HORIZON 2020

ANALYSIS TOOLS FOR AM, LINEAR SETTING

Deliverable D2.3

Circulation:	PU: Public
Lead partner:	CNR-IMATI
Contributing partners:	SINTEF, CNR-IMATI, Fraunhofer IGD, Missler, STAM, Novatra
Authors:	Rafael Vazquez, Lorenzo Tamellini, Massimiliano Martinelli, Hongmei Kang, Federico Marini, Paola Pietra, Vibeke Skytt, Jon Hjelmervik, Oliver Barrowclough, Christian Altenhofen, Stefano Ellero, Tommaso Zerbi, Umberto Battista, Jean-Claude Morel, Luc Bouffin
Quality Controllers:	Eric Neiva, Georg Muntingh
Version:	1.0
Date:	24.12.2016

©Copyright 2015-2018: The CAxMan Consortium

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 Istituto 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

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the CAxMan Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or 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

Version[1]	Issue Date	Stage	Content and Changes
0.9	12.12.2016	100 %	Ready to be submitted to Quality Control
1.0	24.12.2016	100 %	Ready to be submitted to Project Officer

[1] Integers correspond to submitted versions

EXECUTIVE SUMMARY

One of the aims of WP2 within the CAxMan project is to provide the end-user with a seamless tool to perform the thermal and mechanical analysis of the object to be manufactured, using Isogeometric Analysis (IGA). If the predictions meet the requirements of the user on the final performance of the object to be printed, it is possible to proceed with the follow-up stages of the production, which are discussed by WP3, 4, 5, and 6. The performance requirements on the test-cases considered in CAxMan are analysed in WP7.

Performing the IGA-based analysis means in practice to solve with a numerical method a certain partial differential equation, which is performed in CAxMan by IGATools, a C++ library developed at CNR-IMATI. In this deliverable we introduce the equations that CAxMan users might be interested in solving and we detail the steps needed to perform this analysis, especially the problem-dependent inputs that need to be provided by the users: to this end the necessary interactions with other software, specifically TopSolid by Missler and GoTools by SINTEF, are discussed in this deliverable. Then, we show the kind of results that can be obtained with the technology developed by now within the project and discuss the next developments to extend its capabilities.

The content of this deliverable is mostly related to Task T2.4, but some updates on the status of tasks T2.1 (Interoperability to CAD) and T2.2 (Tri-variate shape representation for design and analysis) will be given.

TABLE OF CONTENTS

1 INTRODUCTION

One of the aims of WP2 within the CAxMan project is to provide the end-user with a seamless tool to perform the thermal and mechanical analysis of the object to be manufactured, using Isogeometric Analysis (IGA) (this must not be confused with the computer-based simulation of the printing process itself, which is the focus of WP4). If the predictions meet the requirements of the user on the final performance of the object to be printed, it is possible to proceed with the follow-up stages of the production, which are discussed by WP3, 4, 5, and 6. The performance requirements on the test-cases considered in CAxMan are analysed in WP7.

For the use cases considered for CAxMan (the mould produced by Novatra and the gearbox produced by STAM, see WP7), typical quantities to be predicted are the temperature/heat exchange and the mechanical stress of the object under study[2]. Upon having chosen the physical phenomenon of interest to be analysed, performing a computer-based analysis means in practice to write the problem in terms of a mathematical model, usually in the form of a partial differential equation, and to numerically solve the equation with a suitable approximation method. Within CAxMan, the method of choice is Isogeometric Analysis (IGA), which is a quite recent method first proposed in 2005, see (Hughes, Cottrell, & Bazilevs, 2005) and (Cottrell, Hughes, & Bazilevs, 2009). It is closely related to Finite Element Analysis but is more suitable for seamless integration in the framework of Computer-based design, because it can operate directly over a CAD-like file that satisfies certain requirements. The generation of IGA-suitable CAD files has been thoroughly discussed in Deliverables D2.1 and D2.2. The software library that will perform the Isogeometric Analysis for the CAxMan project is IGATools, developed at CNR-IMATI Pavia, which in turn relies on the GoTools library by SINTEF and on the Missler software TopSolid to generate IGA-suitable CAD files.

Besides choosing the equation to be solved and feeding as input an IGA-suitable CAD file, the user must provide some additional information needed to perform the simulation, and that is related to the properties and the working conditions of the piece to be analysed. More specifically, it is necessary to know the kind of material of which the object is made (in the framework of CAxMan, this will be any of the metals that the 3D printer can handle), and the boundary conditions and volumetric forces to which it is subject. The relevant required information and the possible boundary conditions depend on the problem at hand: for instance, in a thermal analysis problem one needs to specify whether the border of the object is at room-temperature, or whether it is in contact with a heat source.

Once the results of the simulation are available, the user is responsible of taking the decision on whether to accept the design of the piece, and proceed towards the manufacturing process, or modify the design. To take this decision, the analysis results must be exported in a user-friendly way, either as a tabular/chart or as full evaluation of the state variables (temperature, stress, etc.) over the piece to be analysed in a suitable visualization software.

After the discussion of the IGA-based shape representation in D2.2, in this report we describe in detail the necessary information and the sequence of operations to run an IGA-based simulation, with special attention to the required interactions between the different software libraries. This

[2] Although we focus on thermal and mechanical stress, the same procedure can be replicated for any other kind of analysis.

sequence of operations (geometry and data import, solution of the equation by IGA, result visualization and output) constitutes the part of the WP2 workflow highlighted in green in Figure 1. We note that the document is focused on running a single simulation, thus we do not consider any optimization loops (neither shape optimization nor voids placement). We also remark that in this deliverable we are focusing on the linear setting, while the nonlinear setting will be discussed in the future deliverable D2.5. For the examples considered below, the nonlinear setting would correspond to more complex equations, such as large deformations in elasticity, temperature-dependent properties in the heat equation, and Navier-Stokes flow.

The rest of this deliverable is organized as follows: Section 2 gives more details about the mathematical models of the problems that are of interest for CAxMan use cases. Sections 1 and 1 respectively detail the inputs needed to perform the computer-based analysis and the outputs that are returned to the user and the visualization of the results, and the kind of interaction that is expected between the different software libraries involved in this workflow. Finally, Section 1 gives some detail on how the software interaction has been developed so far, on the analysis that can already be performed with the software currently available and discusses future extensions.

