TERRIFIC
Enhancing Interoperability

SEVENTH FRAMEWORK PROGRAMME

Deliverable D6.3

Trivariate isogeometric models from STEP-files, Version 1

Circulation: PU

Partners: SINTEF

Authors: Vibeke Skytt (SINTEF),

Version: 1

Date: September 17, 2012

Copyright

© Copyright 2012 The TERRIFIC Consortium

consisting of:

SINTEF	STIFTELSEN SINTEF, Norway
JOTNE	JOTNE EPM TECHNOLOGY AS, Norway
ECS	Engineering Center Steyr G.m.b.H. & Co KG, Austria
JKU	UNIVERSITAET LINZ, Austria
SIEMENS	SIEMENS AG, Germany
UNIKL	TECHNISCHE UNIVERSITAET KAISERSLAUTERN, Germany
MISSLER	Missler Software, France
INRIA	INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE, France
ALENIA	ALENIA AERONAUTICA SPA, Italy
UNIPV	UNIVERSITA DEGLI STUDI DI PAVIA, Italy

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the TERRIFIC Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or in part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

All rights reserved.

This document may change without notice.

Document History

Vers.	Issue Date	Stage	Content and changes
1	17th September 2012	100%	Ready for submission

Executive Summary

This document contains deliverable D6.3 of the TERRIFIC FoF STREP Project.

This deliverable describes the first version of creation of trivariate isogeometric models given boundary represented solids described in the STEP format, in the TERRIFIC isogeometric toolkit.

Contents

1 Introduction

D6_3 is concerned with the creation of trivariate isogeometric models from STEP file describing a boundary represented CAD model.

The construction of trivariate models is still a bottleneck in isogeometric analysis. Various approaches to this problem are being pursued and in our case we intend to construct block structured models where each block is a NURBS volume. No T-joins are allowed, and the coefficients of adjacent volumes must be identical at block boundaries.

The input for the construction is taken from a boundary represented model and we aim at an automatic process to generate the trivariate model. However, this is a very challenging task and there are a number of restrictions to the type of models we are able to handle:

- The model must consist only of one solid
- The model representation must be of good quality, in particular the directions of the surface normals must be consistent throughout the model, and the gaps and overlaps between adjacent surfaces must be small and not exceed the tolerances given in the model. Further, the model must be physically realizable, in particular no self intersections are allowed.
- The model must have an identifiable thickness direction, but need not to be planar.
- The model must be relatively simple with a limited number of different features.
- The model must consist of large surfaces. Merging of surfaces is performed only for very special configurations.
- Blends cannot be expected to be handled at this stage.

The deliverable contains prototype software for creation of block structured, trivariate models fit for isogeometric analysis given simple boundary represented CAD solids. The code is organized in the GoTools libraries. Three approaches are implemented:

- Recognize and recreate a linear swept model.

- Recognize a and recreate a rotational model.
- Generate a trivariate model be dividing a trimmed volume into regular blocks. The trimming surfaces of the volume corresponds to a given boundary represented solid.

For all three approaches the translation of a face set into a block structured surface model where each block is a NURBS surface is essential. This process is described in section 3. The three trivariate approaches are described in section 4, section 5 and section 6. The communication with the STEP file is described in section 2, future work in section 7 and the report is concluded in section 8.

2 The STEP reader

The STEP interface in GoTools uses JOTNE's tool EDM as an interface to the STEP file. JOTNE has lately made a new interface to this tool, and the GoTools STEP reader are currently being updated to take advantage of that. Simultaneously, the GoTools data structures and data types for curves and surfaces are extended to be able to make use of more of the information in the STEP file. In particular this implies that information related to elementary curves and surfaces are being used in the construction of models suitable for isogeometric analysis.

The STEP reader should currently be used for models with one solid only. The geometry will be read even if a file contains several solids, but the topology information will not necessarily be transferred to GoTools. This restriction will be removed during TERRIFIC.

3 The bivariate case

To be able to create a block structured surface model consisting of non-trimmed NURBS surfaces from a face or a set of faces, is an important ingredient in the creation of block structured trivariate models, but is also important in its own right. The result of the block structuring process applied to a face set is a model fit for 2D isogeometric analysis.

The algorithm takes a face or a face set as input and produces a block structured surface set. This is performed in the two classes RegularizeFace

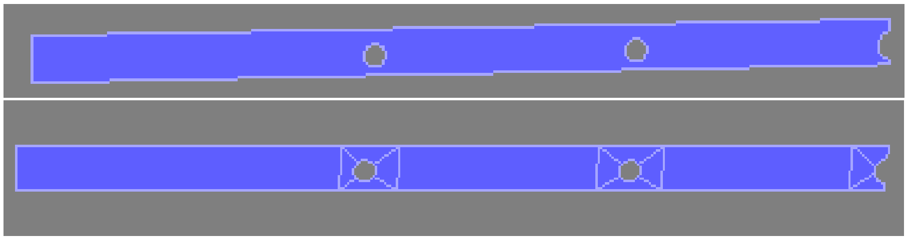

Figure 1: Block structuring of one face with holes

and RegularFaceSet. The outcome is a surface set (SurfaceModel) where no surfaces have holes and the number of corners is 4 or in a few cases 3. The algorithm is automatic and splits trimmed surfaces into several non-trimmed one. Thus, the method is not appropriate if the need is to merge small surface patches into larger patches. Moreover, this delivery is part of an ongoing work, and it is not expected that every face set will lead to a feasible solution.

