Cyril Soler, Nicolas Holzschuch, and François X. Sillion. Accurate Detection of Symmetries in 3D Shapes. *ACM Transaction on Graphics*, 25(2):439 – 464, April 2006. 30, 126

[41] B. I. Mills, F. C. Langbein, A. D. Marshall, and R. R. Martin. Approximate symmetry detection for reverse engineering. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, SMA '01, pages 241–248, New York, NY, USA, 2001. ACM. 22

[42] P. Minovic, S. Ishikawa, and K. Kato. Symmetry identification of a 3-d object represented by octree. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:507–514, May 1993. 24, 25

[43] N. J. Mitra, L. Guibas, and M. Pauly. Symmetrization. In *ACM Transactions on Graphics*, volume 26, pages #63, 1–8, 2007. 3, 26

[44] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.*, 25(3):560–568, 2006. 9, 10, 12, 26, 28, 126, 237

[45] OpenCascade work group. *Object Libraries Modeling Algorithms User's Guide*, version 6.3 edition, September 2008. 59

[46] Wai pak Choi, Kin man Lam, and Wan chi Siu. A robust line-feature-based hausdorff distance for shape matching. In *IEEE Pacific Rim Conference on Multimedia*, pages 764–771, 2001. 21

[47] Stephen Parry-barwick and Adrian Bowyer. Woodwark's method for feature recognition. Technical report, University of Bath School of Mechanical Engineering, 1992. 31, 32

[48] Stephen Parry-barwick and Adrian Bowyer. Symmetry analysis and geometric modelling. In *Proceedings DCTA'93, Digital Image Computing-Techniques and Applications*, pages 39–46, Australia, Sydney, 1993. Australian Pattern Recognition Society. 31, 32

[49] S.K. Parui and D. Dutta Majumder. Symmetry analysis by computer. *Pattern Recognition*, 16(1):63 – 67, 1983. 31

[50] Constantine Kotropoulos Pawe Rotter, Andrzej M.J. Skulimowski and Ioannis Pitas. Fast shape matching using the hausdorff distance. In *Proc. of MIRAGE 2005*, 2005. 21

[51] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3d shapes. *ACM Transactions on Graphics (Proc. Siggraph*, 25:549–559, 2006. 26

[52] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Symmetries of non-rigid shapes. In *Proc. Non-rigid Registration and Tracking (NRTL) workshop. See Proc. of International Conference on Computer Vision (ICCV)*, October 2007. 10

[53] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Full and partial symmetries of non-rigid shapes. *International Journal of Computer Vision (IJCV)*, 2009. 10

[54] D. Raviv, A. M. Bronstein, M. M. Bronstein, R. Kimmel, and G. Sapiro. Diffusion symmetries of non-rigid shapes. In *Proc. 3D Data Processing, Visualization and Transmission (3DPVT)*, 2010. 10

[55] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: The generalized symmetry transform. 14(2):119–130, March 1995. 3, 8

[56] D. Reisfeld and Y. Yeshurun. Robust detection of facial features by generalized symmetry. pages I:117–120, 1992. 3, 8

[57] A Sheffer. Model simplification for meshing using face clustering. *Computer-Aided Design*, 33(13):925–934, November 2001. 63

[58] G.E.M Jared S.J. Tate. Recognising symmetry in solid models. *COMPUTER-AIDED DESIGN*, 35, 2003. 32, 33, 38, 237

[59] Changming Sun and Jamie Sherrah. 3-d symmetry detection using the extended gaussian image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:164–168, 1997. 29

[60] Susan J. Tate. *Symmetry and Shape Analysis for Assemble-Oriented*. PhD thesis, CRANFIELD UNIVERSITY, 2000. 3, 33, 34, 38, 237

[61] Satoshi Kanai Tomohiro Mizoguchi. Euclidean symmetry detection from scanned meshes based on a conbination of icp and region growing algorithms. In *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Brooklyn, New York, USA, August 3-6 2008. 126