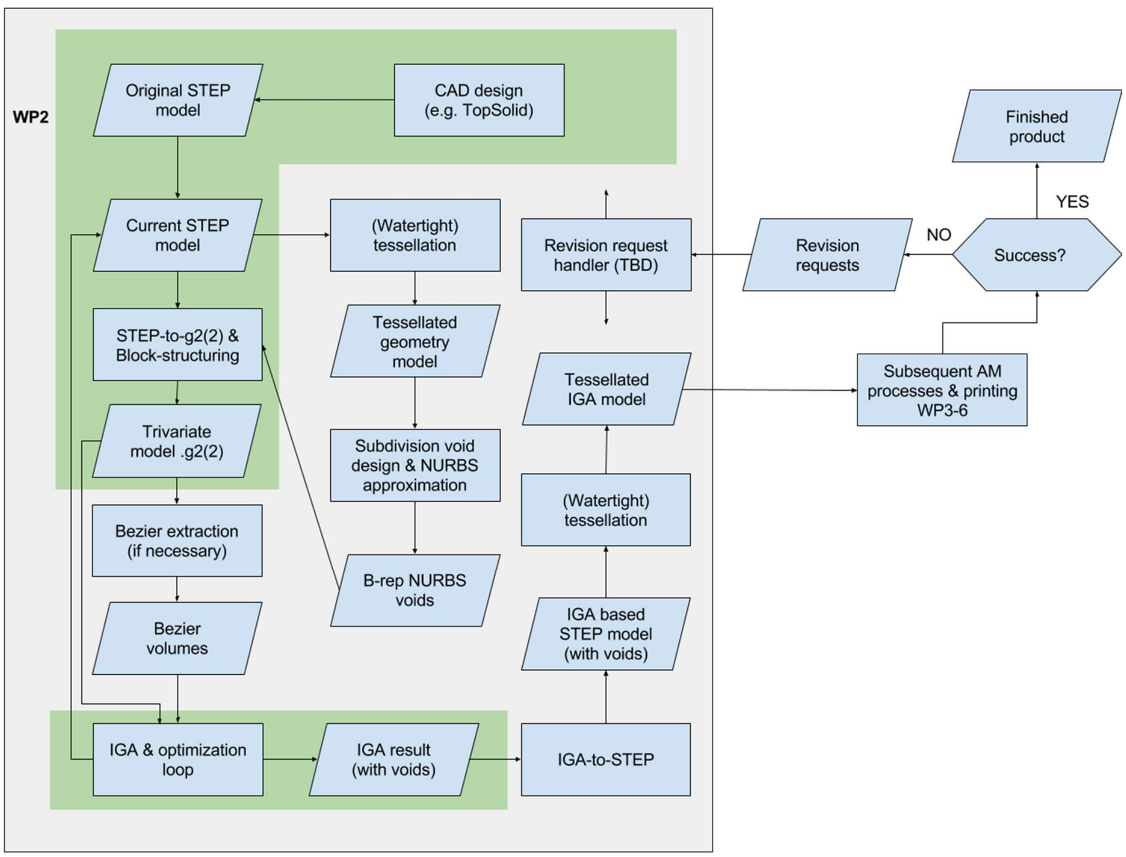

FIGURE 1: WP2 WORKFLOW. THE PORTIONS DISCUSSED IN THIS DELIVERABLE ARE HIGHLIGHTED IN GREEN.

2 MATHEMATICAL MODELS OF THE PROBLEMS

We collect here a list of mathematical problems, that we solve within CAxMan using Isogeometric Analysis (IGA). These problems are model problems selected as a first step towards the analysis of the use cases, and as a first benchmark for the interoperability between the different software libraries used in WP2. As mentioned in the previous section, they are all linear differential equations. Solving these equations entails solving a certain linear system whose entries are computed by certain integrals over the computational domain. This is a standard operation that can be found in any basic textbook on Finite Element Analysis, see e.g. (Cottrell, Hughes, & Bazilevs, 2009), and will not be detailed here. The software and geometry requirements to perform this integration is specified in deliverable D2.1.

2.1 STATIONARY HEAT EQUATION

The first problem that we consider is the stationary (i.e., not time-dependent) heat diffusion equation. The equation of the problem is

$$-\nabla \cdot (k \nabla T) = f$$

where T is the temperature, k is the thermal conductivity, and f is a given heat source. This equation must be complemented with suitable boundary conditions, that can be of three different types: prescribing a given temperature (Dirichlet type), its derivative, i.e., a heat flux (Neumann type), or prescribing the external temperature and the heat transfer rate, i.e., a convection condition (Robin type). These are respectively expressed by

$$\begin{aligned} &T = T_b, \\ &k \frac{\partial T}{\partial \mathbf{n}} = -q, \\ &k \frac{\partial T}{\partial \mathbf{n}} = \alpha (T - T_c), \end{aligned}$$

with T_b the given temperature on the Dirichlet boundaries, q the given heat flux on the Neumann boundaries, and for the convection condition T_c is the ambient temperature, and α is the heat transfer coefficient, which may depend on different parameters (the size and position of the part, the outer material, etc.).

The thermal conductivity depends on the material, and it may also depend on the temperature, that is, $k = k(x, T)$. The dependence on the temperature of k will be neglected for now because it is a secondary effect that would moreover transform the equation into a nonlinear one; the dependence on x might instead be used to describe graded materials so it is useful to retain it, in order to support expected future developments in 3D printing hardware. Note also that deriving an explicit function that details how the thermal conductivity of a graded material changes along the object is not an easy task, which we nonetheless assume can be performed. Finally, we also note that the thermal (and mechanical) properties depend not only on the kind of material used by the 3D printer, but on the printing process itself. The nominal value is usually given by the printer producer.

The physical properties and data needed to run a simulation are summarized in Table 1, with the corresponding units. The data are separated in three groups: material properties, volumetric

sources, and boundary conditions. It is convenient to separate the data in these three blocks, since this is related to the kind of information that has to be passed from the CAD model interface to the IGA solver, which we will see in Section 1.

Property	Units	Notation
Material properties		
Heat Conductivity	W/m K	k
Volumetric sources		
Volumetric heat flux	W/m³	f
Boundary Conditions		
Fixed Temperature	K (or ˚C)	T_b
Heat flux	W/m²	q
Heat transfer coefficient	W/m² K	α
Room temperature	K (or ˚C)	T_c

TABLE 1: PHYSICAL PROPERTIES AND DATA FOR A STATIONARY HEAT EQUATION.

2.2 TRANSIENT HEAT EQUATION

The stationary version of the heat equation in the previous section is mainly intended to serve as a minimal benchmark problem, to implement and test all the necessary functionality to run a simulation. However, for the use cases of WP7 we solve the heat transfer equation in a transient regime, including both conduction and convection. The equation for heat transfer is

$$\rho c_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) + \mathbf{v} \cdot \nabla T = f, \quad x \in \Omega, \quad t \in [0, t_f]$$

where ρ and c_p denote the mass density and the specific heat capacity at constant pressure, respectively, and $\mathbf{v}$ is the velocity in the convection term (valid for heat equations in gases or fluids). The velocity can be given as a datum, or, if the problem is coupled with fluid dynamics, it should come as the solution of a fluid flow problem.

The boundary conditions in this case are exactly as in the previous section, and are not detailed. Since this is a transient problem, it is necessary to provide an initial temperature, that we denote by T_0, and a final time to end the simulation, that we denote by t_f. We gather all the relevant information in Table 2.

Property	Units	Notation
Material properties		
Mass density	Kg/m³	ρ
Specific heat (constant press.)	J / kg K	c_p
Convection velocity	m/s	$\mathbf{v}$
Initial Conditions		
Initial temperature	K (or ˚C)	T_0
Final time	s	t_f

TABLE 2: ADDITIONAL PHYSICAL PROPERTIES AND DATA FOR A TRANSIENT HEAT EQUATION.

2.3 LINEAR ELASTICITY

For the computation of solid mechanics in the linear setting, we solve the linear elasticity problem in the case of small deformations (large deformations would need a non-linear equation instead):

$$-\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{f}$$

where $\mathbf{u}$ is the displacement vector, $\mathbf{f}$ is a given volume force, and $\boldsymbol{\sigma}(\mathbf{u})$ is the stress tensor, which is given as a function of the displacement, in the form:

$$\boldsymbol{\sigma}(\mathbf{u}) = 2\mu\epsilon(\mathbf{u}) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I}$$

where λ and μ are the Lamé parameters, and $\epsilon(\mathbf{u})$ is the strain displacement tensor

$$\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^{\top}).$$

In the engineering literature it is more common to write the equations in terms of Young's modulus E and the Poisson ratio ν, which are related to the Lamé parameters by

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \mu = \frac{E}{2(1+\nu)}$$

As in the previous case, this equation must be complemented with suitable boundary conditions, this time of two different types only: a condition of given displacement (Dirichlet type), and a condition of a given external force (Neumann type), which are respectively written as

$$\mathbf{u} = \mathbf{u}_b,$$
$$\boldsymbol{\sigma}(\mathbf{u})\mathbf{n} = \mathbf{g}.$$

Assuming that the external force is normal to the boundary, it is also possible to give only a scalar pressure value, and the force is then given by $\mathbf{g} = p\mathbf{n}$. We summarize in Table 3 the required data to run the simulation.

Property	Units	Notation
Material properties		
Young's modulus	Pa	E
Poisson ratio	--	ν
Volumetric sources		
Volumetric force	N/m2	$\mathbf{f}$
Boundary Conditions		
Fixed displacement	M	$\mathbf{u_b}$
External force	N/m2	$\mathbf{g}$
Surface pressure	N/m2	p

TABLE 3: PHYSICAL PROPERTIES AND DATA FOR A LINEAR ELASTICITY EQUATION.