The algorithm is based upon a number of rules or templates:

- Split between holes to isolate them
- Split from corners of the outer boundary to holes
- Split if the outer boundary has more than 4 corners. The split is performed from one corner towards a vertex or edge belonging to the outer boundary that is not connected to the given corner vertex
- Split in T-joints to get blocks meeting in corners
- Splitting is performed between two vertices, along parameter directions of the face, towards holes in the face or orthogonal to the face boundary. The splitting curves are defined as intersection curves between the current surface and a plane or a cylinder, or curves in the parameter domain of the surface. In the latter case, the curves will join two vertices or be a constant parameter curve in the surface.

These are general rules and does not fit in every context. It is ongoing work to identify situations where the rules should not be applied and find an appropriate solution to those cases. Moreover, there is not one unique block structuring that is correct for a given case. Consider figure 1. Here the holes in the initial face are isolated and due to the distance between them an

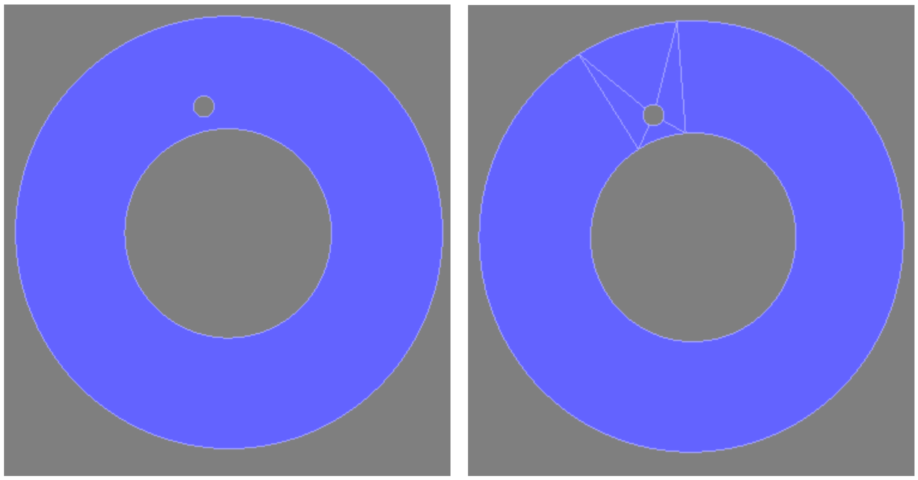

Figure 2: Holes of different sizes

extra surface is placed between the holes. If this distance were smaller, the extra surface would not be appropriate. To the right there is a feature at the boundary that can be seen as part of a hole. In this case, it makes sense to treat is at that, but this decision will be context dependent.

Figure 2 shows a case where splitting between holes to isolate them is not a feasible decision. The size difference makes that rule inappropriate. The small hole is isolated by creating splitting curves from the outer boundary to the large hole. Then splitting from corners to the small hole is performed according to the rules.

Note that the block structuring decisions are taken from purely geometric properties. The algorithm has no information about preferences in the isogeometric simulation. Still, certain aspects of mesh quality are related to geometric properties. Consider for instance figure 3. Here some splitting curves are defined as cubic curves in the parameter domain of the associated surface joining vertices at the face boundaries. Normally, these curves would be linear, but that would create degenerate blocks for this model. The block structuring of the complete boundary of a solid is shown in this figure. This model has a feasible solution in spite of the blends. In the general case, models with blends are considered to be too complex for this delivery as the appropriate solutions for blend cases are very context dependent

Figure 4 shows the block structuring of a trimmed surface with a complex outer boundary. The algorithm performs no recognition of symmetry, but

