HORIZON 2020

IGA BASED SHAPE REPRESENTATION, INITIAL VERSION

Deliverable D2.2

Circulation:	PU: Public
Lead partner:	SINTEF
Contributing partners:	SINTEF, CNR-IMATI, Fraunhofer IGD
Authors:	Vibeke Skytt, Oliver Barrowclough, Rafael Vazquez, Lorenzo Tamellini, Massimiliano Martinelli, Christian Altenhofen
Quality Controllers:	Daniel Weber, Georg Muntingh
Version:	1.0
Date:	09.06.2016

Consisting of

SINTEF	STIFTELSEN SINTEF, Department of Applied Mathematics, Oslo, Norway
Fraunhofer	Fraunhofer IGD, Interaktive Engineering Technologien, Darmstadt, Germany
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz GmbH DFKI Innovative Factory Systems, Kaiserlautern, Germany
CNR-IMATI	Consiglio Nazionale Delle Ricerche Instituto di Matematica Applicata e Tecnologie Informatiche, Genova & Pavia , Italy
CIMNE	Centre Internacional de Metodes Numerics en Enginyeria Civil Engineering, Barcelona, Spain
ARCTUR	ARCTUR Racunalniski Inzeniring Doo, R&D, Nova Gorica, Slovenia
BOC	BOC ASSET Management GmbH, Innovation Group, Wien, Austria
Missler	Missler Software Missler Software Service Department, Ramonville St Agne, France
Jotne	Jotne EPM Technology AS, Aeronautics, Space and Defense, Oslo, Norway
STAM SRL	STAM SRL, R&D Department, Genova, Italy
TRIMEK SA	TRIMEK SA, R&D, Altube-Zuia (Alava), Spain
Tronrud	Tronrud Engineering AS, 3D Printing, Hønefoss, Norway
NOVATRA	NOVATRA, Varennes Saint Sauveur, France

Document History

Version[1]	Issue Date	Stage	Content and Changes
1.0	09.06.2016	100 %	Ready to be submitted to Project Officer

[1] Integers correspond to submitted versions

EXECUTIVE SUMMARY

CAxMan is a research and innovation project on additive manufacturing (AM) with focus on interoperability between the design, process planning, simulation and dimensional quality steps of the product development lifecycle. The topic of this report is geometric modelling techniques for isogeometric analysis (IGA). Standard CAD models used in industry today are given as a boundary represented solids and they need to be remodelled to fit the requirements of the analysis; in particular, a volumetric representation is required.

Three different geometric representation formats have been selected for use in CAxMan: NURBS surfaces and volumes, locally refined splines (where LR B-splines is the preferred format), and subdivision surfaces and volumes for the design of voids. The NURBS format will be used exclusively in the initial stages of the project, while the other formats will be included at a later stage. Furthermore, the concept of trimming to limit the valid part of a surface or volume, is discussed.

On the basis of the CAxMan use cases and requirements for IGA, different topologies of isogeometric shape models are analysed. A pure block-structured model where each block is represented as a non-trimmed NURBS volume is convenient for analysis, but it is not always feasible to define for the complex shapes and topologies we are interested in using only block-structuring. Thus the project will also consider the use of trimmed models consisting of one block and block-structured models where each of the blocks may be trimmed. The three alternatives are discussed in this report, both with regards to construction and information retrieval. Some details are given on how shape and function space information can be accessed by the analysis software.

Some considerations on interoperability issues involving WP2 are presented, and a minimal and a first version of a comprehensive WP2 workflow are discussed. These issues will be elaborated further throughout the project.

This report serves as a combination of a status report describing the current state of the CAxMan software for creation of isogeometric shape models, but is also a specification on how to proceed with the implementation of methods for model generation for IGA.

TABLE OF CONTENTS

1 INTRODUCTION

CAxMan is a research and innovation project on additive manufacturing (AM) with focus on interoperability between the design, process planning, simulation and dimensional quality steps of the product development lifecycle. The topic of Work Package 2 is analysis-based design. Typically, a CAD model needs remodelling before being suitable for numerical analysis. Task 2.2 (Tri-variate shape representation for design and analysis) is concerned with exactly this remodelling and the main content of this report is related to this process. However, it also covers some aspects of Task 2.1 (Interoperability to CAD) and introductions to Task 2.3 (Subdivision volumes) and Task 2.4 (Analysis tools for design). Finally, the report gives a summary of the interoperability with other work packages.

This report describes the generation of shape models specifically aimed at isogeometric analysis (IGA), in the context of additive manufacturing. The initial configuration, prior to the creation of such models is a CAD geometry described in the STEP format (ISO 10303, 2015). In our context, a CAD model is a boundary represented (B-rep) solid model equipped with tolerances used in the model construction. This model consists of a set of surfaces. So-called elementary surfaces like planes, cylinders, spheres and tori are frequently used, while spline surfaces are used to describe freeform shapes. The pure surface representation may be unbounded or otherwise describe a larger area than desired, so the actual surface area will often be bounded, or trimmed, by curves obtained by intersecting these surfaces. Due to the mathematical difficulty of representing intersections exactly, this process implies the possible existence of small gaps and overlaps between adjacent surfaces. The CAD tolerances specify the legal sizes of such gaps.

The models to be created in WP2 must be volumetric. The same or at least a closely related representation format is used to model both the shape and the analysis results, thus following the so-called isogeometric analysis (IGA) paradigm. NURBS is the most common data format for IGA. In CAxMan, volumetric NURBS will be used for IGA-based shape models, but we will also look into the use of LR B-splines in addition to subdivision surfaces and volumes. A remodelling of a CAD model to produce an isogeometric model will often involve an approximation step. The obtained approximation accuracy will be an outcome of the remodelling process.

This report serves as a combination of a status report describing the current state of the CAxMan software for creation of isogeometric shape models, but is also a specification on how to proceed with the implementation of methods for model generation for IGA.

The CAxMan project partners provide some software libraries as background to the project. SINTEF offers the geometry libraries SISL and GoTools. SISL is SINTEF's spline library. It is a mature library written in C and contains NURBS curves and surfaces and methods to define and operate on these entities. GoTools partly overlaps SISL, but is enhanced with representation formats and algorithms including more geometry types, such as trimmed surfaces, NURBS and LR B-spline surfaces and volumes. Furthermore, it can represent topology structures to hold adjacency information for surfaces and volumes and interfaces to STEP and IGES. GoTools also includes some functionality to create block-structured volumetric models from boundary represented models. This will be discussed in Section 4.1 and 4.2. GoTools is divided into a set of libraries and is written in C++. The libraries have GNU Affero licenses and can be found at https://github.com/SINTEF-Geometry.

IGATools belongs to CNR-IMATI and is an Isogeometric Analysis-based solver. IGATools is a stable and well-maintained software, written in C++14 and available for download under the GNU-GPL licence at https://github.com/igatoolsProject/igatools/wiki. Although IGATools provides basic geometrical functionalities, in this project the geometrical modelling will be

handled by GoTools, and IGATools will "only" provide the analysis functionalities (i.e. computation of the state variables and of the final quantities of interest for the mechanical and thermal analysis). IGATools requires that the geometry is described by a tri-variate spline/NURBS representation, thus the representation format implemented in GoTools can be used as an input to IGATools without preprocessing, provided that a) the geometry itself satisfies some additional requirements, and b) GoTools implements some required functionalities. Both requirements will be discussed in Section 3.

Fraunhofer is developing a library, which is able to model, visualize, import and export subdivision surfaces and volumes. To optimize weight of the printed part, voids and cavities will be added to the model, which will be realized with subdivision volumes. To ensure manifoldness and topological consistency, these subdivision volumes will be designed in a separate step. For integration with the main geometry, the outer surfaces of the subdivision volumes will be approximated as NURBS patches that are then combined with the volumetric representation used for IGA. The approximation algorithm to convert subdivision volumes into a set of NURBS patches will also be part of the library.

Several types of volumetric models are applicable in IGA. We will describe the geometry types relevant for CAxMan in Section 2. Section 3 provides information on the properties that must be satisfied by the shape model in order to be used for isogeometric analysis. Section 4 and 5 responds to these requirements: Section 4 with regard to the topology of the shape model addressing both representation and model generation, and Section 5 with regard on how the shape model information can be accessed by the analysis application. In Section 6, a prerequisite for user guidance is presented, and in Section 7, the CAxMan use cases are discussed with a trivariate shape representation in mind. Section 8 is concerned with interoperability and how information is passed between work packages.

2 SHAPE REPRESENTATIONS FOR SURFACES AND VOLUMES

Isogeometric analysis follows, with some adjustments, the isoparametric approach. In this approach, the analysis domain (that is, the shape model) and the solution field should be described using the same function space. The representation must have entities that are described as a sum of differentiable basis functions:

$$F(u,v,w) = \sum_{i=1}^{N} c_i\, B_i(u,v,w) \text{ or } F(u,v,w) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{L} c_{i,j,k}\, B_i(u)B_j(v)B_k(w).$$

In some context, the function space in which the solution field is sought, might require more degrees of freedom than the shape model. Thus, the solution field might be refined relative to the representation of the geometric shape, but the refined basis can always represent the shape exactly.

In isogeometric analysis, the function space is typically fetched from the geometry field using NURBS surfaces or volumes, but locally refined splines can also be used. In CAxMan, we will also exploit subdivision surfaces and volumes for some features. An analysis domain in 2D is represented by one or more surfaces, while in 3D volumetric entities are required.

2.1 NURBS

NURBS is the de facto standard for freeform curves and surfaces in CAD (Farin, 1999) and is supported by STEP (ISO 10303, 2015). A tensor-product polynomial spline surface is defined as

$$\mathrm{F}(u,v) = \sum_{i=1}^{N}\sum_{j=1}^{M} c_{i,j}\, B_{i,p_1}(u)B_{j,p_2}(v),$$

where $c_{i,j} \in \boldsymbol{R}^d$, *i=1, …, n* and *j=1, …, m*, are spline coefficients and d is the dimension of the geometry space. Normally *d=3*. The basis functions $B_{i,p_1}(u)$ and $B_{j,p_2}(v)$ are univariate B-splines with polynomial degree p_1 and p_2 in the first and second parameter direction, respectively. A B-spline surface is a piecewise polynomial surface and in each parameter direction the joints between the polynomial pieces are defined by a knot vector. The tensor-product structure implies that the surface is defined on a regular grid generated by these knot vectors.

The B-splines are piecewise polynomials, have limited support, are non-negative and the sum of all B-splines at any parameter value is one (partition of unity). The set of B-splines is linearly independent. The continuity of the B-splines is defined from the polynomial degree and the multiplicity with which each knot appears in the knot vector. For knots with multiplicity one, the continuity is equal to the polynomial degree minus one.

Frequently used surfaces in CAD modelling like spheres, cylinders, cones and tori cannot be exactly represented by polynomial B-spline surfaces. The same applies to rotational surfaces. However, NURBS offers an exact representation of these types of surfaces. A NURBS surface is defined as

$$F(u,v) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} c_{i,j}\, h_{i,j} B_{i,p_1}(u)B_{j,p_2}(v)}{\sum_{i=1}^{N}\sum_{j=1}^{M} h_{i,j}\, B_{i,p_1}(u)B_{j,p_2}(v)},$$

where the weights $h_{i,j}$ are non-negative real numbers**.** NURBS offers an increased flexibility compared to polynomial splines, but the added freedom is mostly used for the representation of conic curves and surfaces and for surfaces joining to a NURBS surface. The complexity of the expression makes it hard to use the full flexibility of the weights in design. Furthermore, the

rational structure of the formula implies that addition and differentiation involving NURBS surfaces increases the polynomial degree.

Spline volumes are defined similarly. A polynomial spline volume is defined as

$$F(u,v,w) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{L} c_{i,j,k}\, B_{i,p_1}(u) B_{j,p_2}(v) B_{k,p_3}(w),$$

and a rational NURBS volume as

$$F(u,v,w) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{L} c_{i,j,k}\, h_{i,j,k} B_{i,p_1}(u) B_{j,p_2}(v) B_{k,p_3}(w)}{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{L} h_{i,j,k}\, B_{i,p_1}(u) B_{j,p_2}(v) B_{k,p_3}(w)}$$

The properties of spline surfaces carry over to the volumes, and spline volumes are also included in the STEP format.

In the context of IGA, it is important that the B-splines have local support. The number of degrees of freedom for a spline entity can be increased by knot insertion or degree elevation without changing the geometry. The tensor product structure of the polynomial B-spline surfaces and volumes facilitates efficient grid evaluation by evaluating the univariate B-spline basis function only once for each parameter value and combining appropriate B-splines to obtain the result in the grid values. For rational splines, the weights are defined by the geometry model and kept constant during the analysis. A rational spline surface can, thus, be expressed as

$$F(u,v) = \sum_{i=1}^{N} \sum_{j=1}^{M} c_{i,j} R_{i,j}(u,v)\,,$$

where

$$R_{i,j}(u,v) = \frac{h_{i,j} B_{i,p_1}(u) B_{j,p_2}(v)}{\sum_{i=1}^{N} \sum_{j=1}^{M} h_{i,j}\, B_{i,p_1}(u) B_{j,p_2}(v)}.$$

This construction implies that rational spline entities can be used in the same way as polynomial ones in isogeometric analysis, but the computations become less effective. Thus, unless it is important to maintain the exact shape of for instance cylindrical surfaces, polynomial splines should be the preferred choice.

2.2 LOCALLY REFINABLE SPLINES: LR B-SPLINES

The knot vectors of spline surfaces and volumes are global, in the sense that inserting a knot in one of the variable directions results in a refinement over the entire domain in all other variables. Thus, truly local knot insertion is not possible. Several initiatives have been made to remedy this problem, for instance T-splines (Sederberg T. W., 2003) and hierarchical B-splines (Forsey D. R., 2003). We will focus on LR B-splines, which was introduced in (Dokken T., 2013).

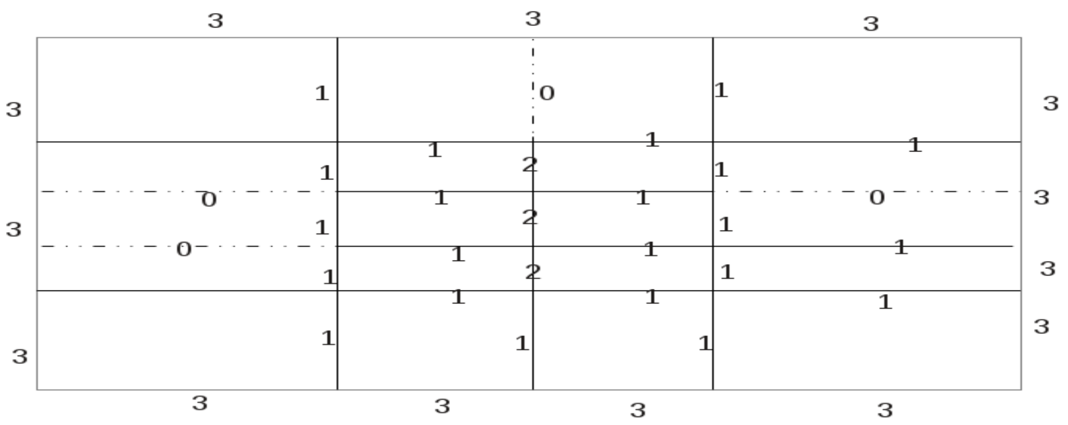

FIGURE 1: THE PARAMETER DOMAIN CORRESPONDING TO AN LR B-SPLINE SURFACE OF DEGREE 2 IN BOTH PARAMETER DIRECTIONS

An LR B-spline surface or volume is defined on a regular domain in 2D or 3D, respectively. The domain is composed of regular boxes, which, in contrast, do not define a regular grid. Figure 1 illustrates the parameter domain for a surface. The numbers indicate the multiplicity of a knot line. The multiplicity may vary throughout the domain and zero multiplicity is allowed. A knot line having multiplicity zero in some intervals, is denoted as local.

An LR B-spline surface or volume is typically created from a tensor product B-spline surface or volume by inserting new knot lines. The knot lines can be defined from the need of representing local details in the model, refining the entity close to boundaries to enable exact continuity of some degree towards adjacent entities in a multi block setting, see Section 3.2, or allowing for adaptive refinement in isogeometric analysis. A new knot line is required to split the domain of at least one basis function and all basis functions with a domain being completely divided by this knot line will be split and represented by two new basis functions defined on a reduced domain.