2.4 FLUID DYNAMICS: STOKES EQUATIONS

The simulation of fluid dynamics could be useful when designing the mould cooling channels. In the linear setting, we consider the equations for an incompressible Stokes flow, which are given by the momentum conservation equation and the mass conservation equation:

$$\rho \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot (\mu \nabla \mathbf{v}) + \nabla p = \mathbf{f},$$
$$\nabla \cdot \mathbf{v} = 0,$$

where ρ is the mass density, μ is the dynamic viscosity, $\mathbf{f}$ is a given external force, and the unknowns are the velocity $\mathbf{v}$ and the pressure p. The Stokes equations are the linear approximation of the more general (nonlinear) Navier-Stokes equations, which describe motion of fluids; this approximation is good when viscous effect are dominant and the fluid motion is smooth and constant; conversely, it is rather inaccurate when the fluid motion presents swirls and vortices. More precisely, the Stokes approximation is good for flows at small values of the Reynolds number, which is a quantity defined as $Re = (\rho U d)/\mu$ where U is the ``characteristic velocity'' of the problem, and $d$ is a ``characteristic length'' (e.g. for flow in pipes, the diameter of the pipe).

From the mathematical point of view, we consider two kind of boundary conditions: Dirichlet conditions, where the velocity is fixed, and Neumann conditions, where an external force is given. That is,

$$\mathbf{v} = \mathbf{v}_b,$$
$$\nabla \mathbf{v} - p\mathbf{n} = \mathbf{g}$$

In the use case problems, and in particular for the simulation of the cooling channels, a Dirichlet condition will be used at the inlet with $\mathbf{v}_b = \mathbf{v}_{in}$, and also on the walls, with $\mathbf{v}_b = \mathbf{0}$. A Neumann condition will be considered at the outlet, with $\mathbf{g} = \mathbf{0}$.

Apart from the boundary conditions, it is also necessary to give an initial velocity, that in general will be $\mathbf{v}_0 = \mathbf{0}$, and a final time to end the simulation, t_f. These data are only required in the time-dependent version of the Stokes problem, and are not necessary in the stationary case. All the required quantities are summarized in Table 4.

Remark: At the inlet it is possible to consider, instead of a given velocity, a given flow rate. In this case the inlet velocity profile $\mathbf{v}_{in}$ will be computed from the flow rate.

Property	Units	Notation
Material properties		
Mass density	Kg/m^3	ρ
Dynamic viscosity	Pa s	μ
Volumetric sources		
Volumetric force	N/m^3	$\mathbf{f}$
Boundary Conditions		

Given velocity	m/s	Inlet: $\mathbf{v}_b = \mathbf{v}_{in}$ Walls: $\mathbf{v}_b = \mathbf{0}$
Stress	N/m^3	$\mathbf{g}$
Initial Conditions		
Initial velocity	m/s	$\mathbf{v}_0$
Final time	s	t_f

TABLE 4: PHYSICAL PROPERTIES AND DATA FOR STOKES EQUATIONS.

3 STEPS AND REQUIREMENTS TO RUN A SIMULATION

The Isogeometric Analysis described in the report is performed in IGATools, but requires inputs from TopSolid and GoTools. As these tools are already in an integration process, we have decided to pass information directly between them in a simple file in XML format, which could be removed when full integration is achieved. Alternatively, AP209 provides the structures required to store all relevant information.

3.1 GEOMETRY DESCRIPTION

The first requirement to run a simulation with IGA is a trivariate representation of the geometry, either with polynomial splines or with NURBS. This representation is closely related to the initial CAD geometry description, and the conversion from CAD to trivariate representation can be obtained by means of SINTEF's GoTools library, which is extended within CAxMan. The details about the geometry representation, and the kind of B-spline manipulations needed to run a simulation (e.g., evaluating a B-spline and its gradient, assessing whether a point is inside or outside the computational domain, etc) were already explained in the deliverable D2.2, and this is not repeated here.

We simply recall that two different geometry representations have been proposed: the multiblock representation, formed by the union of several blocks, each one the image of the unit cube; and the trimmed representation, in which the unit cube is cut by one or several orientable surfaces. A multiblock representation is preferred whenever possible. It is also possible to have a combination of both, where the domain is defined as the union of several blocks, each one the image of the unit cube, and eventually trimmed.

In the following, we explain the requirements in the particular case of a multiblock geometry, and some details for the trimmed case will be added at the end of each subsection.

3.2 CHOICE OF THE PHYSICAL PROBLEM

The second step is the choice of the problem to solve, among those described in Section 2. The choice of the physical problem will determine the physical quantities and the parameters that need to be passed from the user interface (TopSolid) to IGATools, sometimes passing also through GoTools.

Although the information needed is different for each problem, we can classify it in three different types that are valid for all the problems:

- Material of which each block of the object is composed, and material properties.
- Boundary conditions.
- Volumetric sources and loads.

We will detail in the following how this information is provided, using as an example the stationary heat equation, which is the simplest case.

3.3 MATERIAL OF EACH COMPONENT AND MATERIAL PROPERTIES

Most of the physical properties will depend on the chosen material, like the thermal conductivity in our model problem. The user needs to specify the material properties for each subdomain of

the piece, which may be either defined directly by the user, or read from a database. These are the two kind of interactions that should be implemented:

- Assign a material ("name tag") to each block. Since the trivariate block partitioning is generated in GoTools, this requires TopSolid → GoTools → IGATools interaction.
- Give a value to the physical properties for each material: TopSolid - IGATools interaction.

Concerning the first interaction, only one material should be assigned to each block of the multiblock trivariate model. This is an additional constraint that should be added to the partitioning algorithm used by GoTools to decompose the trivariate model into elementary blocks. The conforming conditions between blocks explained in D2.2 still apply, and should also be satisfied between blocks of different materials. GoTools should assign a material name (or a number) to each block, and pass this information to IGATools. For instance, if the object in Figure 2 top-left is made of just one material, then a valid block-decomposition consists of the 5 blocks in Figure 2 top-right. Conversely, if the bottom-left corner is made of a different material (bottom-left plot in Figure 2) the previous block decomposition must be changed as in Figure 2 bottom-right.

The material properties are clearly independent of the geometry; thus the communication can be done directly from TopSolid to IGATools. TopSolid should associate to each material a value of all the required physical quantities, to be read by IGATools. In principle, the choice of the materials does not add further restrictions on trimmed geometries. That is, the trimming surface can intersect blocks made of different materials, without constraints.

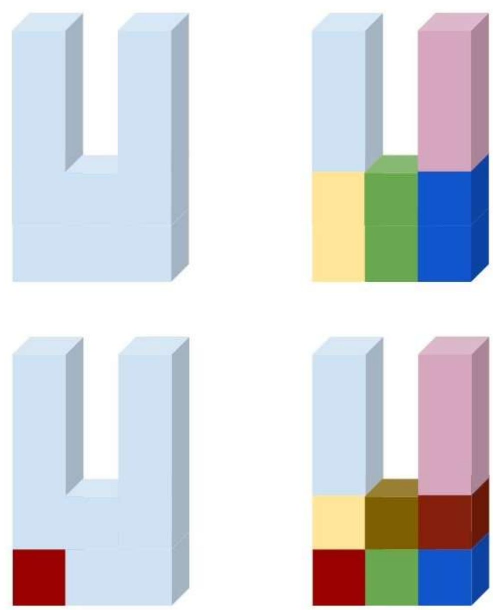

FIGURE 2: MULTIBLOCK PARTITIONING MUST TAKE INTO ACCOUNT REGIONS WITH DIFFERENT MATERIALS.

3.4 BOUNDARY CONDITIONS

The next step is the imposition of suitable boundary conditions. All the required information to impose the boundary conditions should be defined by the user in TopSolid. These are the steps for the imposition of boundary conditions:

- Assign a reference number (or tag) to each boundary condition defined in TopSolid.
- Assign a boundary type (Dirichlet, Neumann, Robin), along with the quantities needed by the boundary condition, to each reference number/tag. (For instance, the temperature T_b for Dirichlet boundary condition.) In other words, two boundary conditions of the same kind but with different values must be treated as different (see Figure 3). This information only needs TopSolid → IGATools interaction.
- In the trivariate geometric model, assign a reference number to each boundary face. It is important to note that only one reference number can be assigned to each boundary face of a block, in the same way that each block must be of the same material. This is an important additional constraint in the generation of the trivariate multiblock model. This needs TopSolid → GoTools → IGATools interaction.

We note that boundary sides of different blocks can share the same reference number/tag. For instance, all the blue faces in the example of Figure 3 can be associated to the same reference number, even if they belong to five different blocks.

Boundary conditions of the same type, but with different properties, should be given different tags. For instance, in Figure 3, to impose a temperature of 30 ˚C on the green face, and 80 ˚C on the red face, the two faces must be assigned different references, even if both are of Dirichlet type.