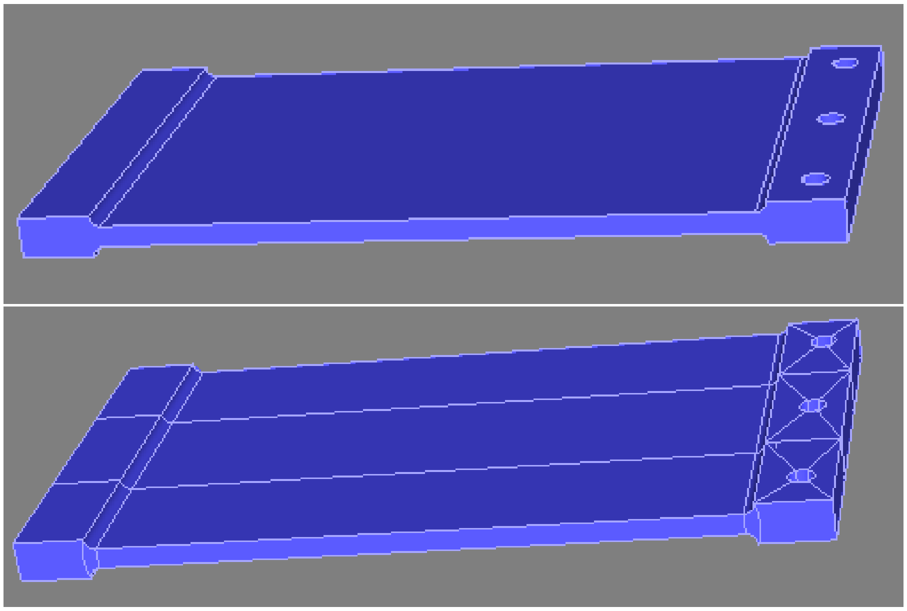

Figure 3: A complete solid

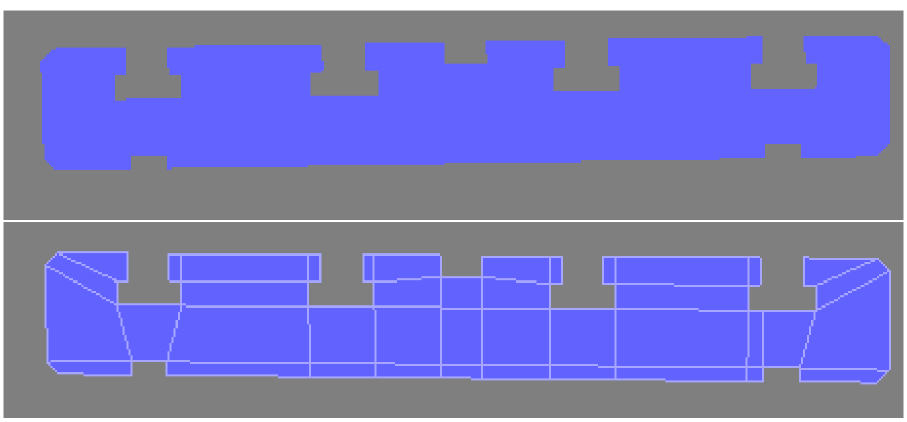

Figure 4: A face with a complex outer boundary

due to the application of the rules, the result will often be symmetric if the input is symmetric. If symmetry is crucial, however, it is recommended to split the model prior to block structuring and mirror the block structured model around the symmetry line.

Highly curved faces and face sets are more complex than single planar faces. Closed models require in addition extra care to avoid many small blocks due to splitting of surfaces in T-joints. The algorithm has some limited functionality for merging of surfaces which typically is applied to the seam of closed surfaces to avoid superfluous blocks in a situation where the position of the seam is inappropriate with relation to block structuring. Moreover, if the block structuring of the boundary surfaces of a solid is intended to be used in the context of creating a trivariate, block structured model, care must be taken such that the result is appropriate for this purpose. A consistent division of faces on either side of a model having only one block in the thickness direction, is crucial.

4 Regenerating linear swept models

Many solids, in particular in the field of construction and mechanics describe plates with holes. Such models can typically be constructed with linear sweep. The algorithm is as follows:

- Check if the model can be constructed by linear sweep
- Extract the surface or surfaces in the top or bottom of the model
- Perform block structuring of this surface set and ensure that it is represented as a set of non-trimmed NURBS surfaces where adjacent surfaces have coincident coefficients at block boundaries
- Create the volume model by linear sweep

A linear sweep sweeps a curve or a surface along a curve to create a surface or a volume, respectively. The resulting entity is created as a tensor product of the input entities. That means that an input surface will not be tilted along the given curve even if the curve is not straight. The linear sweep construction does not depend on the input geometries to be planar or linear.

The method for recognizing a linear sweep given a boundary represented solid is less flexible regarding the input. The surfaces to be swept must lie in

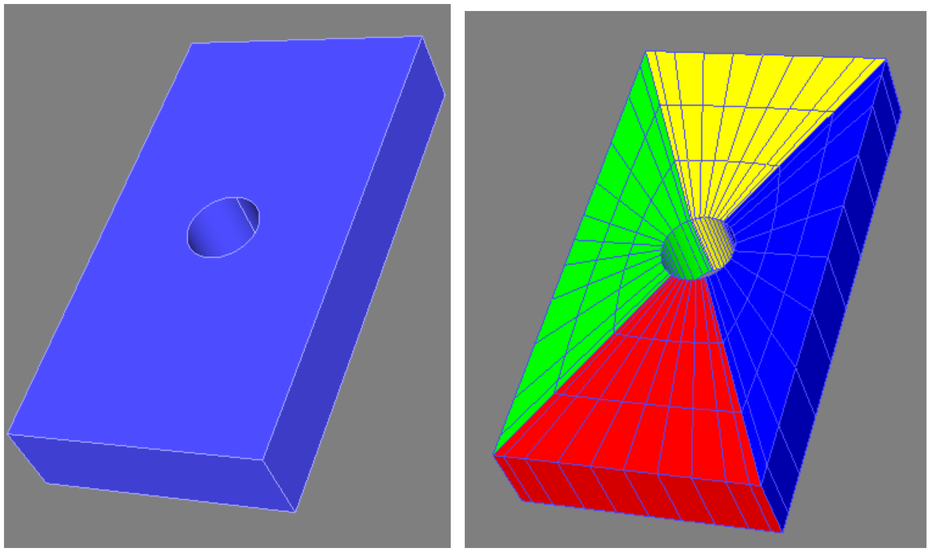

Figure 5: A linearly swept model

a plane and all other surface must either be linear and have the same plane axis or they are linear and the direction of linearity coincides with the plane normal. This implies that the curve along which the surfaces are swept, is linear. These restrictions are introduced to simplify the recognition process, but many models will, nevertheless, fit into this frame work.