An LR B-spline surface is defined as

$$F(u,v) = \sum_{i=1}^{N} s_i c_i B_{i,p_1 p_2}(u,v)$$

and an LR B-spline volume is defined by a similar expression (but with an extra variable w). The bi-variate B-splines, $B_{i,p_1 p_2}(u,v)$, are composed by two univariate B-splines similar to the tensor-product setting, but in contrast to the tensor-product case, the B-splines do not need to relate to all knots in their domain. The tri-variate B-splines appearing in the volumetric entity are composed of three univariate B-splines. Most of the properties from tensor product splines carry over to LR B-spline surfaces and volumes: the B-splines have limited support, are non-negative and the scaling factors s_i ensure partition of unity. The size of the B-splines varies throughout the domain and the number of B-splines defined over each polynomial patch varies as well. The continuity is defined from the polynomial degree and the knot multiplicity. The B-splines are not always guaranteed to be linearly independent, but guidelines for constructions resulting in linear independence exist and this property can also be checked for by the so called peeling algorithm. An LR B-spline entity with linearly dependent B-splines can be refined such that the dependency is removed.

Similar to the tensor product splines, an LR B-spline surface or volume can be rational. A tensor product spline surface can be seen as an LR B-spline surface where all knot lines are global and

have the same multiplicity throughout the domain. The STEP format has recently been extended with locally refinable spline surfaces and volumes (ISO 10303, 2015).

2.3 TRIMMED SURFACES AND VOLUMES

Tensor-product spline entities and LR B-splines are defined on rectangular domains while some CAD surfaces like planes, cylinders and tori have an infinite size. To restrict the extent of a surface, or a volume, to the wanted area, trimming is applied.

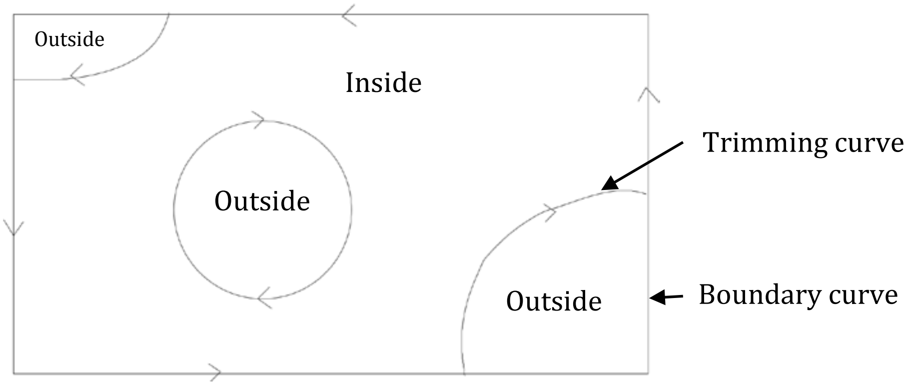

FIGURE 2: THE PARAMETER DOMAIN OF A TRIMMED SURFACE

The valid part of a trimmed surface is limited by curve loops. For a parametric surface, these loops can be represented in the parameter domain, the geometry domain or both. The orientation of the trimming loop defines which part of the total surface is inside the trimmed surface, see Figure 2. The trimming curves will often be obtained through a Boolean operation, i.e. intersection between surfaces, and it will often not be possible to represent the intersection curves exactly. A trimming curve may be represented by three approximations of the exact intersection curve, one curve in geometry space and one curve in each of the parameter spaces of the two surfaces being intersected. The curves are approximated within some tolerance, but they are not exactly equal. Thus, trimming will frequently create small gaps and overlaps. They are handled within the system where the geometry was defined, but may create problems in subsequent operations, such as isogeometric analysis.

Surfaces are trimmed by sequences of curves and curves can have a consistent sequence and orientation. Thus, surfaces can be trimmed by purely geometric entities or one may choose to use topological entities such as edges. Representation of trimmed volumes necessarily involves topological entities. A volume is trimmed by a closed set of surfaces, often denoted as a closed shell. A volume with voids will be trimmed by one outer and one or more inner closed shells. An example of a data structure that is able to represent a trimmed volume is shown in Figure 19.

2.4 SUBDIVISION SURFACES AND VOLUMES

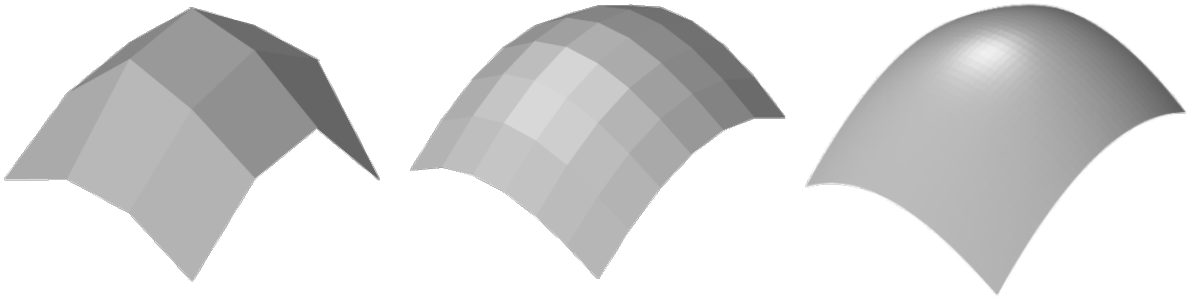

FIGURE 3: ITERATIVE REFINEMENT OF A CATMULL-CLARK SUBDIVISION SURFACE

In computer graphics, subdivision surfaces are a well-established approach to creating smooth and continuous surfaces from a polygon control mesh. Common subdivision algorithms are Catmull-Clark subdivision surfaces (Catmull E., 1978), Loop subdivision (Loop, 1987) or $\sqrt{3}$-subdivision (Kobbelt, 2000). Similar to NURBS or B-Splines, the shape of the surface can be controlled by a relatively low number of control vertices. In contrast to spline-based representations, subdivision surfaces are generated by iteratively refining the control mesh using a certain subdivision rule-set. Theoretically applying this subdivision step an infinite number of times, a so-called "limit surface" will be reached. Depending on the subdivision scheme, this "limit surface" can also be described mathematically and then evaluated directly at any sample point.

Subdivision volumes transfer the concept of subdivision surfaces to the next dimension. Instead of a surface mesh, subdivision volumes are described by volumetric control mesh. In each iteration, cells, faces and edges are subdivided and new elements are created accordingly. This results in a manifold, watertight and consistent "limit volume" with a smooth outer surface. Some algorithms for volumetric subdivision already exist, like Catmull-Clark subdivision solids (Joy K. I., 1999), tetrahedral subdivision by Schaefer et al. (Schaefer S., 2004) or an algorithm by Chang et al. (Chang Y.-S., 2003). Similar to subdivision surfaces, every approach has different requirements regarding the quality, topology and regularity of the control mesh.

In the CAxMan project, subdivision volumes will be used to add voids and cavities to the model in order to optimize its inner structures for additive manufacturing. As a starting point, the refinement rules of Catmull-Clark subdivision solids will be implemented, since they are the most versatile ones regarding the requirements mentioned above and therefore allow the most freedom in designing and manipulating the control mesh for the voids. The rules are as follows:

1. For each cell, add a cell point at its centroid.
2. For each face, add a face point as a weighted average of the two incident cell points and the face's centroid.
3. For each edge, add an edge point as a weighted average of all incident cell points, all incident face points and the edge's midpoint.
4. For each cell, connect its cell point to all its face points.
5. For each face, connect its face point to all incident edge points.
6. For each original vertex (which is not a cell/edge/face point), set its position as a weighted average of all incident cell points, all incident face points, all incident edge points and the vertex's old position.

A more detailed description can be found in (Joy K. I., 1999).

3 REQUIREMENTS FOR IGA BASED ANALYSIS

In this section we describe the criteria that the geometrical description must meet for a successful IGA-based analysis (Hughes, Cottrell, & Bazilevs, 2005), as well as the functionalities that GoTools, or any other software for geometry representation, must provide to be used together with IGATools.

Two different approaches for geometry representation will be considered: block-structured models, which consist of several NURBS blocks, and trimmed models, which consist of a volumetric spline and a number of closed surfaces or surface sets that trim it. The details about the geometry representations used for these models were given in Section 2, while the required topology constructions will be discussed in Section 4; we focus for now on the requirements to perform the IGA based analysis. This section is divided into three parts, which explain the three different types of required quantities and capabilities: I) the ones required to perform IGA analysis in the most simple (single block) model; II) the ones required for the analysis of a multiblock model; III) the ones required for the analysis of a trimmed model.

3.1 GENERAL REQUIREMENTS FOR IGA

Most of these requirements are related to one of the fundamental tasks of IGA that is performed by IGATools, i.e. approximation of integrals (this operation will be referred to as “quadrature” in the rest of this document). Indeed, one of the core steps of IGATools (and of IGA-based solvers in general) is to compute quantities such as:

$$\int_\Omega B_i^{ph}(x,y,z)B_j^{ph}(x,y,z)dxdydz, \qquad \int_\Omega \nabla B_i^{ph}(x,y,z)\cdot\nabla B_j^{ph}(x,y,z)dxdydz$$

where Ω is the physical domain (or a portion of it), $[x,y,z]$ is a point in the physical space that corresponds to the point $[u,v,w]$ in the parametric space via the splines model of the geometry, i.e. $[x,y,z]=F(u,v,w)$ where F is a NURBS or LR B-spline volume as detailed in the previous section. Writing F^{-1} for the inverse mapping from the physical space to the parametric space, i.e. $[u,v,w]=F^{-1}(x,y,z)$, $B_i^{ph}(x,y,z)=B_i(F^{-1}(x,y,z))$ is a generic function of the spline basis in the physical domain, obtained by mapping of the basis function $B_i(u,v,w)$ in the parameter domain with F, see Figure 4, and $\nabla B_i^{ph}(x,y,z)$ denotes its gradient.

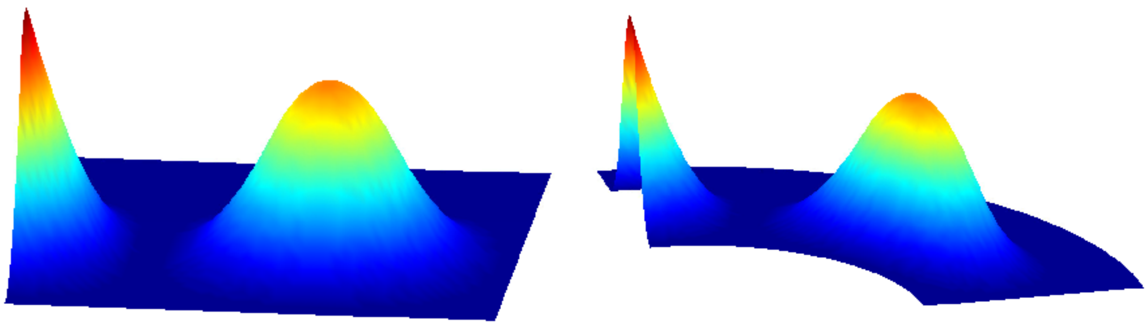

FIGURE 4: TWO BASIS FUNCTIONS IN THE PARAMETER DOMAIN AND MAPPED TO THE PHYSICAL DOMAIN, HERE SHOWN IN THE BIVARIATE CASE FOR SIMPLICITY

These integrals are usually computed by numerical integration, sampling the values of B_i^{ph} and B_j^{ph} (and their derivatives) over a set of K points in Ω, $[x_1, y_1, z_1],[x_2, y_2, z_2],\ldots,[x_K, y_K, z_K]$ and multiplying each evaluation by suitable quadrature weights w_k, namely

$$\int_\Omega B_i^{ph}(x,y,z)B_j^{ph}(x,y,z)dxdydz \approx \sum_{k=1}^{K} B_i^{ph}(x_k, y_k, z_k)B_j^{ph}(x_k, y_k, z_k)w_k$$

$$\int_\Omega \nabla B_i^{ph}(x,y,z)\nabla B_j^{ph}(x,y,z)dxdydz \approx \sum_{k=1}^{K} \nabla B_i^{ph}(x_k, y_k, z_k)\nabla B_j^{ph}(x_k, y_k, z_k)w_k$$

The points $[x_k, y_k, z_k]$ are usually placed on a Cartesian grid in the parametric/reference domain, which is the unit cube, and then shifted to Ω by the mapping F, see Figure 5. The derivatives of F are also required for the computation of the quadrature weights, and for the evaluation of the function derivatives in the physical domain.

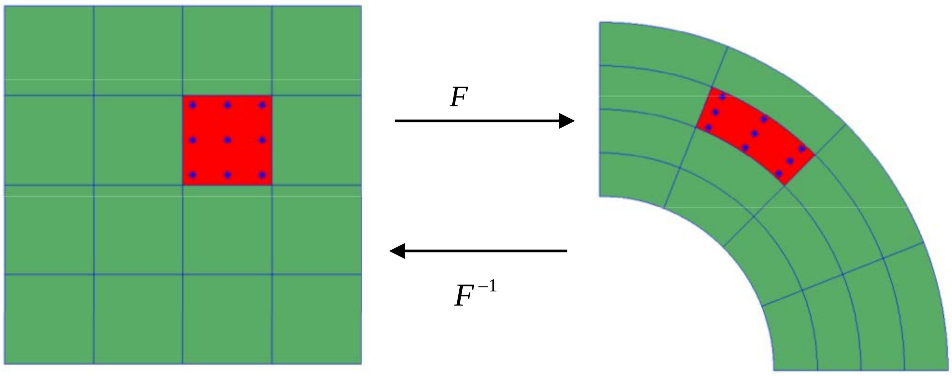

FIGURE 5: DEFINITION OF THE QUADRATURE RULE IN THE PARAMETER DOMAIN AND MAPPING TO THE PHYSICAL DOMAIN, HERE SHOWN IN THE BIVARIATE CASE FOR SIMPLICITY

Thus, IGATools needs GoTools to provide the following functionalities:

- Given a point $[u, v, w]$ in the unit cube, evaluate the mapping $F(u, v, w)$ (i.e. establish the location in the physical domain of a point in the parameter domain) and its derivatives (Jacobian) $\frac{d}{du}F, \frac{d}{dv}F, \frac{d}{dw}F$.
- Conversely, given a point $[x, y, z]$ in the physical domain, compute its parametric coordinates, $[u, v, w] = F^{-1}(x, y, z)$.
- Evaluate basis functions and their gradient at a point $[u, v, w]$ in the parameter domain, $B(u, v, w)$, $\nabla B(u, v, w)$.
- For each point in the parameter domain, list the basis functions that are non-zero at that point. In fact, if the quadrature points are grouped by the elements given by the knot vectors of the splines, the list can be given for each element. In IGA, and also in FEM, this is usually called the connectivity of the degrees of freedom.

Another critical step is related to enforcing boundary conditions on the model. To do this, integrals similar to those written earlier must be computed, this time on the surfaces describing the boundary of the computational domain. The boundary may be described by several mappings $F^k(u, v)$ (typically, one per boundary) from a 2D reference/parameter domain that

can be seen as a restriction of the original 3D domain to one of its bounding faces, on which it is quite easy in practice to define a quadrature rule (a Cartesian grid and the quadrature points and weights).

More precisely, IGATools needs GoTools to provide the following functionalities:

- Evaluate the boundary mappings $F^k(u,v,w)$ and their derivatives (Jacobians)

$$\frac{d}{du}F^k, \frac{d}{dv}F^k, \frac{d}{dw}F^k.$$

- Given a point $[x,y,z]$ on the k-th boundary, compute its parametric coordinates

$$[u,v,w] = (F^k)^{-1}(x,y,z).$$

- Compute the outward normal of each boundary, at any given point.
- For each boundary element, list the (volumetric) basis functions that do not vanish on the element. For tensor-product splines this can be computed inside IGATools, but for more general functions, such as LR B-splines, the information must be given by GoTools.

All the functionalities above are required to perform the IGA analysis of a given model, computing the solution of our problem with the same set of functions that is used in the CAD model to define the geometry. However, in most of the cases the analysis will require h- or p-refinement operations (i.e., knot insertion or degree elevation) to use finer discrete spaces, to obtain more accurate results. Thus, IGATools also needs GoTools to provide the following functionalities:

- A refinement routine to insert new knots in the tensor-product space.
- A refinement routine to refine the mesh locally, using LR B-splines. Local refinement will be considered at a later stage of the project.
- A refinement routine to increase the degree of a spline.

3.2 BLOCK-STRUCTURED MODELS

IGATools already supports block-structured ("multipatch") models, formed by several NURBS blocks. These models can be non-conforming and non-nested, in the sense that the elements (defined by the knot vector) given for two different blocks do not necessarily have to match on the interface. However, we point out that in the general non-nested case in which the two sides of the interface have different tessellations it is necessary to create a new tessellation as the intersection of both. IGATools currently relies on third-party software to create this tessellation intersection.