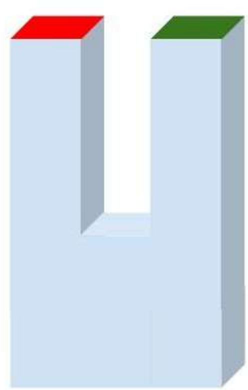

FIGURE 3: THE RED FACE AND THE GREEN FACE HAVE DIFFERENT TEMPERATURE, AND THEREFORE THEY COUNT AS TWO DIFFERENT BOUNDARY CONDITIONS.

For trimmed geometries, boundary conditions are likely to be imposed on a trimming surface. In this case, the reference number or tag should be associated to the trimming surface, while the rest of the information is exactly as for a multiblock geometry. If two different boundary conditions are applied on different parts of the trimming surface, the surface should be split in two (or more) parts, and a different tag be associated to each part.

3.5 VOLUMETRIC SOURCES AND LOADS

In some cases, it is necessary to add a volumetric source or load to the equation, like the term f in the stationary heat equation. The way to impose these sources is very similar to that of boundary conditions and materials above, and therefore it is not explained in detail.

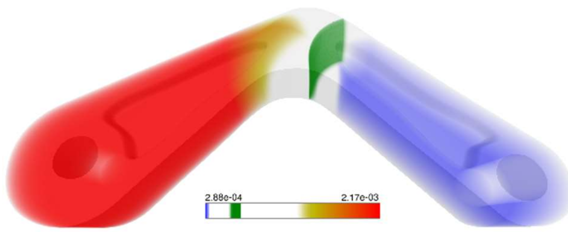

FIGURE 4: SCREENSHOT OF ISOVOLVIZ BY SINTEF. THE OBJECT SHOWN IN THE PICTURE IS THE DEMONSTRATOR OF THE ``TERRIFIC'' PROJECT.

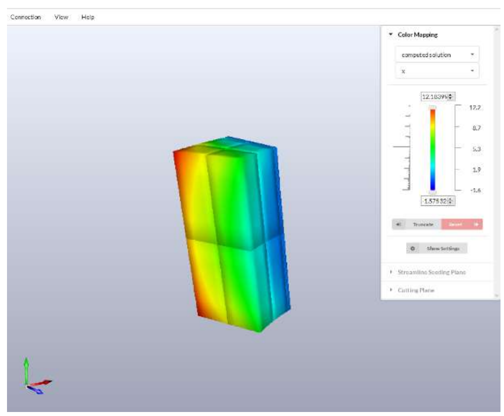

FIGURE 5: SCREENSHOT OF THE CLOUDFLOW REMOTE POST-PROCESSING SERVICE.

Note however that the source and the materials are not necessarily related. One could have a source applied in a region made of different materials, but also have a piece made of one single material, with a small source region in it. GoTools should take into account both constraints when generating the trivariate multiblock model. Like for the materials, the presence of volumetric loads does not add any constraints for trimmed geometries. We remark that, in principle, volumetric terms will not appear in the use cases of CAxMan.

4 OUTPUTS

In order to understand the results from the Isogeometric Analysis, there is a need for efficient visualization tools that allow the user to visually inspect the results. Due to the volumetric nature of the use cases, the tools need to be able to show the inside of the model (volumetric visualization) as well as the analysis results on the boundary of the model.

The IsoVolViz software by SINTEF (Fuchs & Hjelmervik, 2016), see Figure 4, offers both boundary and volumetric visualization using an isogeometric model directly, without first converting the model to triangles / tetrahedra. The solution supports multi-block models, without trimmed blocks. The functionality will be extended based on the needs from the use cases.

As an alternative, the Remote Post-Processing service introduced in the CloudFlow project can be used to visualize simulation results, see Figure 5. It offers post-processing operations like color mapping, defining cross sections and seeding streamlines in a web-based client user interface without the need of downloading the files to a local machine. To use this service, the IGA simulation results will be converted into a VTK format beforehand. CGNS and OpenFOAM formats are also readable by the service.

4.1 VISUALIZATION OF NUGEAR SIMULATIONS RESULTS

For the development of the NUGEAR, STAM is mainly interested in mechanical simulations of the stresses generated during the teeth meshing. To date, the main simulations that STAM performs on the NUGEAR during the design phase are:

- FEM structural on the gear teeth to verify resistance to nominal load;
- FEM structural nonlinear, to verify teeth deformation in case of massive overload (crash load), including strain localization and cracking simulation.

For the development of IGA simulations replacing the FEM analyses currently performed, the first case to be analysed is that of static and linear (small deformations) analyses. Since the contact phenomena will be modelled in the future, the starting point will be based on static load. A small portion of a bevel gear (namely, one tooth flank, see Figure 6) will be loaded with a pressure distribution. This distribution was computed by STAM in order to replicate a typical pressure distribution obtained by FEM simulations (Figure 7).

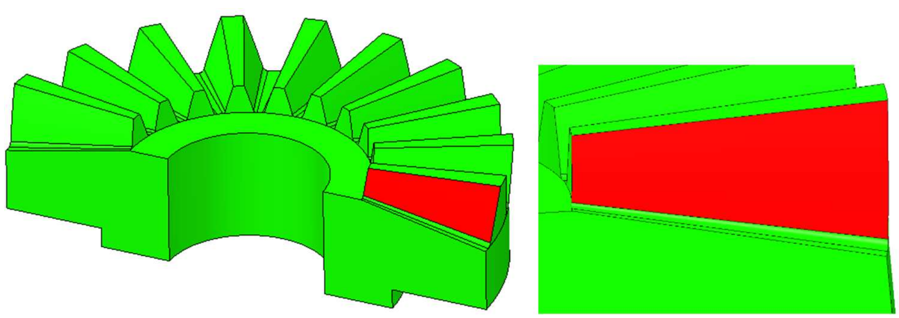

FIGURE 6: TOOTH FLANK OF A GEAR (LEFT) AND DETAILED VIEW (RIGHT).

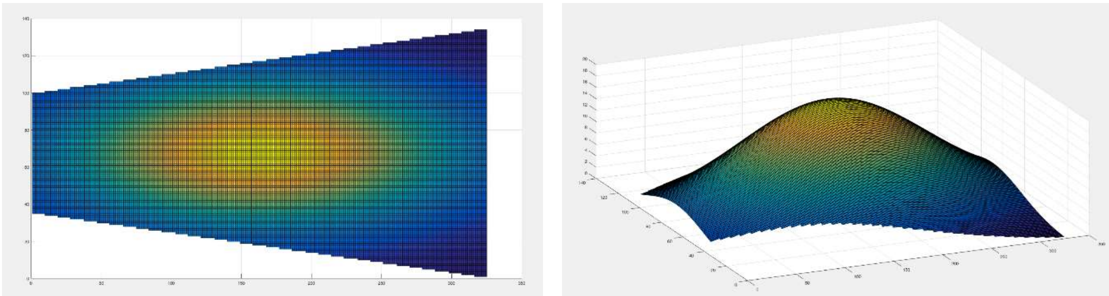

FIGURE 7: PRESSURE DISTRIBUTION OVER THE TOOTH FLANK.

After the simulation run, we expect to analyse the results through visualization of the following features:

- Equivalent stresses according to e.g. Von Mises method and/or others (Figure 8)
- Strain (Figure 9)
- Frictional stresses (Figure 10) and contact pressures
- Displacements (absolute and in one specific direction)
- Magnification of displacements

For the best scope of the results analysis, it is important to include pan/zoom/rotate features to the visualization tool. Also the possibility to have sections in the solid is very useful (Figure 11). As far as the contact pressure is concerned, this can be computed as a result in the next phases of CAxMan, when the problem of contacts is addressed.