The recognition of a linearly swept model makes use of information from elementary curves and surfaces. A plane is for instance linear in all directions while a cylinder is linear in one direction and a sphere is not linear. A spline surface is found to be planar if all surface normals can be found to be coincident within a tolerance. Similarly, the surface is linear is all tangents in one parameter direction are coincident. No trimming curves can be in conflict with the restrictions on a linear swept model in order to recognize the configuration. No attempt is done to recognize elementary curves and surfaces from NURBS representations or approximations of the initial geometry entity. This may be added in a later version of the GoTools libraries.

Figure 5 shows a simple model created by linear sweep from a boundary represented solid as described in this section. The volume blocks are shown in different colours and the control net of the NURBS volumes are visualized to give an impression of the amount of data in the model and also show that the requested coincidence of coefficients at surface boundaries is satisfied.

The topology is similar to the same model created by a more general method in figure 7, but the amount of data in the linear direction is much less in this case.

5 Regenerating rotational models

Rotational models frequently occur in the same application areas as the linearly swept models. These models can be constructed with rotational sweep. The algorithm is as follows:

- Check if the model is rotational and fetch information about the rotational axis
- Construct a planar surface with one boundary curve along the rotational axis. The surface must be larger than a planar section of the rotational model
- Intersect this planar surface with the boundary surfaces of the rotational model and trim it to get a section of the model described as a trimmed surface
- Perform block structuring of this trimmed surface and ensure that it is represented as a set of non-trimmed NURBS surfaces where adjacent surfaces have coincident coefficients at block boundaries
- Create the volume model by rotational sweep

Similar to the linear case, information from elementary curves and surfaces are used to check whether a surface belongs to a rotational model. The boundary surfaces must either be planar or rotational and all rotational surfaces must have the same rotational axis. Planar surfaces can be situated on the top or bottom of the rotational model or at planes going through the axis if the rotation is not complete. Incomplete rotation is, however, not tested due to lack of test cases. The task will be continued after the current delivery.

The first picture in figure 6 shows a rotational CAD model. The second picture illustrates the block structuring of the section surface extracted from this model, and in the two last pictures we can see the final trivariate model. In the last picture, the model is open up by removing two of the outer volume blocks. The model is degenerate along the mid axis.

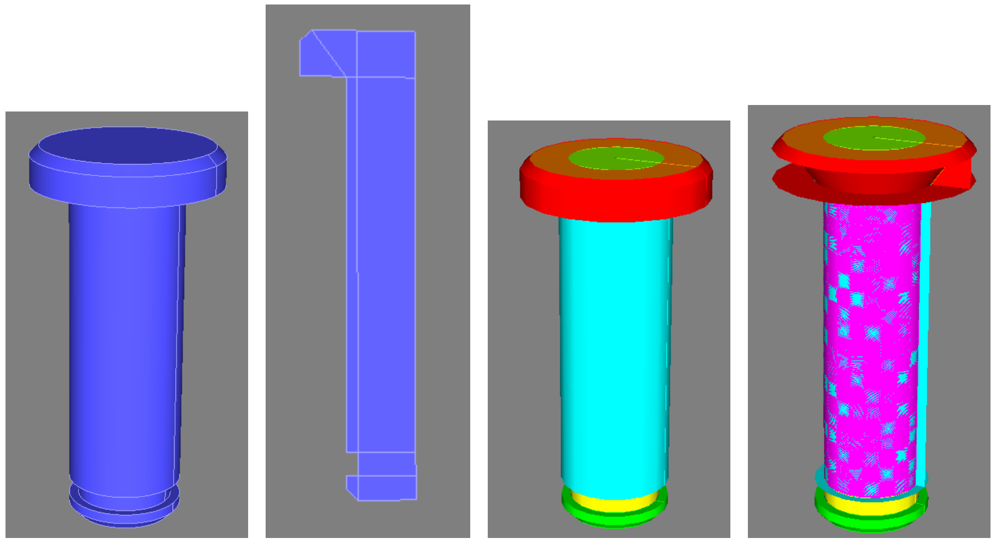

Figure 6: Block structuring a rotational model

6 The general trivariate case

In this section a more general algorithm for creating trivariate, block structured models from boundary represented solids, is presented. Still, the models have to be reasonably simple. The algorithm proceeds as follows:

- Generate a block structured version of the boundary surface set using the methodology described in section 3. The resulting surfaces do not have any inner trimming curves and the outer boundary should consist of four pieces. Most of the topological decisions regarding the number of blocks are taken at this stage.
- Extend the wire frame corresponding to the block structured surface model to apply to the wanted block structured volume model.
- Add boundary surfaces to the volume blocks in the inner of the volume model. Make sure to include sufficient topological information in the model to prepare for the next step in the process.
- Identify the boundary surfaces for each volume block and represent the volume blocks as trimmed volumes with 6 boundary surfaces, each surface having 4 boundary curves.