The requirements that must be fulfilled by the block-structured description to be successfully used in IGA were already listed in Section 8 of the deliverable D2.1. These include that the blocks must meet in a corner-to-corner configuration, the number of blocks should be kept to a minimum, and extraordinary points or singularities should be avoided in areas critical for the analysis (for instance, interfaces should not be placed in the middle of a high-stress region, or of a high temperature gradient). Clearly, this is a problem-dependent issue and some iterations between the geometry creation and the analysis are envisaged. User involvement, as discussed in Section 4 and 6, may be required to enable good choices for the position of singularities.

For the analysis of block-structured models we require, for each block, the information already mentioned in Section 3.1. Besides this, some additional functionality is needed to perform numerical integration on the interface shared by two blocks. Since the integrals may involve basis functions coming from the two blocks, given a quadrature point it is necessary to identify its parametric coordinates in both domains. We require:

- For each shared interface, the labels of the two blocks and of the face of each block that match at the interface.
- Given a point on the boundary of one block, identify its parametric coordinates in the other block. In general, this can be done as in the previous section, but for matching blocks one can take advantage of the relative positions of the two blocks to simplify the computation.

3.3 TRIMMED MODELS

While multipatch capabilities to deal with block-structured models are already present in IGATools, handling general trimmed volumes is still not possible. So far, only 2D and 2.5-D (extrusion, revolution, etc.) geometries are allowed, and actually implementing such functionalities is one of the main goals of this project for CNR-IMATI.

Let us assume that we have a parametrized volume, which we denote by Ω, and for simplicity a single closed surface Σ that trims the volume. Thus, the surface divides the volume in two different regions: the region occupied by the modelled object (also called active region and denoted by Ω_a), and the trimmed or inactive region (denoted by $\Omega_i = \Omega \setminus \Omega_a$). To perform the isogeometric analysis in a trimmed model, it is necessary to compute integrals, as in the previous section, only in the active region. However, it is not yet possible to define the quadrature points in a Cartesian grid of the parameter domain, because the mapped elements will not be entirely in the active region. There exist different approaches to deal with trimmed objects in IGA, and for its easiness and versatility we have chosen to use the Finite Cell Method (Schillinger, et al., 2012). The idea is to approximate the integrals in the following way

$$\int_{\Omega_a} f(x,y,z)dxdydz \approx \int_{\Omega} \alpha(x,y,z) f(x,y,z)dxdydz$$

where the coefficient α is chosen to be equal to one in the active region, and very close to zero in the trimmed region.

Since the domain of the integral is now Ω, we can define a quadrature rule in a Cartesian grid, as we did before. However, the coefficient α has a strong discontinuity, and it is necessary to use more accurate quadrature to obtain reliable results. This is done using *adaptive quadrature,* as illustrated in Figure 6. For each element of the grid we have to identify whether the element is trimmed or not. If it is trimmed, we subdivide it by bisection in four new elements (eight in the 3D case), and keep applying the same bisection algorithm until a suitable stopping criterion is met (typically involving either the size of the quadrature element or some kind of a posteriori quadrature error indicator). We remark that this refinement is only applied to compute the quadrature, but the discrete space and the basis functions to compute remain the same.

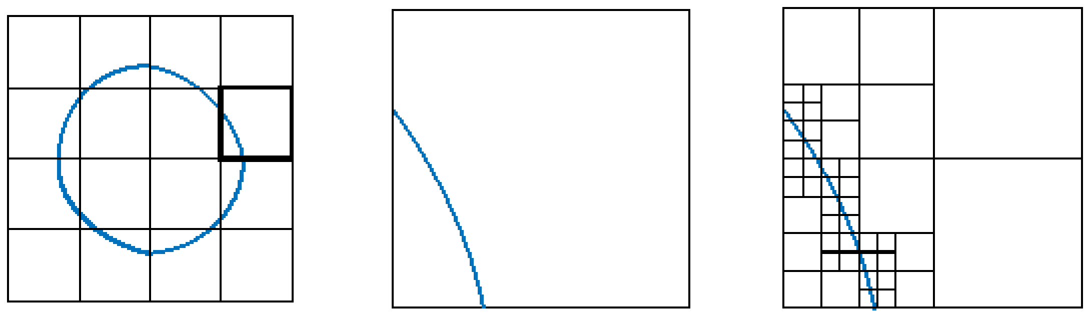

FIGURE 6: DETAIL OF THE CONSTRUCTION OF AN ADAPTIVE QUADRATURE MESH WITH FOUR LEVELS

Adaptive quadrature can be implemented using a quadtree/octree structure, where each node of the tree represents one element. The starting point is the set of elements determined by the knot vector, which correspond to the zero-th level of the tree, and new levels are added to the tree when trimmed elements are subdivided. The octree structure will be implemented in IGATools, but the information about whether an element is trimmed or not must be available from GoTools. More precisely, GoTools should provide the following functionality:

- Given an element in the parameter domain, tell whether its image in Ω is in the active region, in the inactive region, or on the trimmed entity. The element is not necessarily given by the knot vector, particularly for high levels in the tree.
- Given a quadrature point $[x, y, z] \in \Omega$, tell whether it is in the active or in the inactive region.

In general, for a trimmed model it will also be necessary to impose boundary conditions on the trimming surface Σ. The required information is almost identical to the one already mentioned in Section 3.2, with the difference that the trimming surface Σ is not a boundary of Ω, but of Ω_a. Therefore, the quadrature rule to be defined on Σ cannot be induced by the one in the tri-variate model, and it is necessary to define a grid and the quadrature points on Σ to perform the numerical quadrature. The evaluation of the basis functions at these points requires knowledge about their position in the reference domain of Ω. For a purely trimmed model, as the one described in Section 4.3, this does not represent a problem. However, when combining block-structured and trimmed models as in Section 4.4, it is necessary to compute the inverse F^{-1} of the parametrization.

4 SHAPE MODELS FOR ISOGEOMETRIC ANALYSIS

It is rarely possible to represent shape models for isogeometric analysis with one boundary fitted NURBS volume (block). As seen in Section 3, we focus on block-structured models consisting of a set of boundary fitted NURBS blocks and trimmed models containing one trimmed NURBS block. We also envisage the possibility of combining the two as explained in Section 4.4 and the use of locally refined splines and subdivision surfaces/volumes. In the following, we will go into some detail on how the isogeometric shape models are or may be constructed.

4.1 BLOCK-STRUCTURED MODELS

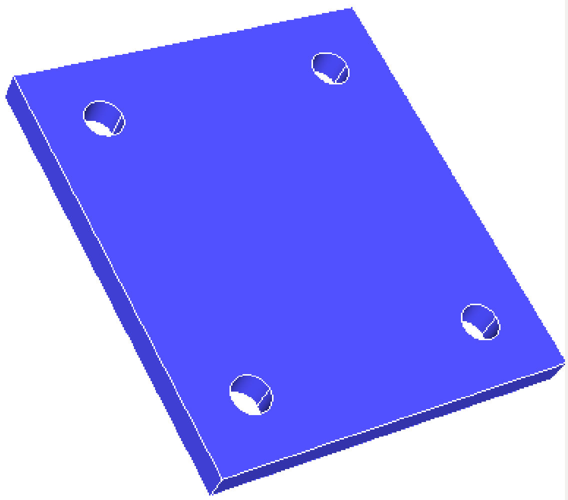

FIGURE 7: BOUNDARY REPRESENTED CAD MODEL

A block-structured isogeometric shape model consists of a number of blocks represented as NURBS surfaces or volumes, depending on the dimension of the model space. This type of model represents a convenient format for isogeometric analysis. GoTools provides some functionality to automatically create such models from boundary represented CAD geometries, but this functionality is limited to certain model types. The produced models have a corner-to-corner configuration, i.e. T-joints are not allowed, and adjacent blocks share common spline spaces. An approximation step is normally required in the translation from the initial CAD model.

We now present a simple volumetric block-structuring example in some detail, highlighting decisions made by the algorithm implemented in GoTools. We also look at how the need for automatic decisions limits the set of CAD models that can be transformed by this approach. We illustrate how the complexity of the decision making rapidly increases with the model complexity.

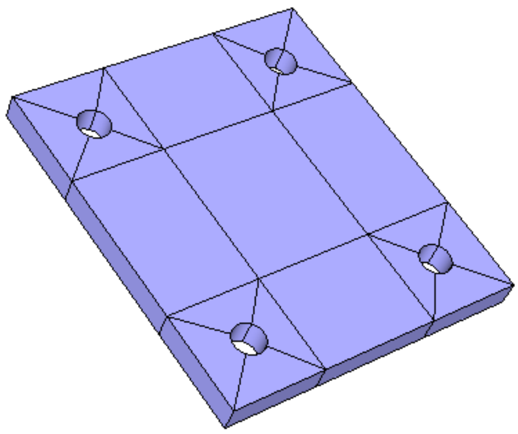

FIGURE 8: BLOCK-STRUCTURING OF THE MODEL BOUNDARY

Figure 7 shows a simple CAD model representing a plate with holes. The model surfaces are trimmed planes and cylinders. The white lines in the cylinder surfaces show the position of an opening of angle zero in the surface description, which will later be referenced as the seam of the closed surface.

The first step is to perform block-structuring of the model boundary as shown in Figure 8. The large planar surfaces are divided to separate the holes and further split from corners in the

resulting subsurfaces to the holes. In the final tri-variate model these surface blocks will become boundary surfaces to a set of volume blocks. This is only feasible if the surface blocks are consistent for "opposite sides" of the model. In this case the "opposite sides" are trivially recognized and consistency is achieved by applying the same rule for the planar surfaces representing these sides. This will not always be the case.

Due to the distance between the holes in Figure 8, the block-structuring algorithm decided on an additional block between the holes to maintain a reasonably regular shape for the surface blocks. Also in Figure 9 the opposite sides are easy to recognize, but the distances between the holes vary for the two main surfaces. The configuration on one side (left) indicates that the holes should be split by only a block boundary, but for the configuration on the other side (right) a similar decision as for the previous case would be natural. To get a consistent starting ground for the volumetric block-structuring, the two opposite surfaces need to be divided using the same pattern. This may be ensured by identifying the surfaces as opposite boundary surfaces for the final model. This identification is trivial for the models in Figure 7 and Figure 9, due to the dominant planar surfaces. However, the model does not need to be very complex before the identification of opposite surfaces is not trivial. Consider, for instance, the model in Figure 10. It is still crucial to perform a consistent splitting between the top and the bottom of the model, but now one of the planar faces has been divided into several pieces. There is no longer a one-to-one match between opposite surfaces.

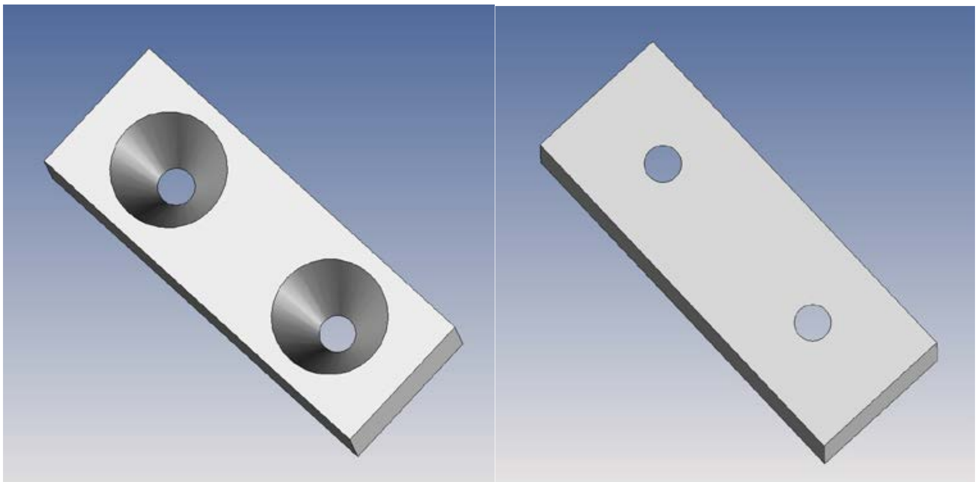

FIGURE 11: MODEL WITH TWO CONICAL HOLES

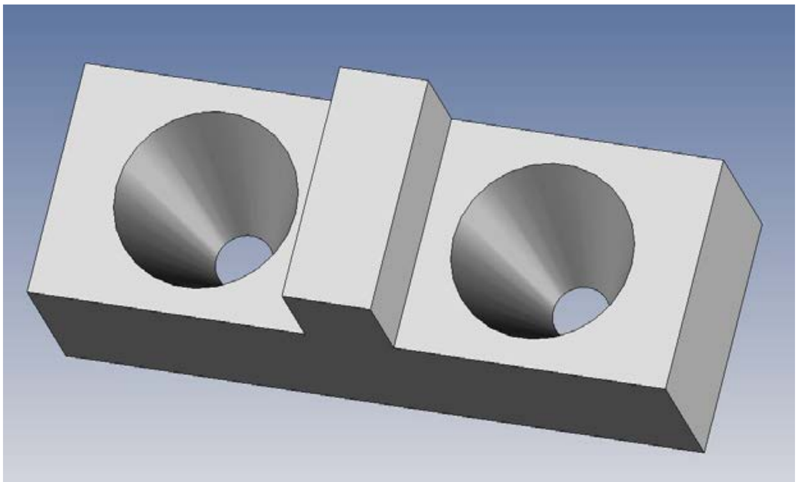

FIGURE 10: ADDED COMPLEXITY TO THE MODEL IN FIGURE 9

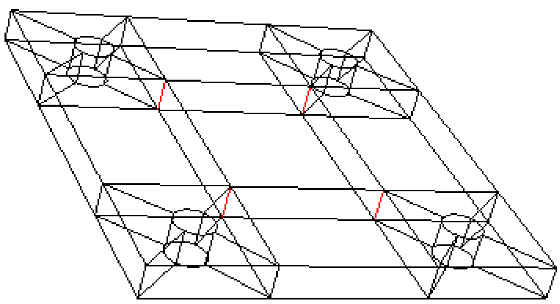

FIGURE 9: SURFACE BLOCK BOUNDARIES CORRESPONDING TO THE MODEL IN FIGURE 7

Returning to the model in Figure 7, we can study the surface blocks at the volume boundaries further in Figure 11. The boundaries of the surface blocks generated from the CAD model are shown in black, but we can see that they do not provide sufficient information to create volume blocks. The structure must be complemented with edges of volume blocks in the interior of the object, i.e. connections between surface block corners at the boundaries must be defined. They are visualized as red lines in Figure 11.

The curves corresponding to the seams of the cylindrical surfaces describing the holes in Figure 7 have disappeared in Figure 11. Instead, surface boundaries better adapted to the block-structuring configuration have been introduced. This modification includes merging two existing surfaces into one larger surface. The decision on which surfaces to merge is taken by the algorithm, but for more complex situations user involvement might be beneficial. Consider e.g. Figure 12; here, one version of the TERRIFIC demonstrator part is shown emphasizing the surface boundaries of the CAD model. The model is composed of a set of surfaces that provides a difficult starting ground for a block-structuring algorithm. If several surfaces at the external blends are viewed as one unit in the process of creating surface blocks, the actual design of the part is easier to recognize.