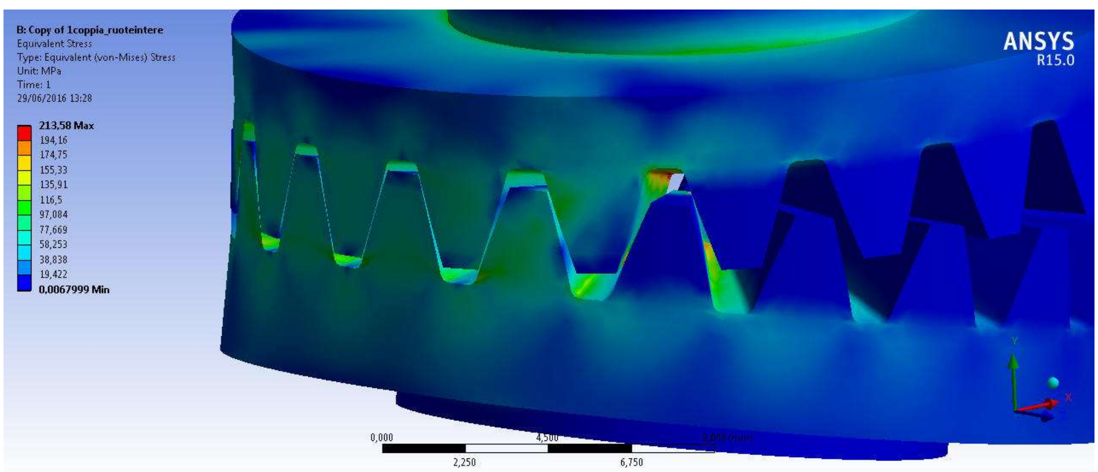

FIGURE 8: EXAMPLE OF RESULTS VISUALIZATION, VON MISES EQUIVALENT STRESS.

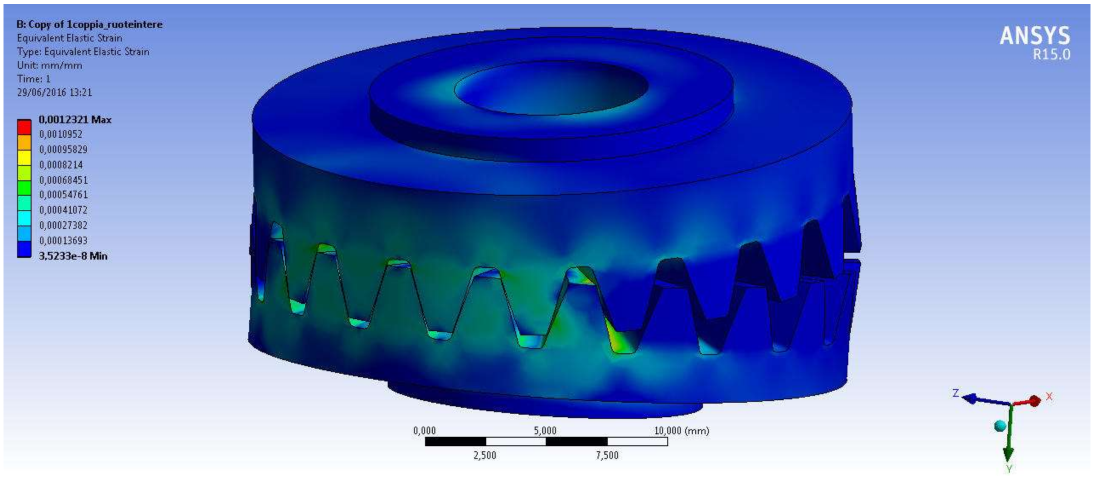

FIGURE 9: EXAMPLE OF RESULTS VISUALIZATION, ELASTIC STRAIN.

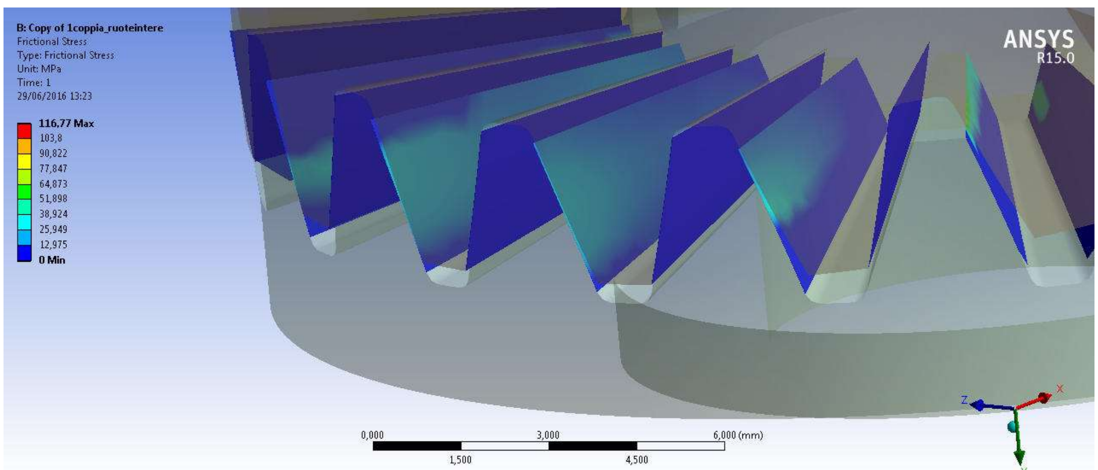

FIGURE 10: EXAMPLE OF RESULTS VISUALIZATION, FRICTIONAL STRESS.

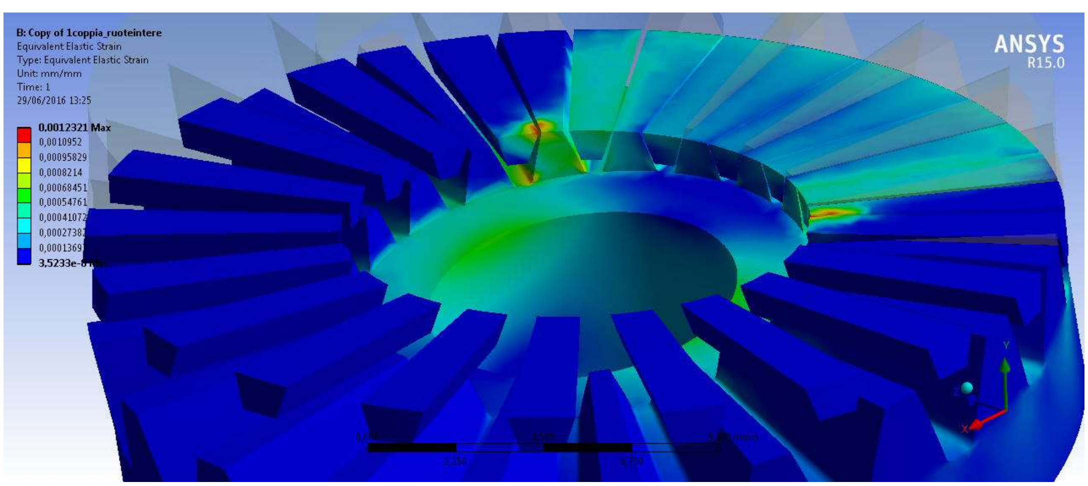

FIGURE 11: EXAMPLE OF RESULTS VISUALIZATION, EQUIVALENT STRAIN IN A CROSS SECTIONED MODEL.

The following step of the IGA analyses will be the contact phenomena modelling. However, the visualization of results after the simulation run will be similar to the list reported above.

An interesting optimisation that STAM does not perform to date is that of the tooth shape. There are several methods to correct the tooth micro-geometry, such as tip relief, end relief, lead crowning and profile crowning. In addition, more sophisticated optimisation approaches exist, that allow for:

- Distributing the contact pressure over the face of the tooth, avoiding edge contact;
- Minimising the transmission error;
- Reducing shocks after motion stops and reverse motion;
- Reducing bending stress at the tooth root.

The micro-geometry (ease-off topography) is generally represented as a polynomial surface (e.g. up to the fourth degree): the polynomial coefficients are the design variables to be determined. The objective functions are: average efficiency loss, maximum contact pressure and transmission error under load.

In order for STAM to perform this kind of optimisation, the results of a simulation will have to make available punctual data about the contact pressure, stress and strain in the points of the tooth flank. These results are needed in addition to the graphic representations discussed above. This information will be then used in the optimisation loop to achieve the goals listed above.

4.2 VISUALIZATION OF MOULD SIMULATIONS RESULTS

The mould has to support high pressures during the injection phase. These pressures can be really huge. It is quite important for the mould to be well designed to cancel the defaults we can get on the plastic part due to the deformation of the core or the cavity, see Figure 12.