[62] Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. Symmetry hierarchy of man-made objects. *Computer Graphics Forum (Special Issue of Eurographics)*, 30(2):to appear, 2011. 9, 11, 126

[63] Wikipedia. Hausdorff distance. 22

[64] J.D. Wolter, T.C. Woo, and R.A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. 1:37–48, 1985. 20

[65] A. K. C. Wong, S. W. Lu, and M. Rioux. Recognition and shape synthesis of 3-d objects based on attributed hypergraphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:279–290, March 1989. 63

[66] M.K.H.Leung X. Yu and Y. Gao. Hausdorff distance for shape matching. In *Visualization, Imaging, and Image Processing*, Marbella, Spain, 2004. 21

[67] Gang Xu and Wenxian Yang. Fast shape matching using a hybrid model. *Computer Engineering and Technology, International Conference on*, 1:247–251, 2009. 21

[68] Raymond K. K. Yip. A hough transform technique for the detection of reflectional symmetry and skew-symmetry. *Pattern Recogn. Lett.*, 21:117–130, February 2000. 28

[69] Raymond K. K. Yip, Peter Kwong-Shun Tam, and Dennis N. K. Leung. Application of elliptic fourier descriptors to symmetry detection under parallel projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 277–286, 1994. 28

[70] Hagit Zabrodsky, Shmuel Peleg, and David Avnir. Symmetry as a continuous feature. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:1154–1166, December 1995. 25, 26, 27

[71] Xuan Zha. Integration of the step-based assembly model and xml schema with the fuzzy analytic hierarchy process (fahp) for muti-agent based assembly evaluation. *Journal of Intelligent Manufacturing*, 17:527–544, 2006. 10.1007/s10845-006-0031-3. 39

UNIVERSITÉ DE GRENOBLE

Shape Analysis of B-Rep CAD Models to Extract Partial and Global Symmetries

Ke LI

Abstract

Symmetry properties of objects described as B-Rep CAD models are analyzed locally as well as globally through an approach of type divide-and-conquer. The boundary of the object is defined using canonical surfaces frequently used when shaping mechanical components. Then, the first phase consists in generating maximal faces and edges that are independent from the object modelling process but that preserve its symmetry properties. These faces and edges form infinite sets of points that are processed globally. The second phase is the division one that creates candidate symmetry planes and axes attached to the previous maximal edges and faces. Finally, comes the propagation step of these candidate symmetry planes and axes forming the conquer phase that determines the local as well as the global symmetries of the object while characterizing its asymmetric areas.

Key words

Shape analysis, CAD, B-Rep model, Partial and global symmetry detection, Hypergraphs, Divide-Conquer process.

UNIVERSITÉ DE GRENOBLE

Analyse de forme des modèles B-Rep CAO pour extraire des symétries partielles et globales

Ke LI

Résumé

Les propriétés de symétrie d'un objet représenté sous la forme d'un modèle B-Rep CAO sont analysées localement et globalement à travers une approche de type diviser pour conquérir. La surface frontière de l'objet est décrite à partir de surfaces canoniques fréquemment utilisées dans les formes de composants mécaniques. La première phase de l'analyse consiste en la génération de faces et d'arêtes maximales indépendantes du processus de modélisation de l'objet mais préservant ses propriétés de symétrie. Ces faces et arêtes constituent des ensembles infinis de points traités globalement. La seconde phase est l'étape de division consistant en la création de plan et axes de symétrie de candidats pour les faces et arêtes maximales générées précédemment. Enfin, suit l'étape de propagation de ces plans et axes de symétrie représentant la phase de conquête et déterminant les propriétés de symétrie locales et globales de l'objet et caractérisant ses zones non-symétriques.

Mots clés

Analyse de forme, CAO, modèle de B-Rep, Détection de symétrie partielle et globale, Diviser-conquérir processus.