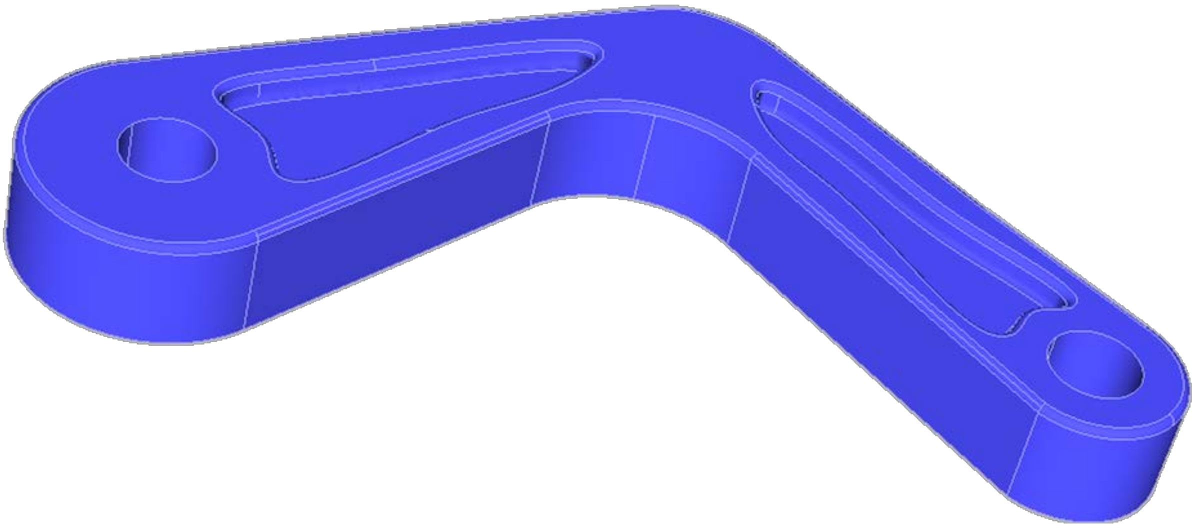

FIGURE 12: THE TERRIFIC DEMONSTRATOR PART WITH EXTERNAL AND INTERNAL BLENDS

A volumetric block-structured version of the model in Figure 7 is shown in Figure 13 (left). In Figure 13 (right), the spline spaces of the volume blocks are refined to create corresponding spline spaces at block boundaries. The operation increases the data size, but enables a simple approach to ensure exact C^0 continuity between the spline blocks. For this model an exact reproduction of the CAD model is possible using NURBS volumes, but currently polynomial spline volumes are used. In other cases, an approximation step cannot be avoided.

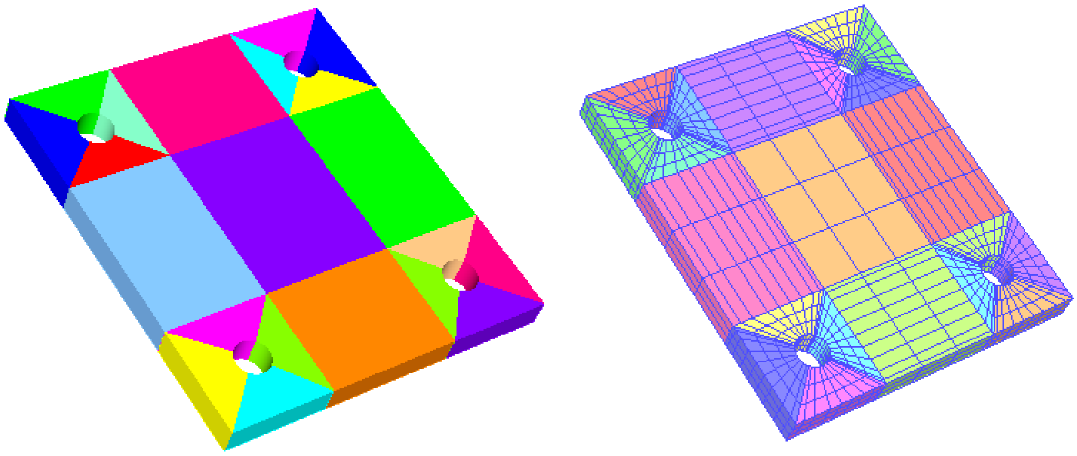

FIGURE 13: VOLUMETRIC BLOCK-STRUCTURED VERSION OF THE MODEL SHOWN IN FIGURE 7

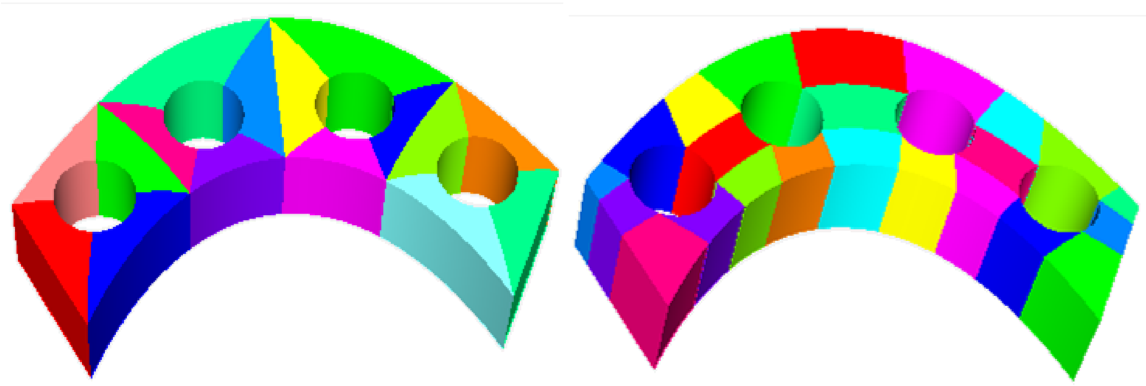

FIGURE 14: TWO VERSIONS OF BLOCK-STRUCTURED MODELS CREATED FROM THE SAME BOUNDARY REPRESENTED CAD MODEL

In general, there is not one unique solution as to how a B-rep CAD model can be translated into a block-structured volume model. Consider Figure 14; here, the same CAD model has been turned into two different block-structured models, both having their pros and cons.

4.1.1 User interaction

Analyzing the block-structuring approach presented above, we can identify the following points where user interaction through a graphical user interface may simplify the decision making process and provide a better block-structured model:

- Identification of pairs of surface groups where the block-structuring of the outer boundary should correspond, see Figure 10.
- Positioning of end points of curves representing surface block boundaries at the model boundary. User involvement in this context is especially important if none of these end points exist already as vertices in the CAD model. An example can be seen in Figure 14 (right). Both block-structuring solutions are feasible, but the outermost blocks in the model to the right could have a more regular shape.
- Specify positions of singularities or extraordinary points for the surface blocks at the outer boundary. This can provide better block geometry, but no less important is the need to avoid extraordinary points in critical areas for the analysis, see Section 3.2.
- When the volume blocks are created, some vertices in the block-structured outer surface should be joined by volume block edges. The identification of these vertices may be made more robust with the help of user involvement, see Figure 11.
- Identification of surfaces in the CAD model that should be handled as one entity in the block-structuring process (surfaces to be merged), see Figure 12.

Most of these identifications can be performed prior to the initiation of the block-structuring process, but some decisions must wait until a block-structuring of the outer boundary exists.

4.2 SWEPT BLOCK-STRUCTURED MODELS

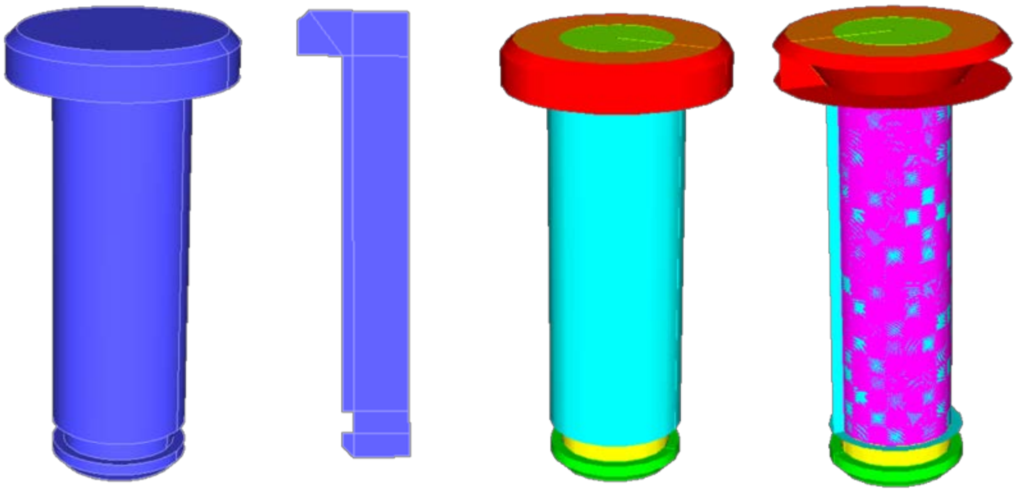

FIGURE 15: BLOCK-STRUCTURING OF A ROTATIONAL CAD MODEL

Some CAD models have a particularly simple format, which can be utilized in the creation of multi-block tri-variate spline models. These are the swept models. The part in Figure 7 is a plate with four cylindrical holes. The part is identical throughout the thickness direction and can, provided that this configuration is recognized, be block-structured in the plane and the volume blocks can be created by linear sweep.

Rotational models can also be recognized and block-structured in the plane. Figure 15 shows one such case. The model (left picture) consists entirely of cylindrical surfaces, cones and planes

trimmed with circular curves. All the rotational entities have the same axis and the normals of the planar surfaces are parallel to this axis. An intersection of the model with a plane through this axis results in a trimmed planar surface. Extracting the part of this surface lying at one side of the axis, we get the surface in the second picture (from the left), which can be block-structured as indicated by the white curves. Rotating the surface blocks around the axis gives the tri-variate block-structured model shown in the third picture (second from the right). The rightmost picture gives some insight into the interior of the block-structured model. Here, only the inner boundary surfaces of the outer volume block are shown. The chess like pattern of one of the surfaces indicates that the inner surface of the outer block coincides with the outer surface of the inner block.

4.3 TRIMMED VOLUMETRIC SPLINE MODELS

A trimmed volumetric spline model aimed at isogeometric analysis consists of one spline volume and a number of trimming surfaces configured into one or more closed surface sets or shells. An intersection between shells is not allowed, and if more than one shell exists, the outer shell represents the boundary of the trimmed volume while the others represent voids.

For these models, not all trimming surfaces will fit the volume boundaries. However, as boundary fitted trimming surfaces offer better analysis accuracy without additional refinements, this property should be aimed at whenever it is achievable. Boundary fitting is relevant only for trimming surfaces belonging to an outer shell. The trimming shell will probably contain some gaps and overlaps: this topic will be discussed in Section 4.5.

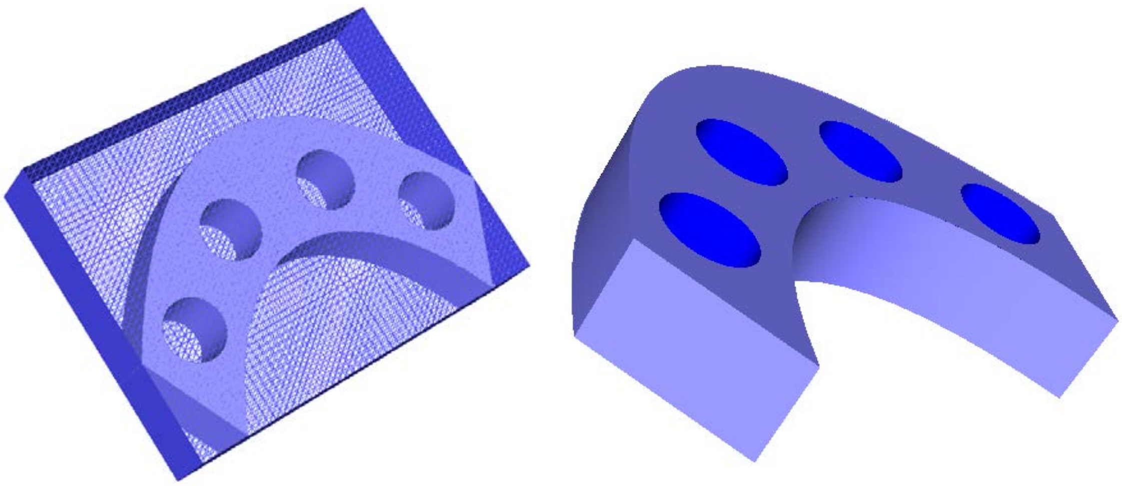

FIGURE 16: TRIMMING APPROACHES FOR THE CAD MODEL CORRESPONDING TO THE BLOCK-STRUCTURED MODELS IN FIGURE 14

Performing isogeometric analysis with trimmed geometry models requires a tight coupling between the geometry representation and the analysis software. This is achieved by integrating GoTools into IGATools. The geometry model will be represented as a GoTools entity while the analysis is performed in IGATools. An API interface for this geometry model is provided in Section 5.1.2.

Consider the CAD solid shown in Figure 16. Two versions of block-structured models corresponding to it are shown in Figure 14, but the model can also be represented as a trimmed spline volume. The simplest approach is shown in the left picture of Figure 16. The bounding box of the model is used to define a spline volume and the initial CAD model defines the trimming

shell. A better solution is visualized in the right picture of the figure. The large CAD surfaces highlighted are selected as boundary surfaces for the underlying NURBS volume. Trimming information related to the holes in the surfaces is omitted in this construction, but are included in the trimming shell. But how do we define which initial surfaces to use as boundary surfaces?

One spline volume has six boundary surfaces. This implies that at most six surfaces or smooth combinations of surfaces can be selected as boundary surfaces. These surfaces should have one or more of the following characteristics:

- The surface is large.
- The surface has outer trimming curves and/or holes. This indicates that it has been defined early in the CAD design process and has been subject to Boolean operations. Thus, it is a carrier of design intent.
- The surface is an elementary surface (plane, cylinders, cones, etc.). For non-planar surfaces, the radius should be large compared to the model size.
- The surface is an underlying surface for more than one trimmed surface.
- The surface or surface set is selected by the user.

4.3.1 User interaction

The following need for user interaction is identified:

- Identify groups of surfaces that should be considered as one unit. This identification can be applied after an automatic suggestion of candidates where surfaces can be added or removed.
- Identify non-adjacent surfaces that should belong to the same volume boundary.
- Identify surfaces that should represent volume boundary, again an automatic, initial selection of candidates should be applied.

4.4 BLOCK-STRUCTURED MODELS ALLOWING TRIMMED VOLUME BLOCKS

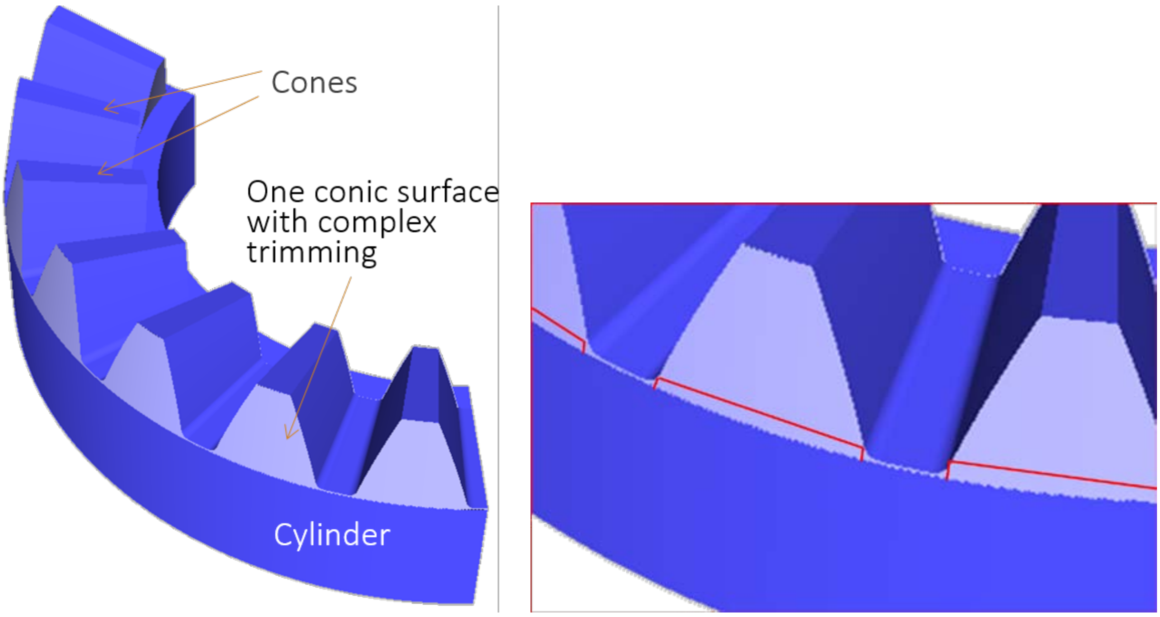

FIGURE 17: PARTIAL NUGEAR CONFIGURATION

Figure 17 (left) shows one CAD solid from the NUGEAR use case. Due to the complexity of the model, the block-structuring would generate many blocks, some of which would be very thin. To the right of Figure 17, a suggestion of surface blocks is visualized. Here, the main part of each tooth is represented as one volume block, but a conical surface joining all teeth leads to separate blocks between the teeth and an extra block at the foot of each tooth, to avoid T-joints. Choosing a representation with only one trimmed volume would, on the contrary, imply that several surfaces that ideally should be boundary fitted cannot be placed at volume boundaries. The piece has nine natural boundary surfaces: the cylinder surfaces at the front and the back of the model, the planar surfaces at the bottom and on each side and inside the teeth, and the conical surfaces at the front, back and top of the teeth. This is too much for one volume, but can be obtained in a model with three volume blocks where one is trimmed. Similarly, the model in Figure 10 can be described by four blocks where two are trimmed.