FIGURE 12: DEFECTED PART DUE TO DEFORMATIONS OF THE MOULD.

To get a good stress analysis and to see the deformations, the main starting point is a flow analysis done on the part. With this analysis we get information about necessary pressure to produce the part by injection and also thermal analysis of the part after the injection cycle.

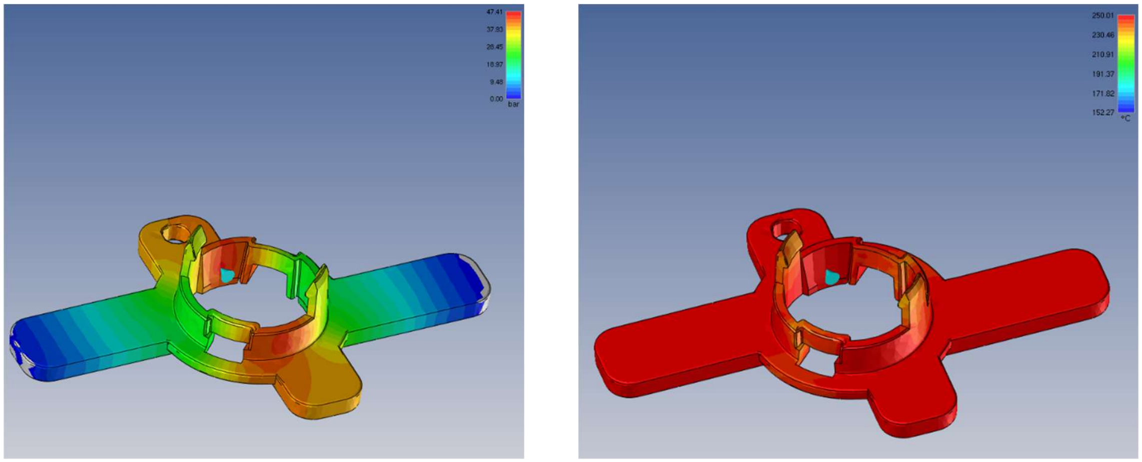

FIGURE 13: LEFT: INJECTION PRESSURE. RIGHT: THERMAL MAP AT THE END OF THE FILLING PROCESS.

From this starting point a mould designer needs to know exactly where and how the core and the cavity will be deformed, by performing a mechanical simulation. According to the simulation results, some modifications can be done first on the rheological side, changing the number and/or the position of the injection points, or by changing the dimensions on the blocks. From the point of view of the mechanical simulation, all the necessary features are already listed for the NUGEAR use case, and not detailed here. See Figure 13 to Figure 16 for some examples.

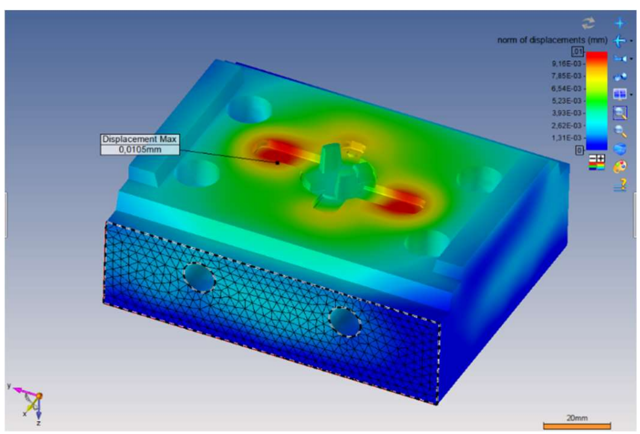

FIGURE 14: DEFORMATION OF THE CORE BLOCK.

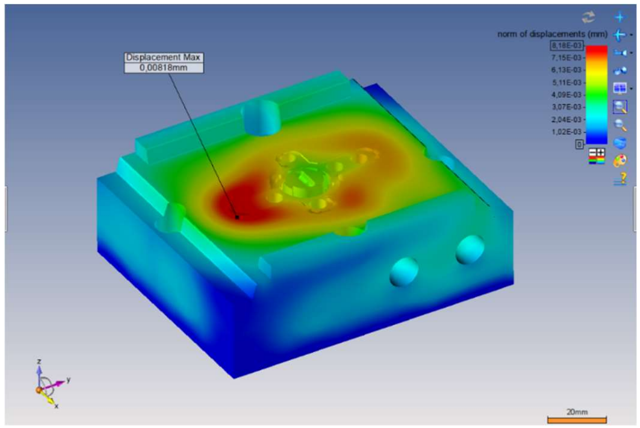

FIGURE 15: DEFORMATION OF THE CAVITY.

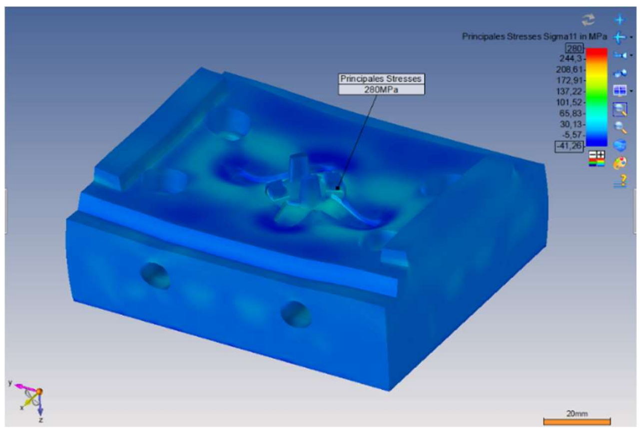

FIGURE 16: DEFORMATION WITH A SCALING FACTOR.

Besides the mechanical simulation, a full thermal analysis is necessary to see how the cooling system defined in each block works. This has influence for the plastic part and also for the injection cycle: a plastic part can be deformed and out of tolerance if a cooling system is badly defined. The analysis of the simulation results will be done through the visualization of the temperature field and its gradient, the computation of punctual values of the temperature, and the value of the heat flux through the surfaces of the cooling system. For the boundary conditions, the variation of temperature during the injection cycle, which ranges from 50 to 250 degrees, must be taken into account during the simulation. The range of temperature during a cycle will be obtained with a rheological software, such as Moldflow or Cadflow.

5 CURRENT STATUS

5.1 READING A GOTOOLS GEOMETRY IN IGATOOLS

Reading a GoTools single-block geometry in IGATools from a file is relatively simple: a g2 file is loaded and parsed using GoTools functionalities and a GoTools object is created. Then, an iterative loop browses through the GoTools object containers where the relevant geometrical information is stored (knots along each direction in the parameter space, their multiplicity, the control points, the degree of approximation in each direction) and these quantities are copied in the corresponding containers in IGATools.

For the multipatch case, instead each block is read from the g2 file in GoTools format and stored in an array. In addition to converting each of these blocks in the equivalent IGATools format as in the previous case, one also needs to store some additional technical information:

- The blocks connectivity/adjacency, i.e., through which faces the blocks are connected (e.g. block1-face2 and block5-face3 touching each other). Observe that the only way blocks are connected is through faces due to the "no T-junctions" assumption.
- Whether the two connecting faces have the same or reverse orientation (i.e., whether an identical movement in the parametric domains of the two faces results in the same movement in the physical domain on both faces or not).
- The degrees-of-freedom connectivity across connected faces (e.g. degree of freedom N on face A corresponds to degree of freedom M on face B). Such a matching is always available due to the assumption that the meshes on each block are conforming. Note also that the matching depends on the orientations of the two connecting faces.

The block connectivity is provided to IGATools from GoTools in a straightforward manner. The orientation check is done in IGATools, as well as the degrees-of-freedom connectivity. After importing and storing this technical information, no more interaction between GoTools and IGATools is needed to compute the numerical solution of the equation. The case of trimmed geometry is different and is discussed in the following.

The solution of the problem on general trimmed volumes, and in particular when the trimmed elements do not have six faces, is still under development, and it will require to extend the g2 file format of GoTools. Although the details of the method will be given in a revised version of deliverable D2.2, we explain here the main idea to understand the requirements for the interaction of the two software libraries.

First, it is necessary to know whether an element is completely inside the domain, completely outside, or trimmed. In the case it is trimmed, to perform numerical integration one must define

a suitable partition of the trimmed element into subelements[3]. This can be done by a reparametrization of the trimmed element in the parametric domain, for which it is necessary to map the trimming surface (actually, its intersection with the element) to the parametric domain of the trimmed volume, as we explain in the example below.