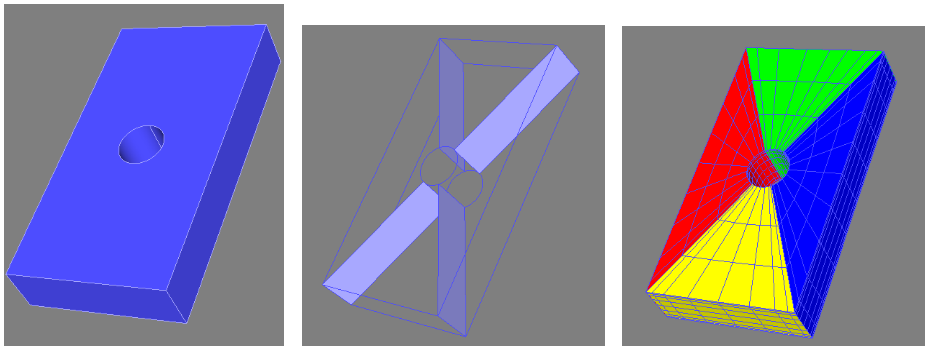

Figure 7: A simple trivariate, block structured model

- Replace the trimmed volume blocks by NURBS volumes. This process normally requires approximation. The boundary curves and the boundary surfaces are approximated while the volumes interpolates their boundary conditions.
- Ensure C^0 continuity and coincidence of boundary coefficients between volume blocks.

Figure 7 illustrates this process for a relatively simple model. The first picture shows the initial model. The second picture shows the wire frame corresponding to the block structured outer boundary and the surfaces that must be constructed to all boundary surfaced for all volume blocks. In this case the wire frame of the block structured surface model and the block structured volume model corresponds. Finally, the completed volume model is shown, the block structuring is visualized by colours.

Figure 8 shows a case where the two more dedicated methods for volume creation do not apply. The block structured model has only one block in the thickness direction, but this direction varies throughout the model. First the initial boundary represented solid is shown, second the wire frame of the block structured surface model of the boundary together with the additional edges required to generate the wire frame model corresponding to the volume model. In the next picture we can see how several internal surfaces in the volume model meet in these additional edges. These surfaces are created by first identifying the loops where a surface in the volume model is missing, and then create the surfaces in the parameter domain of an initial spline volume covering the area where the model is situated. If $V(u, v, w)$ is this spline volume