In general, all blocks in a combined model can be trimmed, but the internal model faces, i.e. the block boundary surfaces, should preferably not be trimmed.

The creation of a combined model inherits many of the complexities from block-structuring and boundary fitting for trimmed models. Appropriate surfaces for boundary fitting must be defined and grouping of surfaces into larger surfaces can be beneficial. Some block-structuring steps need to be applied, but the process must be stopped at the right stage and the selection of where to insert splitting curves and surfaces is crucial. The surfaces selected for boundary fitting will provide guidelines for the block-structuring.

4.4.1 User interaction

The need for user interaction is quite similar to the trimmed model case, but also the selection and definition of vertices representing end points of splitting curves is relevant.

4.5 MODEL CONTINUITY

Block-structured models are guaranteed to have C^0-continuity between blocks. This implies that the outer boundary of the model is also C^0-continuous. This guarantee does not exist for trimmed models. If the shell of the initial boundary represented CAD model is used to define the trimming, this cannot be expected to be continuous in the sense of being watertight. This problem can be approached with two strategies, and both should be pursued.

The lack of watertightness occurs in the context of approximation of intersection curves. If the number of intersection curves is reduced, the occurrence of gaps will be diminished. If two trimming surfaces meet with higher order geometric continuity, meaning continuous normal vector direction or continuous curvature, the two surfaces may be replaced by one single continuous surface. If the transition is also parametrically continuous (C^1 or higher), an exact representation with one surface is possible. Otherwise, the new surface must approximate the initial ones. In this context, the adaptivity property of the LR B-spline surfaces can be beneficial.

Intersection curves following constant parameter curves in both intersecting surfaces need not to be approximated. All three versions of the intersection curve can then be represented exactly and C^0-continuity is easily ensured. Forcing a curve to follow a constant parameter direction, normally requires a non-linear reparameterization of the surface, again an operation leading to an approximation.

It will not always be possible to enforce a trimming shell to be exactly C^0-continuous. Even if some intersections curves may be positioned along constant parameter curves, the configuration of curves trimming a surface may deny this solution of all the curves. Furthermore, sharp corners in the shell should correspond to surface boundaries. Thus, the replacement of several initial surfaces by one continuous surface is not always feasible. When a gap occurs, the functionality computing information for subsequent operations must be able to provide consistent output close to the gap. By regarding the geometry curve as the master, the application will see a watertight model. In order to get a consistent behaviour close to a boundary, for instance in the context of evaluation, it is important that the adjacent surfaces approximate this geometry curve accurately. Otherwise, there will be a non-smooth behaviour between points evaluated on the surface close to this curve and points evaluated on the curve.

4.6 MODELLING OF VOIDS

In order to improve the inner structures of the part for additive manufacturing, voids and cavities will be added to the model. In the CAxMan project, they will be designed using subdivision volumes (see Section 2.4), which allow for smooth and organic shapes while at the same time guaranteeing manifold and consistent geometry.

Adding voids to the part will be done in two ways:

1. Automatic placement of voids based on a set of geometric rules and boundary conditions.
2. Manual design of voids and cavities by a user, using an interactive 3D modeling software.

4.6.1 Automatic placement of voids

For this approach, voids will be automatically distributed and shaped inside the object with respect to the following boundary conditions (as stated in Deliverable D2.1):

- the minimal distance between two neighbouring voids;

- the minimal space between a void and the outer surface of the object;
- the minimal percentage of remaining material.

Additionally, more boundary conditions can be added based on the manufacturing technique and the particular use case:

- necessity for powder-removal channels for powder-based manufacturing;
- necessity for non-removable support structures inside a void to allow manufacturability;
- symmetry constraints along certain axes or planes for moving/rotating parts (important for the NUGEAR use case).

By using a set of distribution parameters and the total weight of the final part as an objective function, this automatic placement algorithm could also be integrated into an optimization loop to improve the quality of the result.

4.6.2 Manual design of voids and cavities

As an alternative to the automatic placement of voids described in Section 4.6.1, voids and cavities could be designed manually by a user using a special 3D modelling tool. This will become necessary, when some regions of the part have to be treated specially and expert knowledge is needed in order to ensure full functionality of the part, after adding voids.

In order to provide the user with some guidance, the CAD surface model will be tessellated and displayed as a semi-transparent outer hull. The user can then design the voids inside this surface by adding and removing voids and manipulating the control mesh of each void (adding new cells, transforming faces, edges, vertices).

The application could also be used to correct and fine-tune the result of the automatic distribution if needed.

4.6.3 Integration of voids into the tri-variate model

Subdivision volumes are not a standard CAD representation and cannot be integrated into a tri-variate NURBS model directly. Therefore, a conversion step is needed.

As described in Section 2.4, Catmull-Clark subdivision solids will be implemented. Their outer limit surface is identical to the one of Catmull-Clark subdivision surfaces, which can be directly described as bi-cubic tensor product B-spline patches for regular vertices (vertices with a valence of 4) in the control mesh. At extraordinary points, however, the limit surface takes a more complex form.

To increase the flexibility of the control mesh and not restrict it to regular vertices only, a general solution will be developed. An algorithm will be developed to approximate the outer surface of each void with a connected set of NURBS surfaces that can then be integrated into the tri-variate model as a trimming shell.

5 HARVESTING SHAPE MODELS FOR ANALYSIS

In this section we go into some detail on the functional interfaces linking the shape representations presented in Section 2 and 4 with the requirements for isogeometric analysis presented in Section 3. The GoTools library from SINTEF will provide these interfaces.

5.1 GOTOOLS

GoTools is used to create a tri-variate isogeometric shape model from an initial CAD model and the library also constitutes a front end towards the isogeometric analysis performed in IGATools.

5.1.1 The `isogeometric_model` module

`isogeometric_model` is a GoTools module designed to keep track of information related to block-structured isogeometric models as defined in Section 4.1 or 4.2. These models contain a number of NURBS surfaces or volumes that have at most four or six neighbouring surfaces or volumes, respectively. The data structure of the `isogeometric_model` includes the shape model and data structures storing associated analysis results and boundary conditions. No geometric modeling or isogeometric analysis is performed in this module. Its purpose is to store information and provide a convenient interface for harvesting information related to the geometric model, the analysis and the relation between the two.

This module was specified as a part of the Exciting project (EC-Project EXCITING SCP8-2007-GA-218536 in the 7th Framework programme) and was implemented in the Exciting and TERRIFIC projects (FP7-2011-NMP-ICT-FoF 284981). The following text is partly fetched from Exciting report no. 6.3.

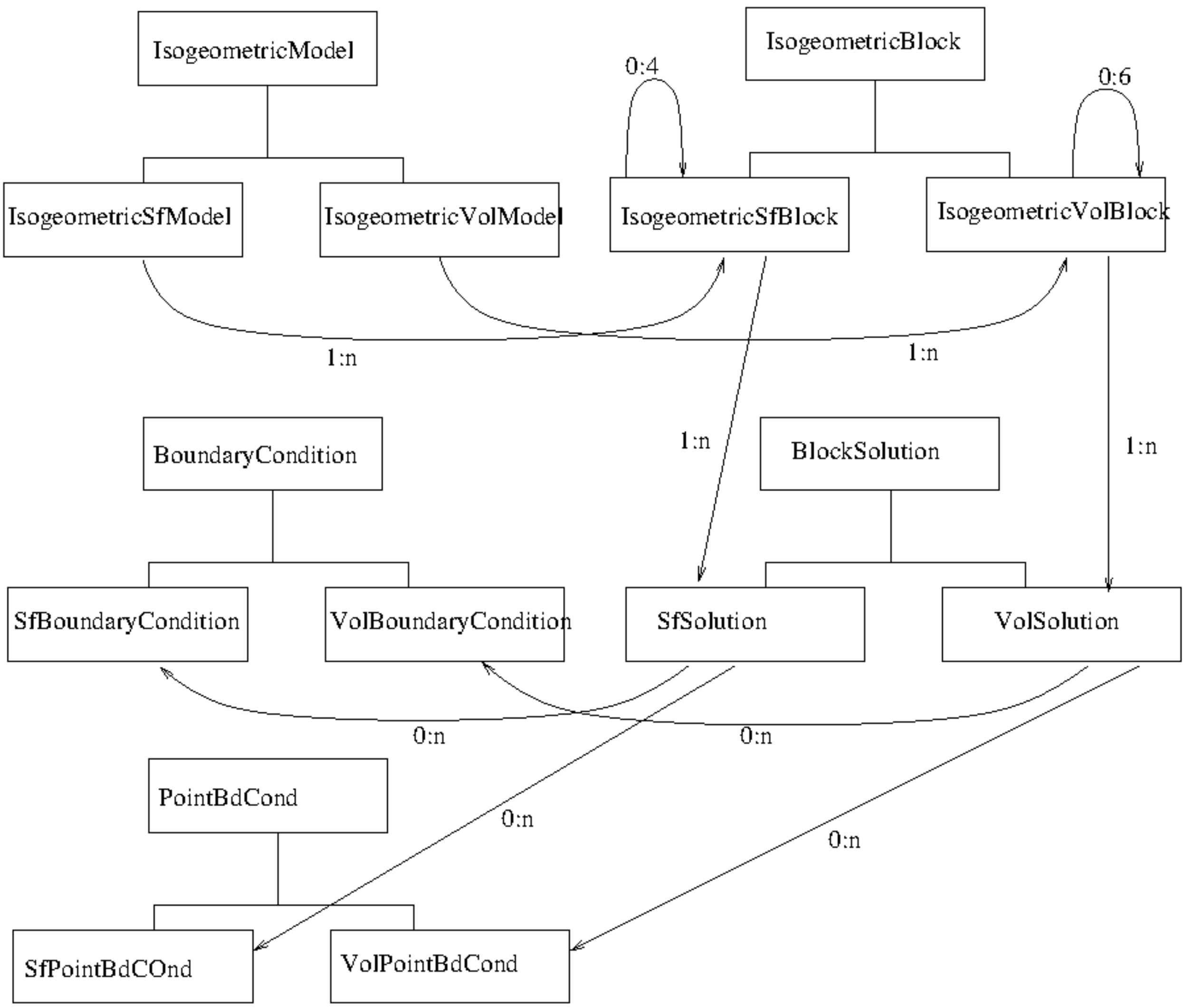

FIGURE 18: DATA STRUCTURE OF THE ISOGEOMETRIC_MODEL MODULE

Figure 18 shows the data structure of the `isogeometric_model` module. A model can be of either surface or volume type, i.e. an `IsogeometricSfModel` entity or an `IsogeometricVolModel` entity. A model entity points to at least one block being represented in the classes `IsogeometricSfBlock` and `IsogeometricVolBloc`. The shape model itself is created in GoTools outside of this module, imported into this structure and distributed to the appropriate number of `IsogeometricSfBlock` or `IsogeometricVolBlock` entities. The blocks know about their neighbours, at most four for surfaces and at most six for volumes. A block is connected to a number of solution entities related to the part of the physical domain represented in the block. Normally, there is one solution associated with a block, but there may be more. A solution has knowledge about boundary conditions and pointwise boundary conditions. These conditions are assigned to the models, and they are further distributed to the appropriate block and solution.

The various classes belonging to the `isogeometric_model` module provide an interface to the spline objects that is convenient when used in the setting of isogeometric analysis. It is for instance possible to:

- Refine a solution space or perform degree elevation.
- Given one block, fetch information about neighbouring blocks.

- Ensure that all adjacent solution spaces share the same spline space to get a correspondence of coefficients at the boundary.
- Fetch information about coefficient correspondence at block boundaries, using the local enumeration of coefficients within a block.
- Fetch information about coefficients belonging to the model boundaries, again using local enumeration.
- Assign Dirichlet boundary conditions to the model boundary representing the boundary condition in the spline space of the corresponding blocks. The boundary conditions will be represented/approximated in the spline space of the corresponding solution and be fetched as a spline curve or surface, depending on the dimension of the shape model, or as spline coefficients.
- Perform pre-evaluation, store the result and return the information as appropriate in order to utilize the tensor-product structure of spline surfaces and volumes.

The structure of the `isogeometric_model` module is relatively static. Limited geometric updates can be performed from the application. No topological changes are allowed. The solutions fields are refined whenever the geometry entities are refined, but the geometry entities are refined with respect to the solution entities only when this is issued from the application.

This class does not incorporate all aspects of isogeometric analysis and it is not intended to do so. The main information related to numerical analysis must be defined by the solvers, as there are many different ways of computing it depending on the technique and the problem formulation. In particular:

- The assembly of the stiffness matrix and right hand side is only supported by providing interfaces for evaluation of splines, the assembly itself is left to the application.
- There are no quadrature rules implemented in this class.
- How the degrees of freedom are enumerated globally is not determined. Only the local enumeration within one single block is fixed: We start counting along the first parameter direction, then the second parameter directions and so on. This affects the information about corresponding coefficients as well as the representation of the solution space.
- Although the positions of the boundary conditions are stored within the data structure, the implementation of these and how to incorporate them into the system of equations is not determined. This is also the case for connecting two blocks. In order to support adjacency, functionality exist to maintain the consistency within the blocks and to get a list of corresponding pairs of degrees of freedom. How this information is used is left open to the application.

The structure of the `isogeometric_model` module is designed with pure block-structured models in mind. The blocks are boundary conforming and adjacent blocks share a common spline space. No T-joints are allowed. This currently invokes some further restrictions:

- The `isogeometric_model` module currently supports NURBS only. LR B-spline surfaces and volumes are not supported.
- Adjacency information is supported through block boundaries and uniqueness of neighbours is expected. This implies that no adjacency information is available when a boundary curve degenerates to a point or when a boundary surface degenerates to a point or to a curve.
- The block topology is rigid. A block-structured model where some blocks are trimmed is not supported.

The `isogeometric_model` module does not currently support trimming, but will be extended in order to support the type of model described in Section 4.4. Much of the functionality provided by this module is of interest also in the trimmed case, but for some operations, the result must be restricted with respect to the valid domain of the surface or volume. The data structure contains information about the outer boundary of the model, but the representation does not include topology information. The main effort in extending this module to handle trimmed blocks is related to enhancing the data structure with complete trimming information and use this to limit the result of queries to the module. For instance, if a grid point is outside the trimmed domain the grid evaluation result for this point must be marked. Similarly, functionality for fetching coefficient correspondence at block boundaries should distinguish between coefficients corresponding to basis functions lying entirely inside the domain, entirely outside or at the boundary. Furthermore, queries on whether or not an element crosses a trimming boundary must be lifted to extend the functionality of this module. Data structures representing adjacency and the relation between the physical domain and the solution relate to the underlying non-trimmed blocks and will not be affected by the introduction of trimming.

5.1.1.1 User interaction

The module provides support for storing boundary conditions in relation to the associated shape model and analysis result. Boundary conditions for the analysis are defined on the boundary of the model, but different faces (or edges in a 2D setting) may be associated different boundary conditions, and it is even possible to restrict the condition to a part of a boundary entity. Defining and specifying a boundary condition is a complex operation that can benefit from an appropriate graphical user interface.