The reparametrization of a trimmed element requires access to Boolean operations in order to identify the part of the trimmed element where the parameter domain needs to be approximated by a non-trimmed spline volume. GoTools provides a rich set of intersection functionality and also some Boolean operations. Let us look at a simple example:

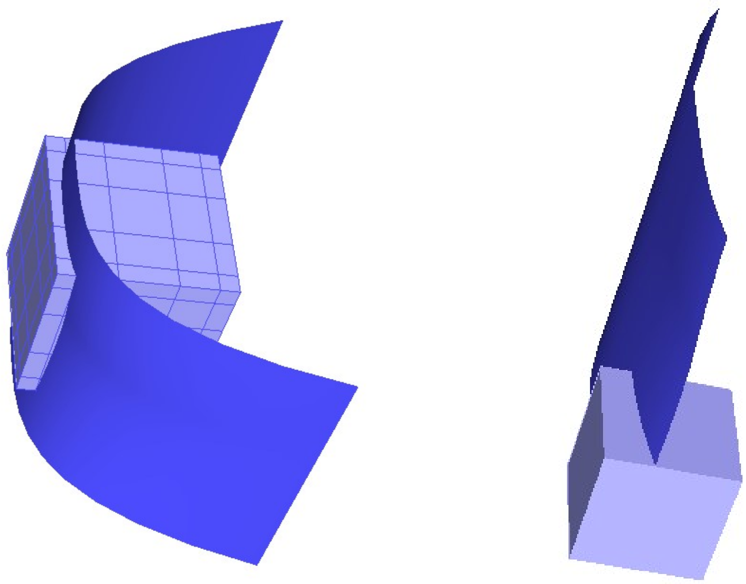

FIGURE 17: LEFT: TRIMMED SPLINES VOLUME. RIGHT: ZOOM ON AN ELEMENT.

In Figure 17 we have created a cubic spline volume consisting of 27 polynomial elements, 3 in each parameter direction. The visualized lines show the control mesh of the boundary surfaces. The volume has the same shape in geometry and parameter space. We intersect the initial volume with a spline surface to create a trimmed volume. A number of elements are intersected with this surface and we want to reparametrize these in the parameter domain. The picture on the right shows one of these elements together with the trimming surface.

The next step is to restrict the element to the valid part. This is performed by applying a Boolean operation that splits the element with the trimming surface. The two element pieces are represented as trimmed volumes. The piece lying inside the original trimmed volume, blue in Figure 18, is selected. Until now all operations have been performed in the geometry space.

[3] In the first version of D2.2 an adaptive quadrature technique, based on an octree, was defined. This is extremely time-consuming, because the elements of the octree are trimmed as well.

To approximate a trimmed volume by a spline volume, all boundary surfaces are extracted and approximated by spline surfaces. For simplicity, we assume that the volume is surrounded by 6 surfaces. Otherwise, either a degenerate volume must be constructed or further splitting must be applied. We note that this does not affect the stability of IGA, since the reparametrization is only used for numerical quadrature and not for defining the IGA basis functions, but in principle further splitting is preferred rather than degenerate volumes. In the situation where 6 spline surfaces define a closed shell, these surfaces can be interpolated to create a spline volume. We apply a generalization to the well-known Coons approach for surfaces (Pratt., 1985).

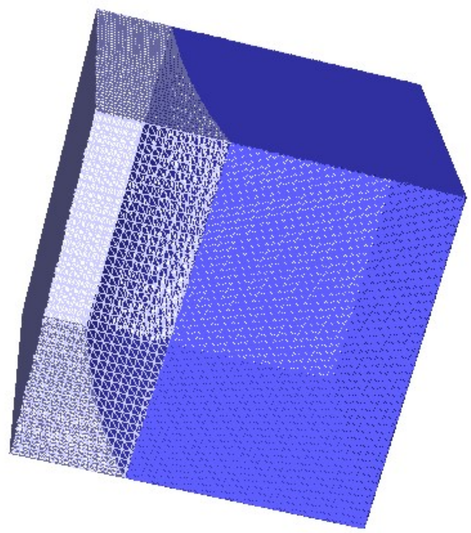

FIGURE 18: CHOOSING THE INTERNAL PART OF A TRIMMED VOLUME.

The volume construction is in principle the same regardless of whether we work in parameter or geometry space, as both spaces have dimension 3. The surface approximation algorithm is given an evaluator based surface and creates a spline surface. Thus, if the boundary surfaces of the trimmed element are evaluated in the parameter domain, an approximation of the trimmed element in the parameter domain will be created. Since this is an approximation, whenever the IGA space is refined a new reparametrization must be computed, in order to improve the accuracy. The boundary surfaces of the trimmed element, including the trimming surface, have knowledge about the volume they belong to. This enables point evaluation to be directed to the parameter space associated with the element. In Figure 19 we can see the parameter domain approximation of the elements intersected by the trimming surface.

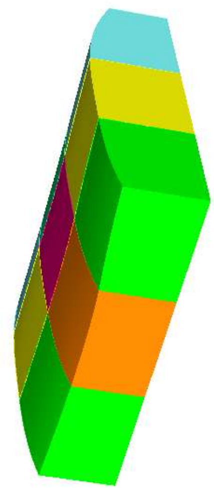

FIGURE 19: PARAMETER DOMAIN APPROXIMATION OF THE ELEMENTS INTERSECTED BY THE TRIMMING SURFACE.

5.2 PRELIMINARY RESULTS

We now show some preliminary results for the stationary heat equation and the linear elasticity equation. The analyses performed in this section must be understood as simple toy problems to show the capabilities of IGATools. More realistic tests will be performed in the future, also for the other mathematical models, once the TopSolid-GoTools-IGATools interaction is more evolved. The results are obtained over half of the bevel gear provided by STAM, with a geometry that consists of 42 blocks, see Figure 20. At this stage the visualization is done with ParaView[4], the use of the other software described in Section 1 will be decided later in the project, according to the needs of the consortium partners.

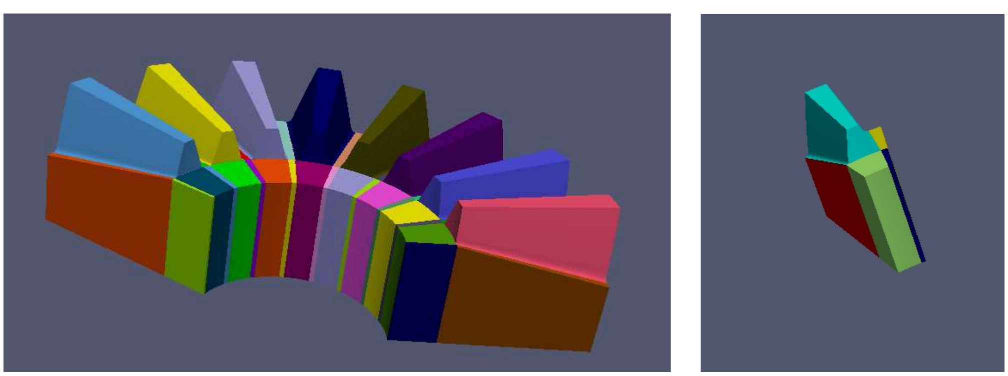

FIGURE 20: LEFT: MULTIBLOCK DECOMPOSITION OF STAM BEVEL GEAR.
RIGHT: DETAIL OF ONE TOOTH: 5 BLOCKS ARE NEEDED.

[4] http://www.paraview.org

5.2.1 Stationary heat equation

As a first example we solve the stationary heat equation (see Section 2.1). For this simple test, the heat conductivity is assumed to be equal to 1. In this setting, we perform two different analyses:

a) A test with a homogeneous Dirichlet boundary condition all over the surface (i.e., the temperature on the surface of the gear is assumed to be 0 ˚C) and an internal heat source (i.e., a volumetric heat flux) equal to 1 W/m^3.
b) A test where the temperature of the internal face of the gear is fixed at 10 ˚C and the rest of the gear exchanges heat with the external medium (e.g. air or water) at a fixed rate of -1 W/m^2.