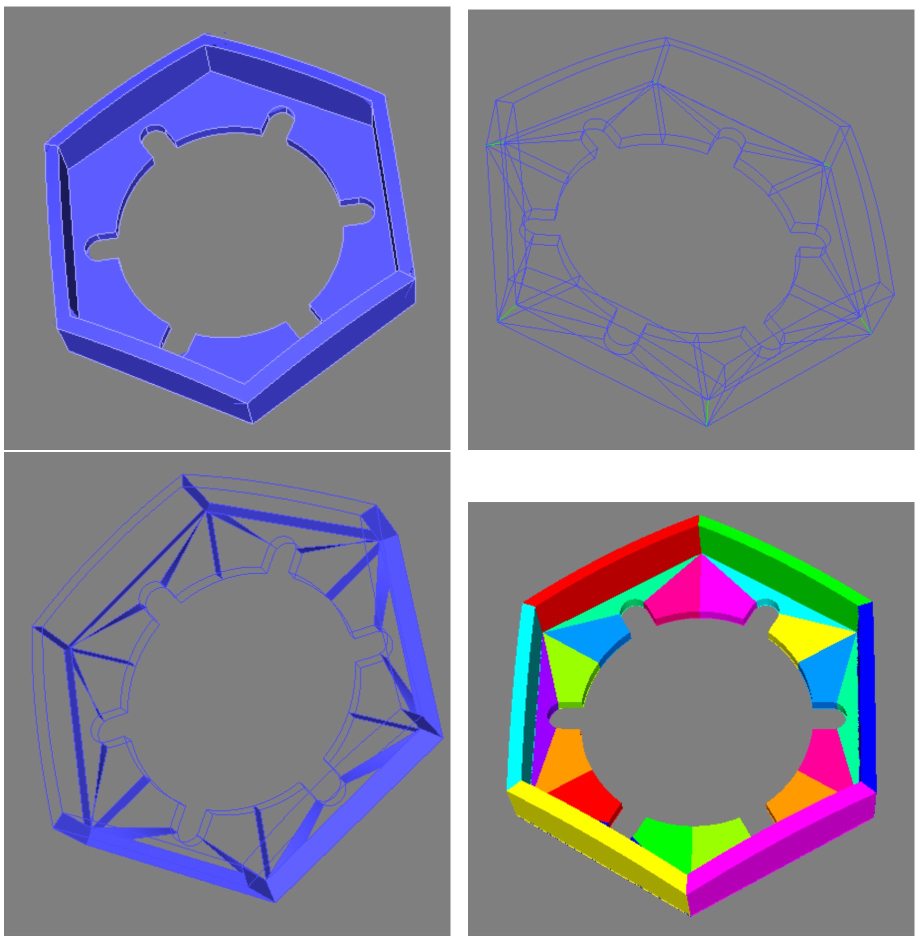

Figure 8: The process of volumetric block structuring

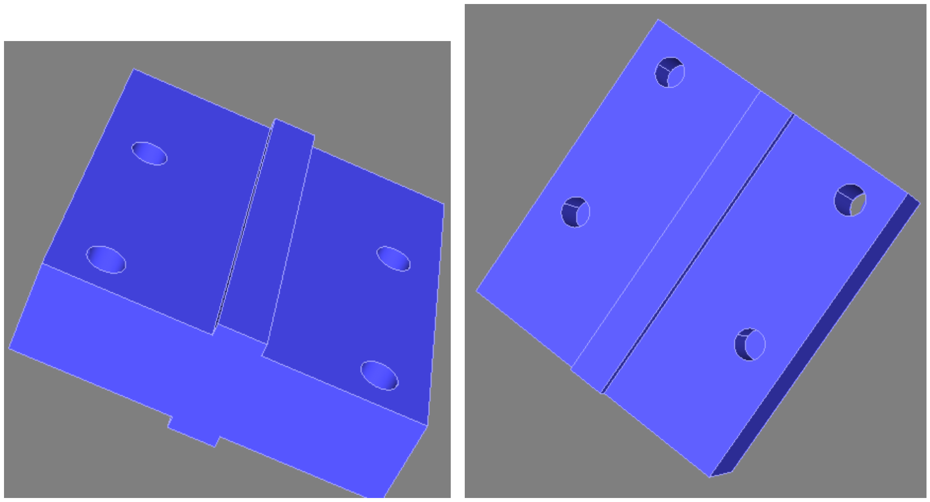

Figure 9: A more complex model

and $p(s,t) = (u(s,t), v(s,t), w(s,t))^T \in R^3$ is the surface in the parameter domain of V, then the visualized surface is $f(s,t) = V(u(s,t), v(s,t), w(s,t)$. $p(s,t)$ interpolates the curves in the parameter domain of V corresponding to geometry curves in the identified loop. The reason for creating parameter domain surfaces in this context is to avoid the surfaces to fall outside of the model when the model is curved. Finally, the final volume model is shown in the lower left picture.

Figure 9 shows a more complex model. The required amount of blocks in the thickness direction exceeds one in some areas and the number is varying throughout the model. The model has four holes, but only one of them passes through the entire model. Block structuring of the boundary is quite straight forward, and we can see the result in figure 10. Note that the topology of the block structuring of the top of the model is different to the bottom. This is due to holes not going through the model.

We identify additional edges to complete the wire frame of the volume model as before and add the extra surfaces required to construct volume blocks. However, studying figure 12, it is clear that not all blocks will be regular, i.e. they have 6 boundary surfaces and be possible to represent by a non-trimmed NURBS volume. Close to the indentations, the volume block boundaries are in them selves boundary represented solids. This indicates that a recursive approach can be useful and this is indeed the case.