5.1.2 API for trimmed isogeometric models

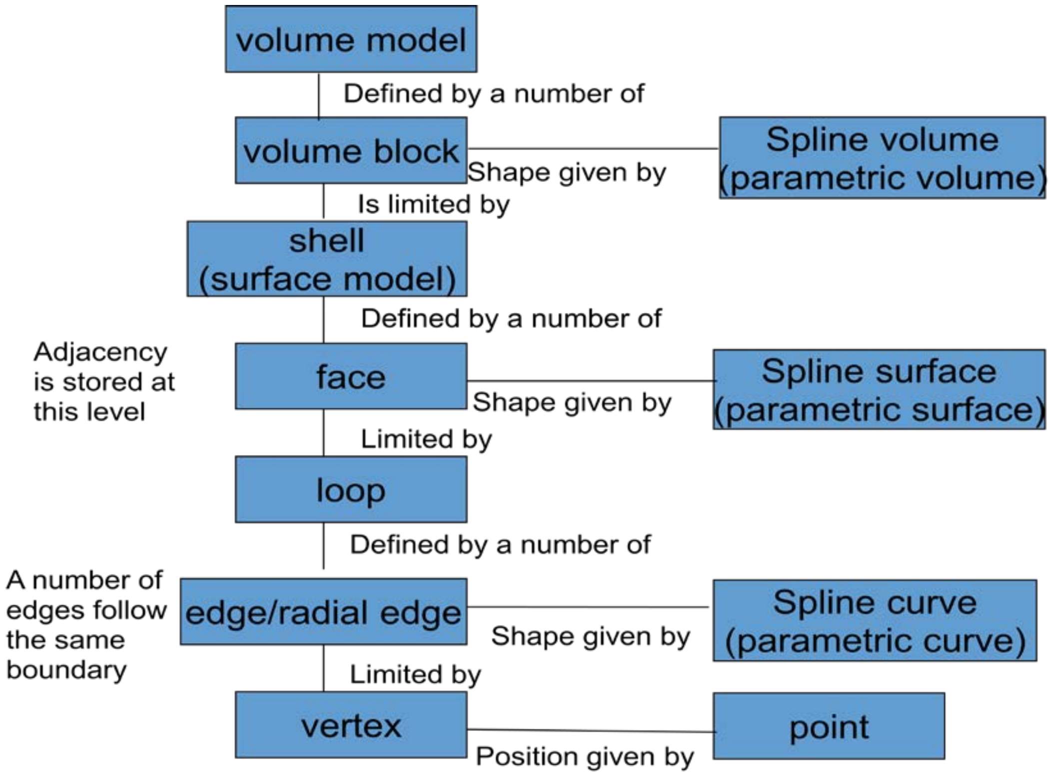

FIGURE 19: DATA STRUCTURE FOR MULTI BLOCK MODELS INCLUDING T-JOINTS AND TRIMMING

Figure 19 shows the data structure of a volume model consisting of several possibly trimmed volume blocks. The structure is general purpose and allows T-joints, so it needs to be restricted for use in IGA. During the construction of a block-structured isogeometric model, this data structure is used to keep track of trimming information and adjacency. A non-trimmed block-structured model can be extracted from this structure and enhanced by data structures of solution fields and boundary conditions to create an `isogeometric_model` described in Section 5.1.1. Many queries to the `isogeometric_model` will be passed on to the entities described in this section.

Each entity defined in Figure 19 has corresponding entities in the GoTools data structure. A trimmed single block model corresponds to a volume block, which in GoTools is implemented in the class `ftVolume`. This is a topological entity providing a link to a geometry entity represented as a spline volume and one or more shells defining the outer trimming boundary of the model and potentially voids. The outer boundary can trim the volume and can also contain trimmed surfaces. A model with several trimmed blocks can be represented as a volume model.

The trimmed volume model is assumed to have been created already and we will now explain the GoTools functionality meeting the requirements of isogeometric analysis outlined in Section 3. Assuming that the volume does not have any void, we can check if the volume has a purely geometric representation without trimming, we can fetch the outer shell of the possibly trimmed volume, perform a point in volume test and fetch the associated geometric representation of the non-trimmed underlying volume.

Most operations on the volume block are performed on this associated geometric volume entity. We can perform evaluation and iterate for a closest point. Information on the parameter domain

of the volume is available and we can check if the volume is actually a spline volume. In that case, some more functionality is available. Evaluation in a grid can be made more effective by a dedicated grid evaluation. The number of degrees of freedom can be extended by knot insertion and degree raising. Important in the context of isogeometric analysis is the access to the basis functions of the spline volume: evaluation including grid evaluation is available as well as information on the polynomial patches (elements). Two spline volumes represented in the same spline space may be added and queries about whether or not the volume is represented in a left handed coordinate system or the volume has got some kind of degeneracy, are available.

The shell (entity `SurfaceModel` in GoTools) is a collection of faces. The number of faces can be queried and a face can be accessed as a topological or geometrical entity. A number of operations like closest point and intersections with lines, planes, spline curves and other surface models can be applied to the shell as a whole. The number of boundaries of a shell can be obtained. A closed shell or a solid has zero boundaries. To check the feasability for isogeometric analysis, a check on a corner-to-corner configuration can be made.

A face has similar functionality as a topological volume including evaluation and closest point computation. A point in face test is available and the face normal can be obtained. In addition, the geometrical surface, which may be trimmed, can be extracted. A general parametric surface can provide information on the parameter domain and we can check if the surface is actually a spline surface. In that case, various grid evaluators and the access to the basis functions are available similar to the case for spline volumes.

GoTools also contains namespaces collecting free functionality related to specific classes. In this context a query on whether or not a specified face in some shell corresponds to a boundary surface of the volume is of interest.

The requirements of Section 3 have led to the following plans for GoTools extensions and improvements:

- Check if an element crosses a trimming boundary, surface or volume.
- Restrict grid evaluation to a trimmed domain.
- Improvement of existing functionality, in particular related to "point-in-volume" test.

In GoTools, spline coefficients and associated basis functions are ordered according to their position in the parameter domain of the surface or volume. Elements are indexed according to their foremost, lower, left corner.

5.2 API FOR SUBDIVISION VOLUMES

To design, calculate and export voids and inner structures, Fraunhofer will develop a library containing the necessary functionality. Since voids and cavities will be integrated as subdivision volumes within the CAxMan project, the focus of the library will be on volumetric polygonal meshes and subdivision algorithms. It will be based on a volumetric Half-Face data structure that allows fast access to neighbouring elements for each cell, face, edge and vertex of the mesh. This is a key requirement, when dealing with subdivision surfaces or volumes.

Although the voids will be internally represented as subdivision volumes, only the outer surface is relevant for the integration with the other modules developed in WP2. Therefore, the library will only expose the surface-related functionality as an API for outside use.

The main functions will be:

- Get the control mesh of the outer surface of a void.
- Get a tessellated mesh of the outer surface for an arbitrary subdivision level.
- Get the position and derivatives of the evaluated limit surface at any parameter values.
- Get the surface normal of the evaluated limit surface at any parameter values.
- Get the basis functions of the outer surface of a void.

For integration into the NURBS model, the subdivision library will also provide functionality to approximate the outer surface of each void as a set of connected, watertight NURBS patches. Therefore, the following additional functions will be made available through the API:

- Check if a given point is inside a void or not.
- Check whether the void boundary intersects a specified element (one polynomial patch in the volume).
- Calculate the NURBS surface approximation for a void.
- Get the control points and the parameterization of the approximated NURBS surface.

6 USER GUIDANCE

In the previous sections, we have described when user guidance through a graphical user interface is beneficial. This section summarizes this information.

Most of the need for user guidance appears in the context of model generation. The user input should be fetched at different stages of this process. Typically, an investigation stage should be applied prior to any user guidance. This process should result in suggestions on CAD surfaces that carry much shape information and are candidates for being boundary surfaces in a trimmed approach. It should also suggest groups of surfaces that constitute opposite surfaces in one parameter direction of the tri-variate representation and surfaces that can be merged into larger entities. The result of this pre-process should then be presented to the user, who is given the opportunity to deselect surfaces from the different groups and add new surfaces not identified by the automatic process. At this stage also singularities (extraordinary points in a block-structured approach) and other additional vertices in the model should be identified.

If a block-structuring process is applied, new user information can be added after the generation of surface blocks. At this stage, vertices in the outer shell can be identified as end points of block boundary edges of volume blocks.

The manual design of voids and cavities uses an interactive 3D modelling tool as described in Section 4.6.2. This modelling tool is independent of the graphical user interface presented above.

7 MODEL ANALYSIS

In this section, we will analyze models related to the CAxMan use cases and decide on tri-variate shape models appropriate for the particular cases.

7.1 GEAR MODELS

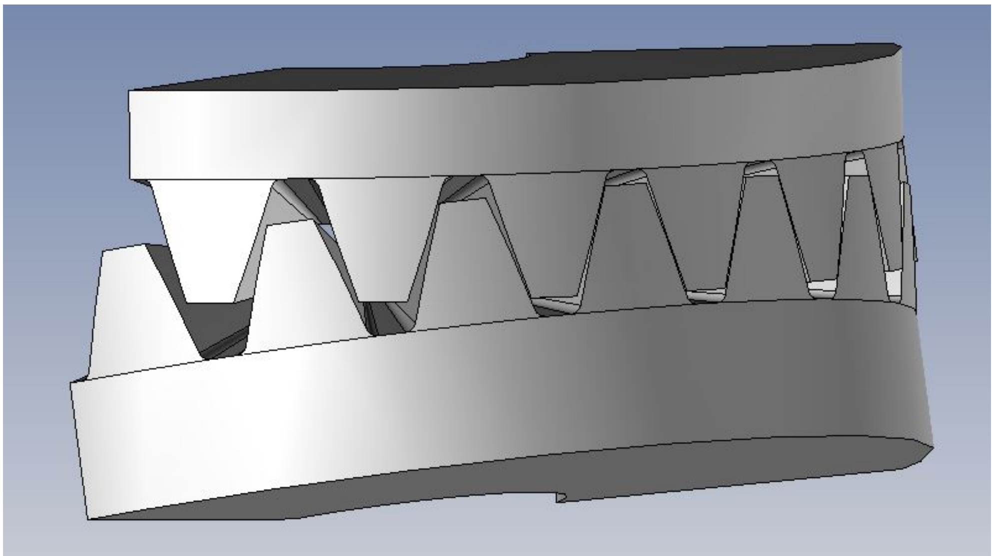

FIGURE 20: SINGLE-COUPLED GEAR MODEL

Two versions of the NUGEAR model are currently provided as CAxMan use cases, one single coupled model and one doubly coupled model, both having straight teeth. The gear models are in essence cylindrical, a property that should be maintained in the trivariate remodeling. The single-coupled model consists of two pieces which are modelled as separate boundary represented solids, see Figure 20. The two models have quite similar properties and we will discuss the lower gear part.

In Figure 21, the lower part of the gear model is shown. The main surfaces are cylindrical, conical and planar, but the model also contains some B-spline surfaces placed between the teeth. Each tooth end is represented as one trimmed conical surface, but all these surfaces relate to the same underlying cone. The top surfaces of the teeth are also conical.

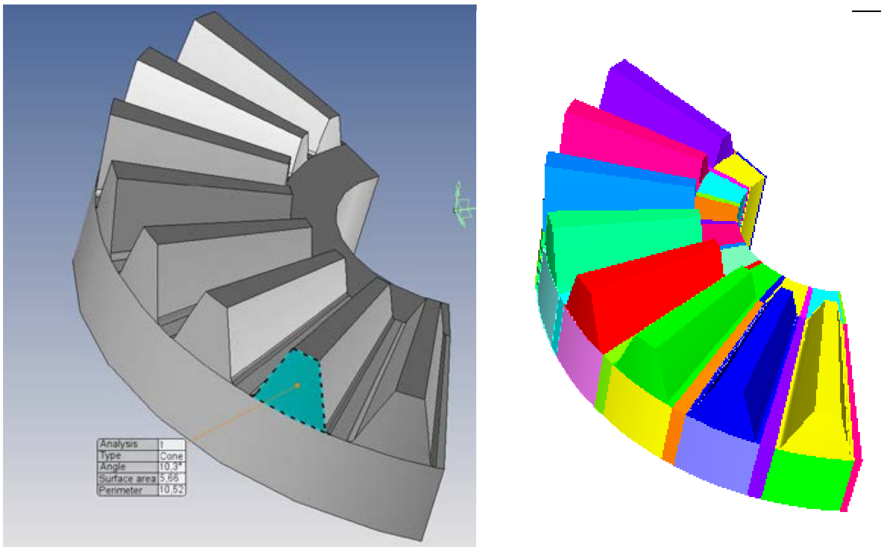

FIGURE 21: LOWER PART OF THE SINGLE-COUPLED GEAR. BOUNDARY REPRESENTED MODEL (LEFT) AND BLOCK-STRUCTURED TRI-VARIATE MODEL (RIGHT)

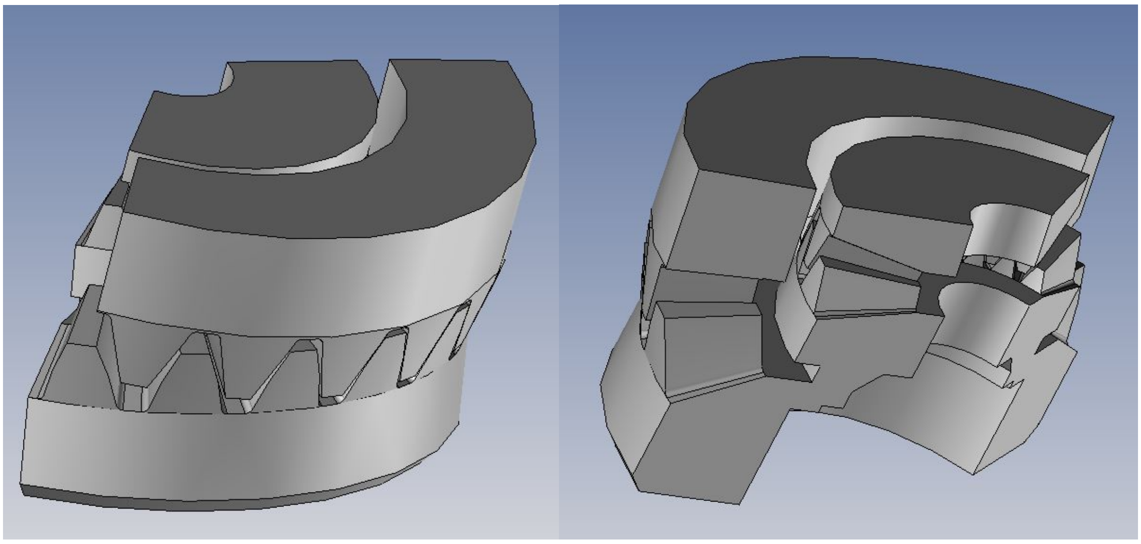

FIGURE 22: DOUBLE-COUPLED GEAR MODEL

This model has an obvious tri-variate representation using the block-structured approach where each tooth is represented as one NURBS surface and the remaining part of the model is split into blocks accordingly to avoid T-joints. The version shown in Figure 21 (right), approximates the conical and cylindrical surfaces with polynomial B-spline surfaces. As the teeth join smoothly with the support, the blocks representing the teeth will have corner degeneracies.

The double-coupled gear is more complex, as can be seen from Figure 22. It consists of three solids, one is similar to the solids in the one couple gear, but the other two differ. All the parts are essentially cylindrical.

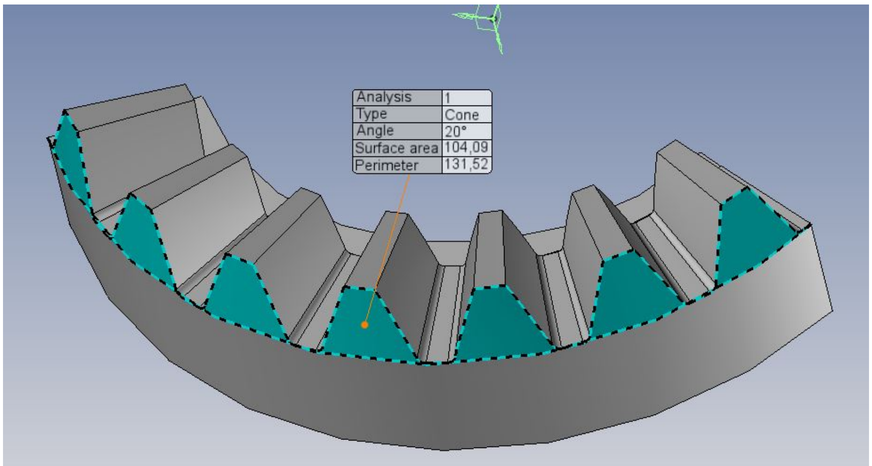

FIGURE 23: OUTER PART OF THE DOUBLE-COUPLED GEAR

Consider Figure 23. This part has much in common with the one we studied for the single-coupled gear, but there is one essential difference. Instead of each tooth having separate conical surfaces at their ends, this model has one single trimmed conical surface representing the tooth end for all teeth (the turquoise surface in Figure 23). A block-structured approach following the pattern of the single-coupled gear where each tooth is represented by one block, would result in very many blocks where some of them are quite thin. An indication of a block-structured solution is shown in Figure 17. Other gear models of the same type have a thicker layer between the cylinder and the teeth, which would lead to blocks having a more regular shape. The number would, however, still be high. This part could be represented by a trivariate model combining block-structuring and trimming. Three blocks meeting in a corner-to-corner configuration, where one of the blocks is trimmed is sufficient. There will be two cylindrical blocks at the bottom of the part represented as non-trimmed NURBS volumes and one trimmed NURBS block representing the teeth. The boundary surfaces for adjacent volume blocks will be non-trimmed.

