

Gerrit Wltte

# An isogeometric digital twin for Gunnerus

Master's thesis in Marine Technology  
Supervisor: Prof. Josef Kiendl  
June 2019



Gerrit Wltte

# An isogeometric digital twin for Gunnerus

Master's thesis in Marine Technology  
Supervisor: Prof. Josef Kiendl  
June 2019

Norwegian University of Science and Technology  
Faculty of Engineering  
Department of Marine Technology

 **NTNU**  
Norwegian University of  
Science and Technology





## Abstract

Isogeometric analysis (IGA) is a new approach in the world of Finite Element Analysis (FEA). The idea is to merge the Computer Aided Design (CAD) model directly into FEA. This was not possible until now due to different basis functions in CAD model and the FEA model. After introducing IGA in 2005 by [9], a lot of research improved the approach, which is why the idea came up to combine IGA with a digital twin. A digital twin is a digital representation of a physical structure, which is updated all the time and fed with information from the physical twin. Updates can be structural changes, cracks in the structure or a swap of parts. In this field IGA will increase the efficiency of the workflow, since the updated digital twin can be directly used for structural analysis. Digital twin will be a key technology for the upcoming Industry 4.0 and therefore the importance and usage of digital twin will increase.

To analyse a big structure, such as a digital twin, the existing implementation needs some improvements in regards to efficiency.

After describing the mathematical background of NURBS and the state of the art of IGA, different approaches are discussed to improve the efficiency and extend the existing code. Knot removal, neglecting small trimmed areas and polynomial degree for integration are discussed. Especially neglecting of small trimmed areas shows good improvements regarding the condition of the stiffness matrix. A value of 1% for the ratio between trimmed area and element area seems to be a good value, which is proved by different geometries. Further the import of IGES files was updated to read in multiple patches and the plotting of trimmed surfaces in physical space to plot multiple patches trimmed patches.

The work with the digital twin of 'Gunnerus' shows some problems with the digital twin itself and with the existing IGA code. The digital twin consists of too many single patches to couple every patch by hand and the thickness of every plate is lost. Therefore only small areas of 'Gunnerus' are modelled. On the other hand is that the existing element cases for trimmed elements are not sufficient. Tests with 'Gunnerus' have shown that there are trimming lines on the boundary and cuts through two vortex points of one element. This case is not covered by the existing one. This is not within the scope of this thesis and needs further research.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Geometrical Description</b>	<b>3</b>
2.1	NURBS Curves and Surfaces . . . . .	3
2.1.1	Knot vector . . . . .	3
2.1.2	Basis function . . . . .	3
2.1.3	B-Spline curves . . . . .	5
2.1.4	B-Spline surfaces . . . . .	6
2.1.5	NURBS . . . . .	7
2.1.6	Refinement . . . . .	7
2.2	IGES code . . . . .	8
2.3	Trimming in CAD . . . . .	12
<b>3</b>	<b>Isogeometric Analysis</b>	<b>14</b>
3.1	NURBS-based Isogeometric analysis . . . . .	14
3.2	NURBS-based Kirchhoff-Love shell . . . . .	15
3.3	Patch coupling . . . . .	16
3.3.1	Penalty coupling of non-matching isogeometric Kirchhoff-Love shell patches . . . . .	17
3.3.2	Patch coupling for trimmed surfaces along a trimming line . . . . .	19
3.4	Trimming in IGA . . . . .	19
3.4.1	State of the art . . . . .	19
<b>4</b>	<b>Improvements of IGA and the internal research code</b>	<b>25</b>
4.1	Validation of the existing IGA trimming code . . . . .	25
4.2	Flying Nodes . . . . .	28
4.2.1	Benchmark test for flying nodes . . . . .	30
4.3	Knot removal . . . . .	32
4.3.1	Benchmark test for knot removal . . . . .	35
4.4	Degree for Integration . . . . .	37
4.5	Plotting of trimmed surface . . . . .	38
<b>5</b>	<b>Gunnerus</b>	<b>39</b>
5.1	Digital twin Gunnerus and structural model . . . . .	40
5.2	Problems with structures of Gunnerus . . . . .	41
5.3	Simulation with parts of Gunnerus . . . . .	44
<b>6</b>	<b>Conclusions and Outlook</b>	<b>48</b>
<b>A</b>	<b>Validation of merging the two codes</b>	<b>52</b>