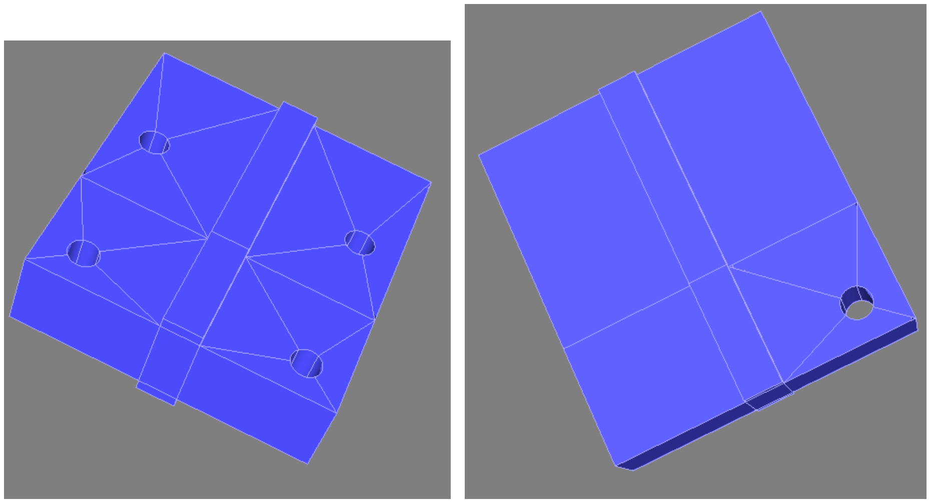

Figure 10: Block structuring the model boundary

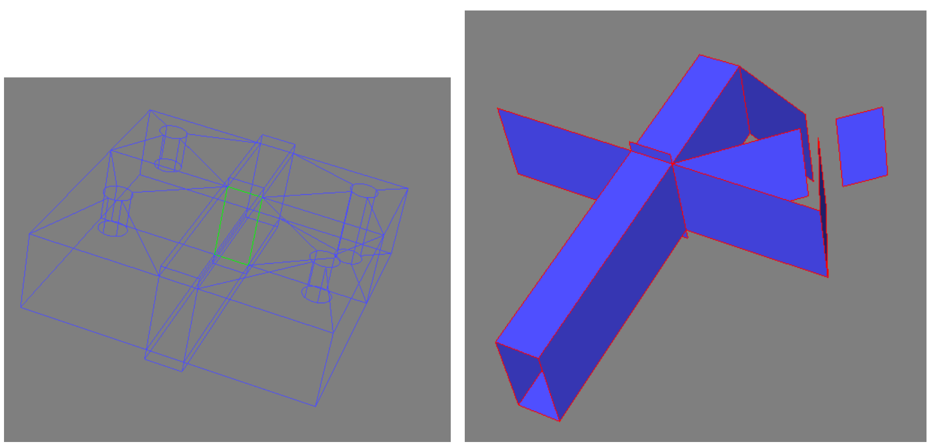

Figure 11: Block structuring of the volume model, first pass

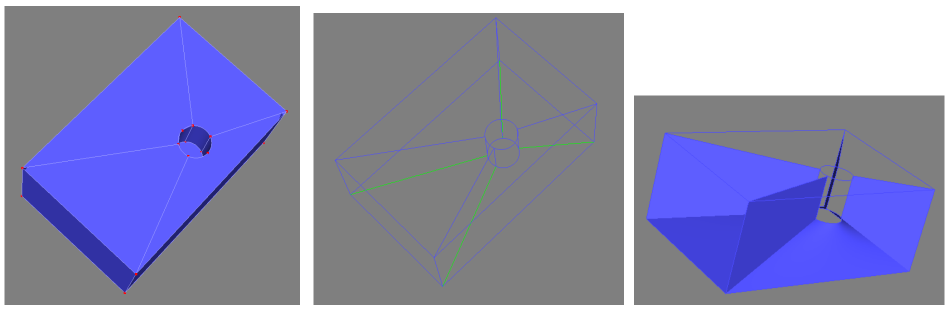

Figure 12: Block structuring of the volume model, second pass

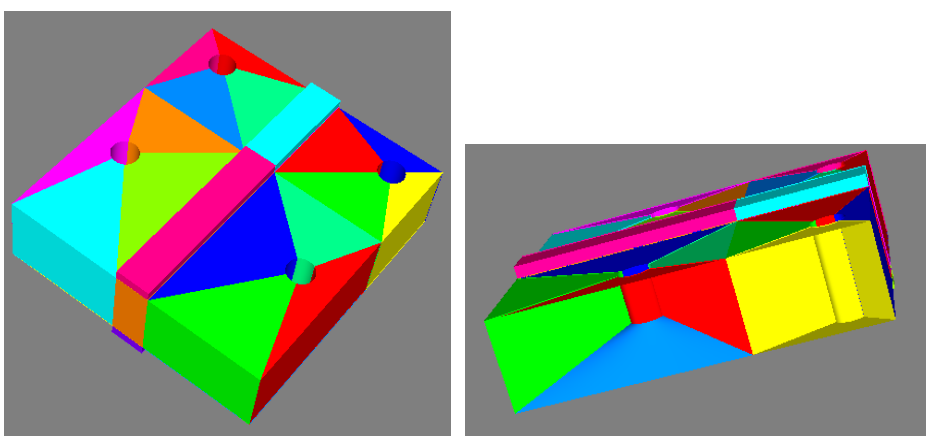

Figure 13: Final volume model

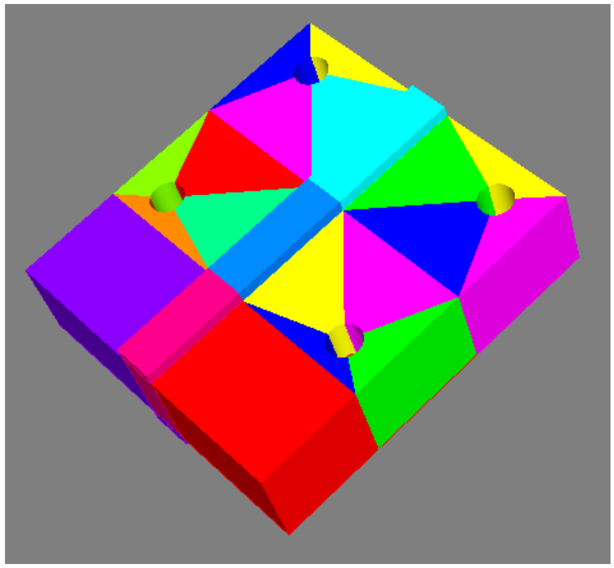

Figure 14: A different volume configuration

Figure 11 shows one sub solid close to an indentation and the volumetric block structuring of this model. The configuration for the edges missing in the wire frame of the volume model is different than what we have seen before and the criteria for the search is also different. Instead of identifying edge loops that should be divided, we look for vertices in concave areas that should be connected to vertices in convex areas. The additional surfaces are constructed as before, and figure 13 shows the concatenated result. In the second picture, some surfaces are removed to make it possible to look into the model and see the different volume configurations for the different types of holes.

Figure 14 shows a previous version of the volumetric block structuring corresponding to this model. Other decisions regarding distances from hole to the outer boundary of surfaces were taken, and this influences the entire model.