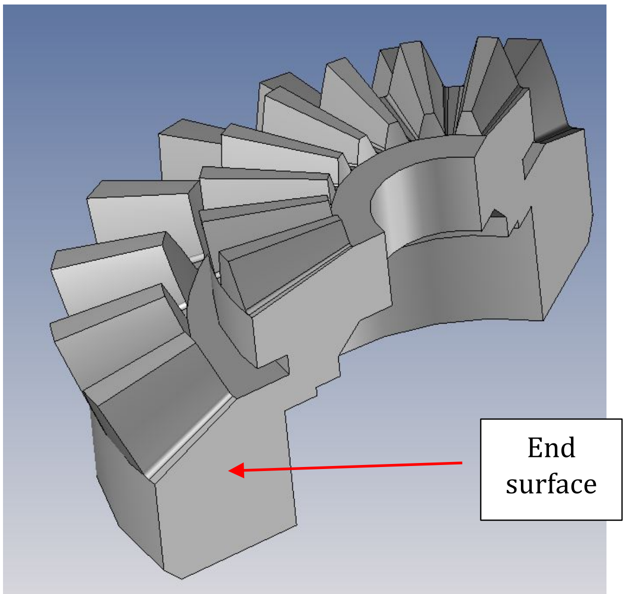

FIGURE 24: THE MIDDLE PART OF THE DOUBLE-COUPLED GEAR

The last solid pictured in Figure 24 is more complex. In one row, the end surfaces of the teeth are separated, in the other they are not. This alone indicates that a block-structured solution is less attractive, and finding a block-structuring topology would be very demanding. A model combining trimming and block-structuring implies that the end surface shown in Figure 24 needs to be block-structured. Taking the corner-to-corner configuration requirements for the block-structuring into account and wanting to avoid surface blocks with an edge collapsing, this part already reveals a lack of good solutions. Thus, a trimmed, tri-variate model seems to be the most adequate choice. The underlying spline block must take the cylindrical shape of the model into account. Due to the complex shape of this part, few surfaces can be boundary fitted, but the cylindrical and planar surfaces can be modelled as isoparametric surfaces in the volume block. Furthermore, many of the model edges will lie in isoparametric surfaces perpendicular to the cylinder axis. This implies that there is a good potential for creating a watertight trimming shell.

Gears with curved teeth are expected to have similar properties to the ones discussed in this section, but they imply an increased complexity. The definition of a the block-structured topology in the cases where this is an appropriate solution, becomes more complex and it will be more difficult to ensure watertight shells for the trimmed models.

7.2 INJECTION MOULD

The second CAxMan use case is related to injection moulds and particularly a mould for creating a holder for ear plugs designed as a CAxMan gift. The item itself can be seen in Figure 25.

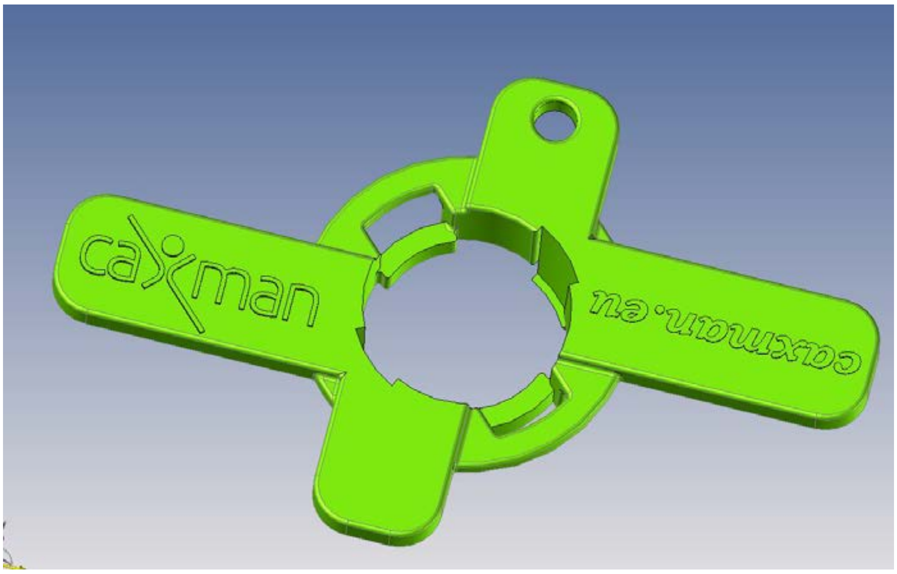

FIGURE 25: THE CAXMAN GIFT

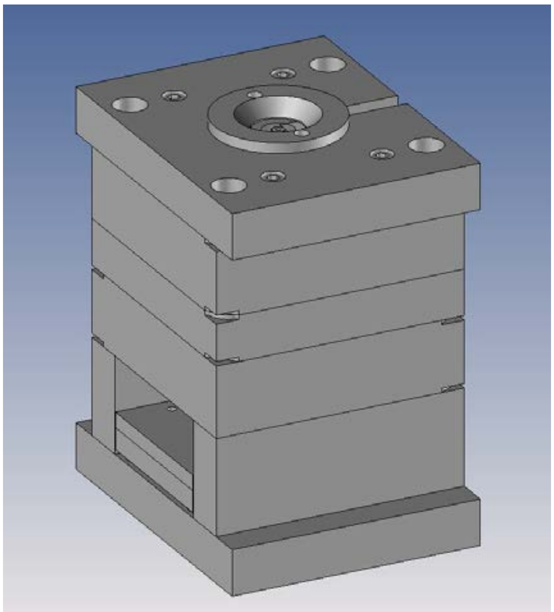

FIGURE 26: INJECTION MOULD FOR THE CAXMAN GIFT

The gift item will not be manufactured by additive methods, but it is still interesting to consider the opportunities for creating a tri-variate model. The CAD model contains several surface types: The large surfaces are planar or conic, but also cylinder surfaces, spheres, tori and B-spline surfaces are included. The complexity of the part indicates that a trimmed triviariate model is the most feasible. The top surfaces of the letter lie in a common plane, which is raised relative to the main top plane of the item. Thus, the spline volume boundary cannot coincide with the top plane, but this plane can still be represented as a constant parameter surface. The middle part of the object is essentially conical. The associated conical surfaces have a 2 degree angle in the apex. This structure should preferably be reflected in the volume description to let the shape bearing surfaces become constant parameter surfaces in the volume.

An injection mould aimed at creating a particular item is often more complex than the object itself. Figure 26 shows the complete mould for the gift. The mould consists of a large number of solids with shapes of varying complexity including some standard components like bolts and different types of plates. Only the moulding parts (Core and Cavity) will be printed, but a couple

of other parts will also be included in this discussion to extend the scope of shapes being analyzed.

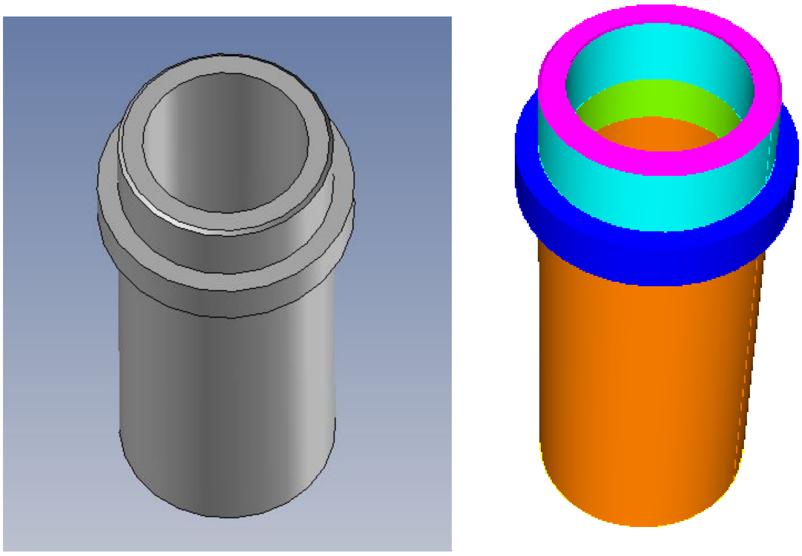

FIGURE 27: GUIDE BUSH WITH CENTERING COLLAR

Figure 27 (left) shows a rotational object consisting of planar surfaces with circular trimming curves, cylinder surfaces and cones. Thus, the object can be block-structured using the strategies explained in Section 4.2, as can be seen in Figure 27 (right).

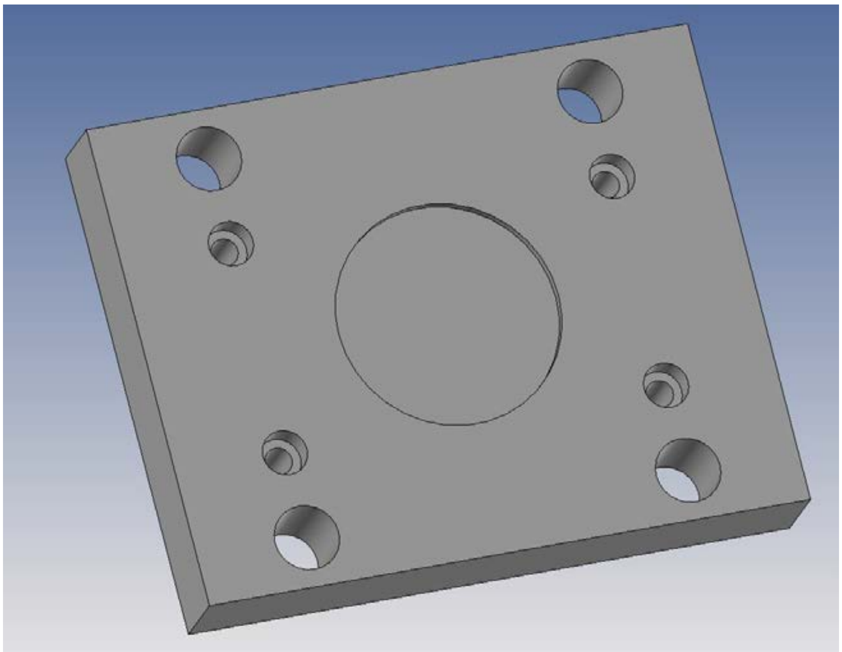

FIGURE 28: PLATE

Figure 28 shows a plate with holes and intrusions. The surfaces are planar and cylindrical. This type of part has a block-structured topology following the algorithm outlined in Section 4.1, but the number of blocks will be high and the construction methodology will be complex. Alternatively, the part can be described as a trimmed model where the main planar surfaces of the part is boundary fitted. It is not feasible to force the trimming curves to be constant parameter curves in the parameter domain of the planes, but as all representations of all trimming curves have an exact representation as lines or circles, exact C^0-continuity can be obtained.

48)

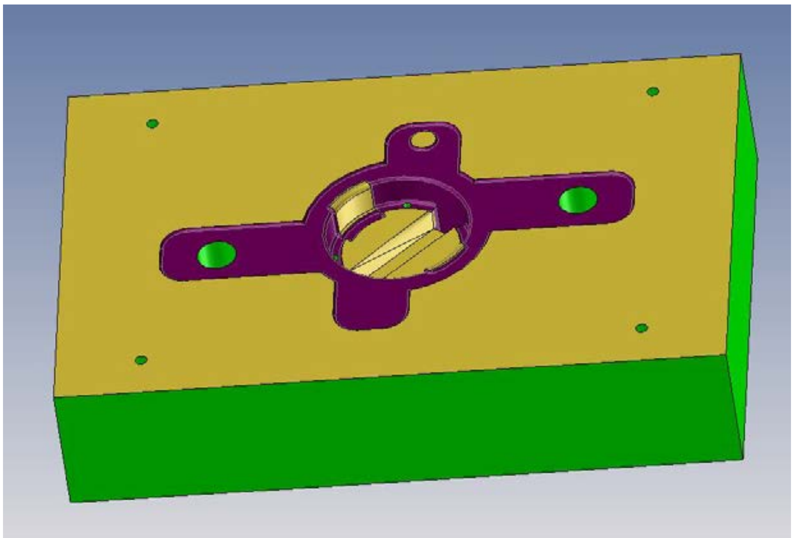

FIGURE 29: INJECTION MOULD, LOWER PART

The mould parts in Figure 29 and Figure 30 are intended for additive manufacturing. The lower part (Figure 29) fits well with the trimmed approach, see Section 4.3. The underlying spline volume can be designed to follow the side surfaces of the box. The trimming shell around the gift item consists of many surfaces, both analytic (tori, cones, planes, cylinders) and freeform. Adjacent trimming surfaces meet in sharp edges for some configurations and smoothly for others. It might be advantageous to simplify this shell by merging surfaces to get a simpler topology and a reduced number of gaps and overlaps. Still, it is important to interpret the trimming information in a consistent way when harvesting information from the model, see Section 4.5.

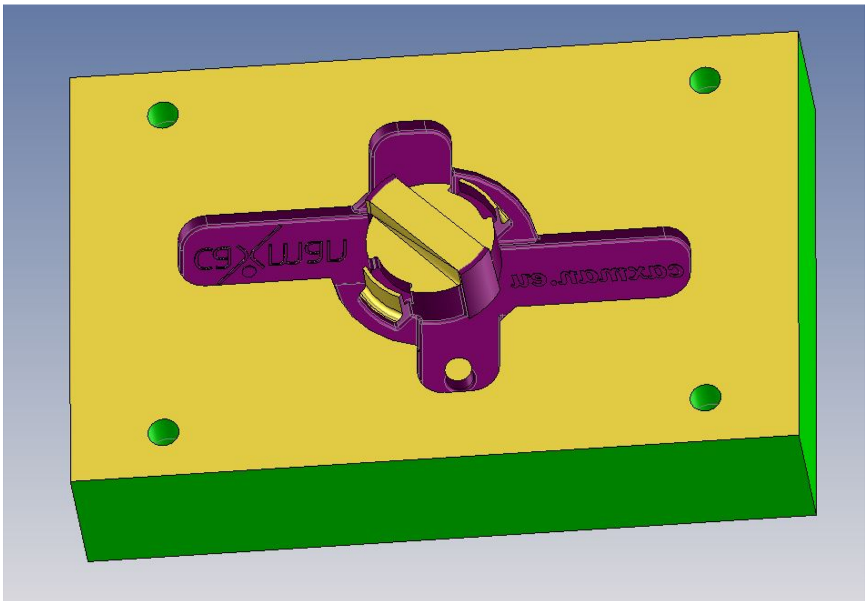

FIGURE 30: INJECTION MOULD, UPPER PART

The upper part of the injection mould, Figure 30, should be treated similarly to the lower part. The main difference is that the central part of the mould is raised compared to the box around it. Thus, the box cannot be exactly fitted with the underlying spline volume. However, all box sides can be represented as constant parameter surfaces in the spline volume.

8 INTEROPERABILITY AND WORKFLOWS

In this section we present the aspects of interoperability with the outside world and with other work packages in CAxMan as well as detailing the initial internal workflows of WP2.

In the main workflow of CAxMan, which is depicted in Figure 31, WP2 only has one output (to the process planning stage done in WP3) but several differing inputs (e.g., STEP CAD model or any number of revision requests from later stages of the AM workflow). In this section we describe various interfaces and possible routes along the foreseen WP2 workflow.