The results of test a) are shown in Figure 21. In particular, we show the value of the temperature inside the gear, removing the inner portion of the ring without the teeth and in a cross section of one tooth. As expected, the inner core is hotter and the temperature decreases toward the surface. Note that the colour of the outer surface is constant blue everywhere, i.e., the Dirichlet boundary condition (prescribed temperature) is correctly imposed.

The results of test b) are shown in Figure 22. This time, we show the entire gear and a cross section. The result correctly predicts that the temperature steadily decreases even on the surface (due to the Neumann condition which prescribes heat exchange between the object and the surrounding environment) as we get further from the face kept at fixed temperature (in red in the figure).

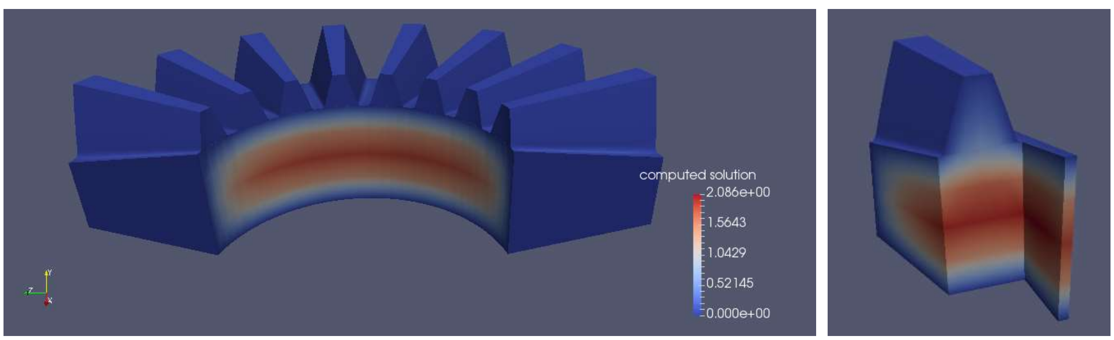

FIGURE 21: SOLUTION OF STATIONARY HEAT EQUATION - TEST A). LEFT: FULL SOLUTION; RIGHT: CROSS SECTION OF ONE TOOTH.

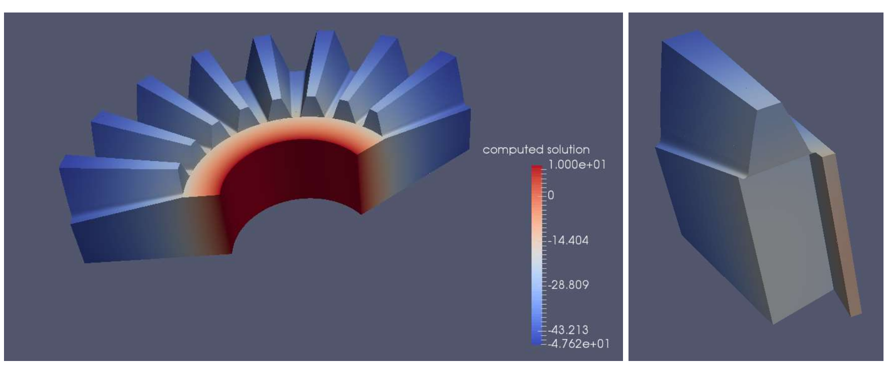

FIGURE 22: SOLUTION OF STATIONARY HEAT EQUATION - TEST B). LEFT: FULL SOLUTION; RIGHT: CROSS SECTION OF ONE TOOTH.

5.2.2 Linear elasticity equation

Next, we turn our attention to the elasticity equation described in Section 2.3. For this test, we slightly simplify the setting described in Section 4.1 and apply a constant pressure on the flank of a tooth (instead of the bell-shaped profile), yet with the same total resulting force. This condition is formally a non-homogeneous Neumann condition. The bottom face of the gear is kept fixed (i.e., a homogeneous Dirichlet condition), while the rest of the surface is subjected to a stress-free boundary condition (i.e., a homogeneous Neumann boundary condition). The gear is assumed to be made of Titanium 64 alloy, one of the materials supported by the 3D printer used under the premises of CAxMan.

Figure 23 shows the deformation of the gear under the above-described loading scenario, while Figure 24 shows the Von Mises equivalent stresses.

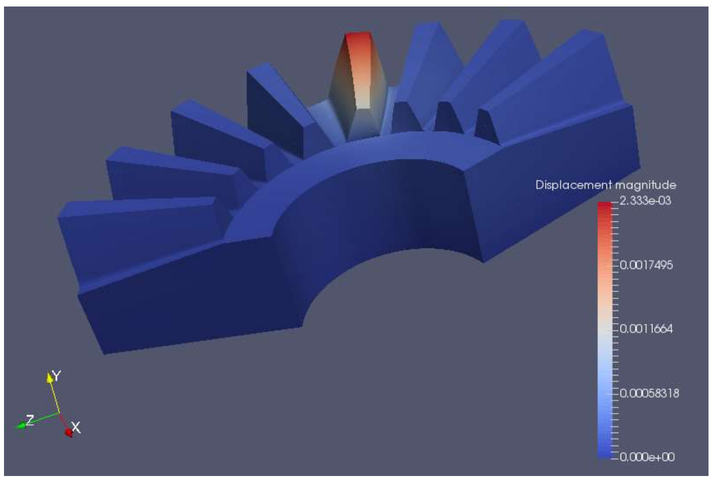

FIGURE 23: LINEAR ELASTICITY TEST, MAGNITUDE OF DISPLACEMENT. FOR VISUALIZATION PURPOSES, THE DISPLACEMENT HAS BEEN MAGNIFIED BY A FACTOR 400.

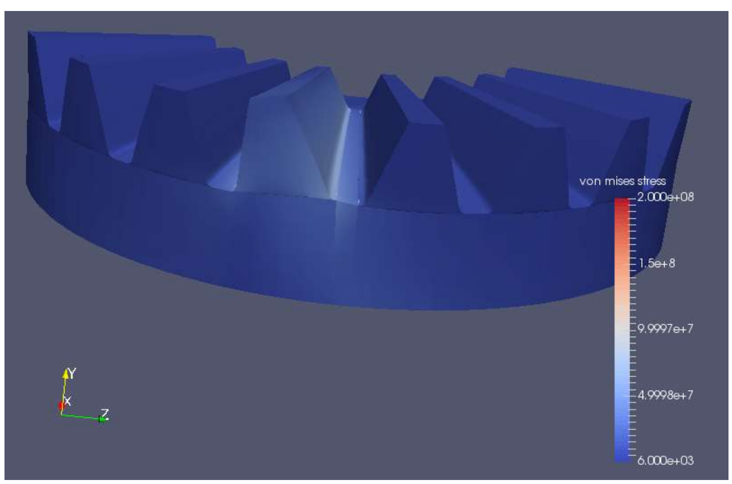

FIGURE 24 LINEAR ELASTICITY TEST, VON MISES STRESS.

6 CONCLUSIONS

We have presented in detail all the requirements and software interactions for the analysis tools in the linear setting, for thermal, mechanical and fluid flow problems. We have also included some preliminary results for the stationary thermal and linear elasticity problems. The task will continue with the implementation of the fluid flow problem, and a stable time discretization scheme, necessary for the simulation of the cooling channels in the mould use case.

Concerning the interaction of the three software libraries, the GoTools-IGATools integration is ready for multi-block geometries, while the reparametrization method for trimmed geometries is currently under development. The full interaction of these two libraries with TopSolid for the problem description and analysis, explained in Section 3, is still at an early stage, and will be developed during the next months. In fact, the interaction of the libraries goes beyond Task 2.4, and it will likely continue during the development of the analysis tools for the nonlinear setting.

7 REFERENCES

CAxMan Consortium. (2016). Deliverable 2.1. *Requirements: Design for additive manufacturing*.

CAxMan Consortium. (2016). Deliverable 2.2. *IGA based shape representation, initial version*.

Cottrell, J. A., Hughes, T. J., & Bazilevs, Y. (2009). *Isogeometric Analysis: toward integration of CAD and FEA.* John Wiley & sons.

Fuchs, F., & Hjelmervik, J. (2016). Interactive Isogeometric Volume Visualization with Pixel-Accurate Geometry. *IEEE Transactions on Visualization and Computer Graphics, 22*(2), 1102-1114.

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.

Pratt., I. F. (1985). *Computational Geometry for Design and Manufacture.* Chichester: Ellis Horwood Limited.