Figure 15 presents the topological data structures in GoTools. When a boundary represented model is read, all topological information with regards to surface sets are defined and in this context, also a spline volume is defined. This volume will be larger than the model itself and together with the boundary surface model, it will describe a trimmed volume represented as a ftVolume. During the process of model transformation, the topology

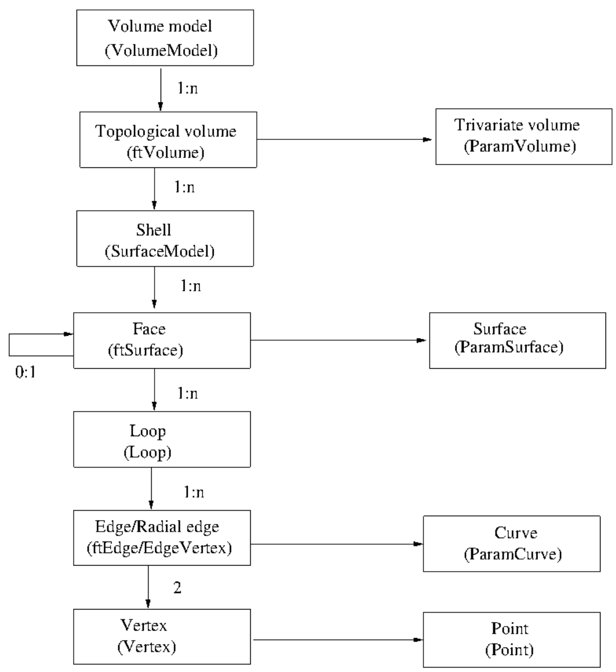

Figure 15: Topological data structures for volume models

information is always maintained. When the internal surfaces in the volume model is created, they will have two instances with different orientation and the adjacency between these two surfaces are stored in the topology model. Furthermore, each surface is associated with the block it will belong to, which allows us to extract the different blocks in the next step of the process.

When the block topology is defined, it is time to generate the NURBS representation of the block. At this stage, each block is surrounded by 6 4-sided surfaces, but the block is still represented as a trimmed volume. The geometry generation process is approximative. All boundary curves of the boundary surfaces associated to the same parameter direction in the non-trimmed volume block are approximated in the same spline space using a least squares approximation with a smoothing term. The boundary surfaces are generated as Coons patch surfaces between these boundary curves and updated using a similar approximation method as for the curves. If the twin surface of a boundary surface internal to the model is already defined as a NURBS surface, this information is reused. The volume is defined by a Coons patch type method interpolating its boundaries.

The surface set assembled from all outer boundary surfaces of a block structured volume model is a boundary represented solid. Thus, a CAD model can trivially be fetched from a volume model. It is, however, not a typical CAD model. All surfaces are non-trimmed NURBS surfaces and the positional continuity between the surfaces is exact.

7 Outlook

The current delivery contains functionality to transfer some boundary represented solids to block structured, trivariate models. Still, many models cannot be handled. We will continue the work and extend the category of allowed models. The effort will be in the following areas:

- Models with blends. All types of blends should not be handled identically. In some cases it makes sense to isolate the blend, in other the blend should be connected to one of the adjacent surfaces or split in the middle and connected to both. Thus, the introduction of models with blends will be gradual.

- Models with higher complexity. In particular, we have concentrated on models with complexities only on one direction. We will start to look

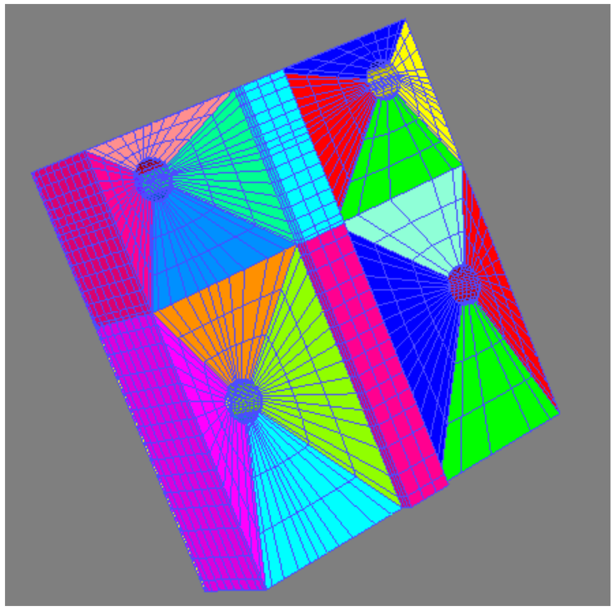

Figure 16: The control net corresponding to a block structured volume model

at models with features (holes, indentations, extra material) in more than one direction and combinations of different types of features.

- Models consisting of more than one solid. Each solid will be handled separately, but information about block structuring in one solid will be used in block structuring of adjacent solids.

In cooperation with work package 5, we are involved in an activity to extend the STEP standard to be able to store block structured, trivariate models. This work will continue.

Figure 16 gives an illustration of the data size of one of the volume models we have seen previously. In order to have correspondence between coefficients from adjacent volume blocks at block boundaries, a high data size due to some details in the model, will spread through the entire model. This, is not a problem for small models, but for models of this size and larger, it can lead to high data sizes. To eliminate this problem, we will introduce the concept of **LR B-splines**. LR B-splines allows for local refinement of the spline space, a concept that is advantageous both for geometry construction and isogeometric simulation. The first use of the new representation format in model construction will be to avoid high data sizes to spread from one block to another.

8 Summary

This delivery is dedicated to the construction of trivariate models from CAD models and is the first delivery of this functionality associated to the TERRIFIC toolkit. The delivery is included in the GoTools libraries, and the functionality will be extended in the deliveries in M18 and M36.

The current version of the tools has a number of restrictions on the type of input models that can be handled and may also lack generality as one model may be handled satisfactorily while another model that possess many of the same properties may not. The restrictions are in this report presented in the context of the method where it apply. The restrictions will be released during the project period.