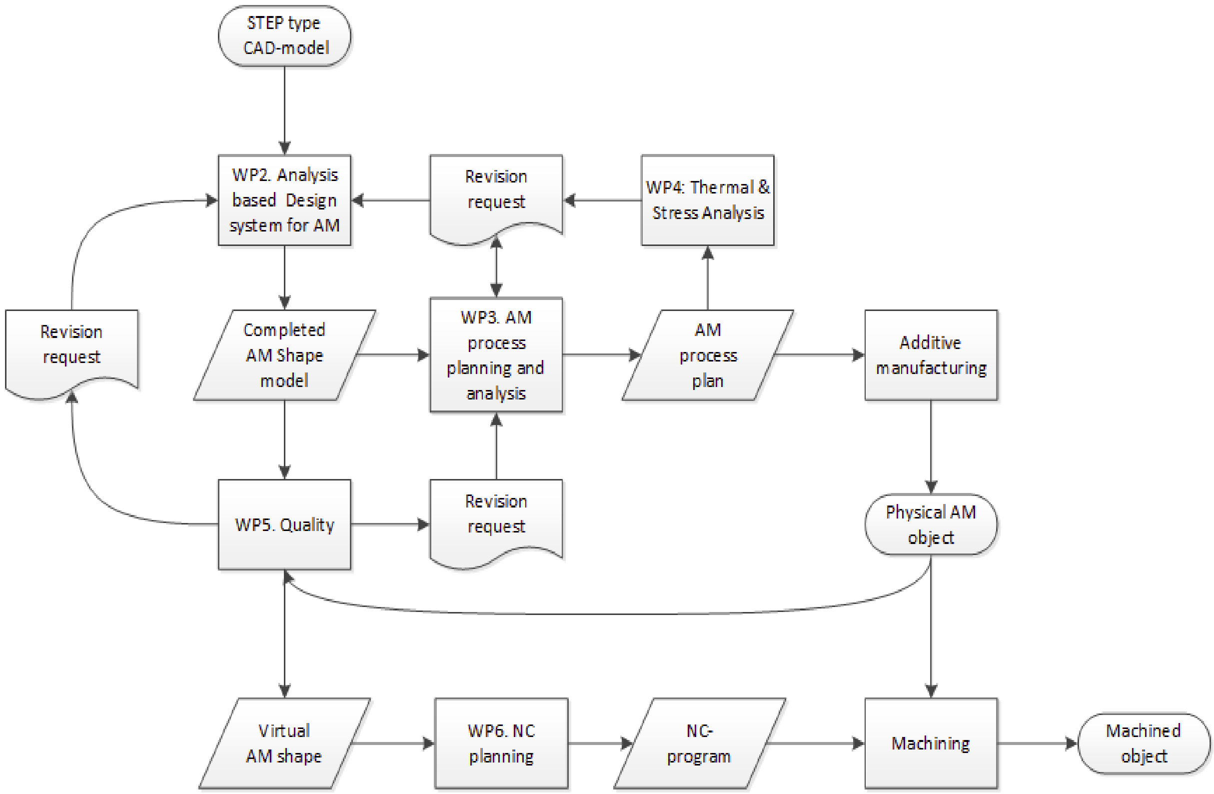

FIGURE 31: CAXMAN WORKFLOW

8.1 THE RELATION BETWEEN STEP AND IGA ADAPTED SHAPE MODELS

A STEP file contains one or more boundary represented solids whose entities are defined in STEP Part 42, an integrated generic resource for geometry and topology representation (ISO 10303, 2015). Prior to defining the isogeometric shape model, each B-rep solid is read into GoTools and represented as a `SurfaceModel`. This entity is the GoTools version of a shell. A shell can be open or closed while a B-rep solid consists of one or more closed shells, one outer shell and possibly some voids. SINTEF is working on the topic of extending the capabilities of GoTools, among others, to support the data model available in the ISO 10303 STEP type boundary solid. Consequently, SINTEF is also enhancing the native data storage format of GoTools to a new version called g22, aimed at maintaining the topological information contained in the STEP file.

The algorithms of WP2 may result in changes to the original design which should be stored in a standard file format. This means that the isogeometric shape model must be converted back to STEP format to reflect the possible shape changes and also the insertion of voids. The voids will be designed using subdivision volumes, but when the design process is complete, the boundaries of the voids will be approximated by NURBS, ensuring their compatibility with STEP. This will enable us to reflect the results of the shape optimisation and other design routines in the master model.

The shape models considered in this report are tri-variate and consist of one or more spline volumes which can be trimmed. STEP Part 42 supports spline volumes and has recently been extended with the topological volume entity `volume_with_faces`. This is an abstract entity that can be defined either as `volume_with_parametric_boundary`, which represents a non-trimmed volume entity, or `volume_with_shell`, which can represent a trimmed volume. The volume entities can be collected in a `connected_volume_set`. The topological volume entities cannot represent voids, thus a model with voids must be split into several pieces to include pieces of the voids into the outer boundaries of the submodels.

In the longer term, we aim to investigate to what extent the isogeometric model can be stored in STEP format, in order to include variable information about the object interior, such as material distribution or porosity. We will also look at how LR B-spline surfaces and volumes can be utilized in STEP for these purposes; these entities are already included in the Part 42 extension. Furthermore, the last version of AP 242 supports the extensions on topological volumes and locally refined splines.

8.2 FORMATS FOR INTERFACING WITH OTHER WPS

8.2.1 Output formats from WP2 to WP3

In the initial stages of the project, WP3 needs to receive a representation of the geometry that is compatible with process planning tools that are available today. Thus, in the first instance, WP2 will output a watertight tessellation of the smooth WP2 geometry.

Tessellations can be generated from either boundary represented CAD models or from volumetric representations. The main requirement coming from the process planning in WP3 is that the mesh is manifold; that is, it does not contain self-intersecting triangles, gaps between triangles or other geometric errors that cause the volume to not be well defined.

There are a number of tessellation formats to choose from, many of which are outlined in D3.2 — AM Process Planning Workflows. It has been agreed that the initial format will be OFF, which is a simple format defining polygons by a list of coordinates (of vertices) followed by a list of indices (of these vertices) defining triangles. This choice gives us flexibility as to whether we label vertices or triangles. It is expected that this representation will be replaced by STEP tessellations in the subsequent development phases.

Later in the project we intend to explore how some parts of the process planning may benefit from directly utilizing the isogeometric representations from WP2. In particular, the slicing of the model can be performed more accurately and with better preservation of the original design properties (e.g. material distributions) when done directly on the isogeometric model. In this case a new format must be established that encompasses the concepts presented in the earlier sections, and which conforms to STEP standards. If this is not possible, extensions to the STEP standard may be proposed.

8.2.2 Input formats from WP3–WP5 to WP2

The conversion to a tessellation from a smooth (spline) representation that is made when passing the model to WP3 results in a loss of information. In the case that the process planning is successful, the tessellation can be sampled from the smooth model at a high resolution that is sufficient for printing. However, it is expected that in some cases, feedback must be provided to WP2, coming from WP3 or the later stages of WP4 and WP5 that utilize the tessellated model, in

the form of revision requests. Such requests will indicate problems with the design that cause issues in certain parts of either the digital or the physical model (see D3.2 — Section 2, for a list of expected types of revision request). To make the connection between the respective parts of the smooth and tessellated model, additional information will be attached to the triangles/vertices when performing the tessellation, which we will call 'annotations'. The revision requests can then be formulated by referring to the affected triangles, and the part of the smooth model to be changed can be inferred from the annotations. More details on the annotation format are provided in the appendix of D3.2.

In current additive manufacturing workflows, this feedback is typically not possible, or at least much more difficult to implement, since the tessellation in STL format does not give any direct information about which part of the model it was generated from. We foresee that by preserving this connection between the smooth and tessellated models, it will be much easier to integrate seamless feedback from the process planning to the design stage, thereby easing interoperability and better supporting the use of automation or semi-automation.

8.3 INTERNAL WP2 WORKFLOWS

In this section we give some details of the workflows that are internal to WP2. Initially, WP2 will implement the basic functionality outlined in the minimal workflow below, but new components will be integrated in the workflows as and when they become available in the CAxMan infrastructure.

8.3.1 Minimal WP2 workflow

The minimal WP2 workflow is to tessellate the input STEP model and provide the tessellation as output to WP3, without the core WP2 components of void/cavity design and isogeometric analysis. The main requirement here is that the tessellated model defines a closed and printable volume.

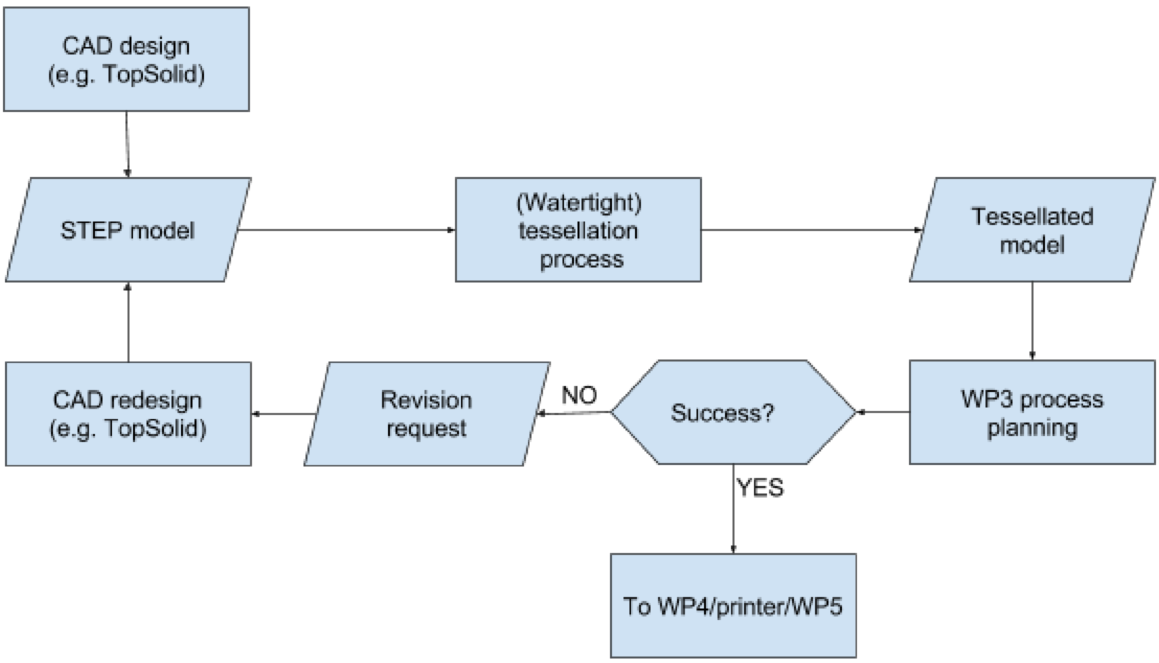

FIGURE 32: MINIMAL WP2 WORKFLOW

We expect the input STEP B-rep file to be defined with surfaces and tolerances such that it defines a closed volume. Since the first step of WP3 will be to check the integrity of the tessellation (e.g. for watertightness), the requirement for watertight STEP files is not strictly necessary, but lack of watertightness should result in a revision request that indicates to the designer which parts of the design need fixing. It is, however, required that if the STEP file is watertight, the tessellation produced preserves this watertightness. This ensures that any issues arising are fixed in the design, and not just in the tessellation.

The workflow of this section is typical of the processes used in industrial additive manufacturing today in order to prepare a STEP CAD file for process planning. In this case, the 'revision requests' can be thought of as some sort of failure in the process planning or later stages, that should be handled by an operator. This workflow may thus be used as a benchmark to evaluate the performance benefits of CAxMan when compared to the comprehensive workflow to be developed, as outlined below.

8.3.2 Comprehensive WP2 workflow

The aim of WP2 within the duration of the CAxMan project, is to link several components of analysis and design in workflows that allow comprehensive control over the analysis-based design process, prior to being sent further to AM process planning. The work of WP2 must also consider input coming from several possible revision requests generated in later stages of the AM process. Revision requests will be handled by a generic "revision request handler", which will pass both the data and the revision request to the required part of the WP2 workflow. Revision requests can require either user guidance or can be automated in some cases. A first version of the comprehensive workflow is outlined in Figure 33 below.

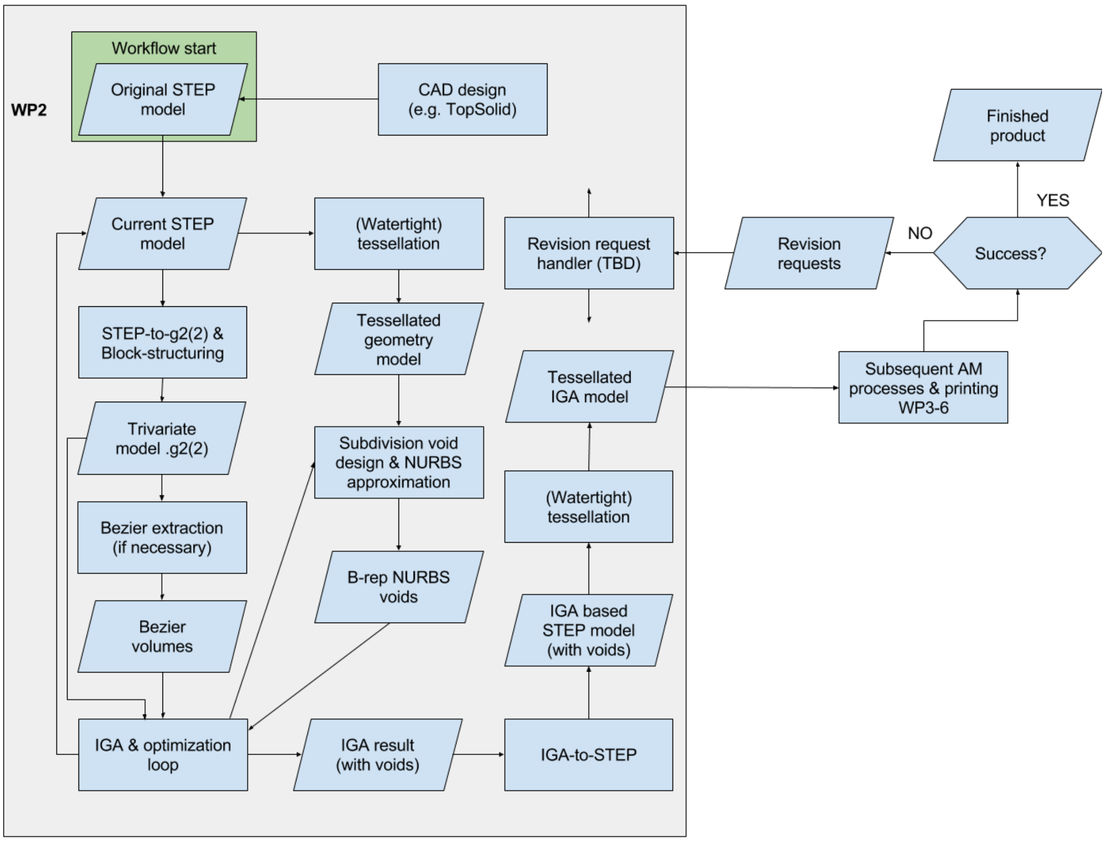

FIGURE 33: COMPREHENSIVE WP2 WORKFLOW

As in the minimal workflow, we begin with the original STEP model, typically originating from some CAD software, such as Missler's TopSolid. The model will enter the design loop by following two parallel paths: one converts the model to tri-variate form (in GoTools), and the other prepares the model for void design using subdivision techniques. The tri-variate Bézier extracted model, together with the NURBS approximated voids will make up the input to the isogeometric analysis stage (performed with IGATools). The results from the analysis could result in loops to optimize the design, either by redesigning the voids, or by adapting the tri-variate model. When the optimization loops have completed, the resulting model will be converted back to STEP format to become the new master model. Finally, this STEP model will be tessellated with annotations for further processing and printing in the subsequent work packages. The steps of the subsequent work packages may require a change in the original design, resulting in a revision request. The details of how the specific revision requests will be handled within WP2 will be specified in a later deliverable, but a list of possible revision requests arising from WP3 is already provided in deliverable D3.2.

This workflow will serve as a basis for the initial WP2 implementations in the CAxMan infrastructure. However, it is expected that the workflow will be refined and iterated in the subsequent stages of the project as the CAxMan services and revisions requests are implemented.

9 REFERENCES

Catmull E., C. J. (1978). Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10 (6)*, 350–355.

Chang Y.-S., M. K. (2003). An interpolatory subdivision for volumetric models over simplicial complexes. *Shape Modeling International, IEEE*, 143–152.

Dokken T., P. K. (2013). Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design 30(3)*, 331–356.

Farin, G. (1999). *Curves and surfaces for CAGD, a practical guide.* San Fransisco: Morgan Kaufmann Publishers.

Forsey D. R., B. R. (2003). Hierarchical B-spline refinements. *SIGGRAPH 88 Conference Proceedings 4*, (ss. 205–212).

Hughes, T., Cottrell, J., & Bazilevs, Y. (2005). Isogeoemtric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg. 194*, 4135–4195.

ISO 10303. (2015). *Part 42. Integrated resources: Geometric and topological representation.*

Joy K. I., M. R. (1999). The refinement rules for Catmull-Clark solids. *Tech. rep., Citeseer*.

Kobbelt, L. (2000). Root 3-subdivision. *In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (ss. 103–112). ACM Press/Addison-Wesley Publishing Co.

Loop, C. (1987). *Smooth subdivision surfaces based on triangles.* University of Utah — Dept. of Mathematics: Thesis (M.S.).

Schaefer S., H. J. (2004). Smooth subdivision of tetrahedral meshes. *In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometric processing*, (ss. 124–154).

Schillinger, D., Dede, L., Scott, M., Evans, J., Borden, M., Rank, E., & Hughes, T. (2012). An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput. Methods Appl. Mech. Engrg. 249–252*, 116–150.

Sederberg T. W., Z. J. (2003). T-splines and T-NURCCS. *ACM Trans. Graph 22(3)*, 477–484.