---

<b>B</b>	<b>Benchmark test flying nodes</b>	<b>53</b>
<b>C</b>	<b>Benchmark test knot removal</b>	<b>55</b>
<b>D</b>	<b>IGES file</b>	<b>55</b>
<b>E</b>	<b>Convergence study for parts of Gunnerus</b>	<b>57</b>

## List of Figures

Figure 1	Cubic B-Spline curve with the clamped knot vector $\Xi = [0, 0, 0, 0, 0.2, 0.5, 0.8, 1, 1, 1, 1]$ and its basis functions . . . . .	4
Figure 2	B-Spline curve with $C^1$ and $C^2$ continuity . . . . .	5
Figure 3	B-Spline surface . . . . .	6
Figure 4	NURBS curve, same control points as in Figure 1a but with different weights . . . . .	8
Figure 5	two different orders of applying knot insertion and order elevation. Figure (a) shows the original configuration, which will be refined. On the left side first knot insertion and then order elevation has been done. On the right side first order elevation and then knot insertion took place. This is the new k- refinement [9] . . . . .	9
Figure 6	Trimming of a patch in Rhino . . . . .	13
Figure 7	vectors along the patch coupling line of patch $S^A$ [8] . . . . .	16
Figure 8	Patch coupling along a trimming line . . . . .	18
Figure 9	1 <sup>st</sup> searching step [19] . . . . .	20
Figure 10	2 <sup>nd</sup> searching step [19] . . . . .	21
Figure 11	the three cases of trimmed elements: three vertex points active (left), two vertex-points are active (middle), one vertex-points are active (right) [19] . . . . .	21
Figure 12	Rotation of quadrilateral trimmed elements . . . . .	23
Figure 13	Deck-1 wrong deformation . . . . .	25
Figure 14	deformation of element cases 7, 8, 9, 10 . . . . .	26
Figure 15	deformation on three different points on ‘Trapezium’ . . . . .	27
Figure 16	deformation of ‘plate with hole’ . . . . .	27
Figure 17	Flying nodes . . . . .	28
Figure 18	Sub elements of elements with one active vortex point . . . . .	29
Figure 19	geometries for benchmark test flying nodes . . . . .	30
Figure 20	Gauss points . . . . .	32
Figure 21	Knot removal for Deck-1 . . . . .	34
Figure 22	new Gauss points after knot removal, Tol 0,01 . . . . .	35
Figure 23	Points of measurements . . . . .	36
Figure 24	results of benchmark test for knot removal . . . . .	36
Figure 25	changes on trimming curve due to knot removal . . . . .	37
Figure 26	‘Gunnerus’ [21] . . . . .	39
Figure 27	Gunnerus as a digital twin . . . . .	40
Figure 28	Gunnerus as a CAD drawing with all structural elements . . . . .	41
Figure 29	frame 50 of Gunnerus in parametric space . . . . .	42
Figure 30	Gauss integration points for frame 50 in original shape . . . . .	43
Figure 31	remodelled frame 50 . . . . .	44

---

Figure 32	Deck-1 . . . . .	45
Figure 33	second geometry of Gunnerus . . . . .	45
Figure 34	results of convergence study . . . . .	46
Figure 35	Deck roof . . . . .	47

---

**List of Tables**

Table 1	format of the Directory Entry section [1] . . . . .	10
Table 2	parameter data for Type 126 [1] . . . . .	12
Table 3	comparison of FEA and IGA based on NURBS [9] . . . . .	14
Table 4	input parameters for validation analysis . . . . .	27
Table 5	results of the convergence study . . . . .	31
Table 6	input parameter for benchmark test (flying nodes) . . . . .	31
Table 7	input parameter for benchmark test of knot removal on Deck-1 . . . . .	35
Table 8	main dimensions of Gunnerus . . . . .	39
Table 9	input parameters . . . . .	46
Table 10	trapezium . . . . .	52
Table 11	plate with hole (left,down) . . . . .	52
Table 12	results of benchmark test $eps = 0,01$ . . . . .	53
Table 13	results of benchmark test $eps = 0,03$ . . . . .	54
Table 14	results of benchmark test knot removal . . . . .	55
Table 15	results of convergence test . . . . .	57

---

## Nomenclature

$\alpha_d$	penalty parameter for displacement
$\alpha_r$	penalty parameter for rotation
$\delta W^{pd}$	penalty virtual work for displacement
$\delta W^{pr}$	penalty virtual work for rotation
$\delta W_{int}$	internal virtual work
$\eta$	knot
$\eta_{\alpha\beta}$	membrane strain
$\mathcal{L}$	interface curve
$\nu$	Poisson's ratio
$\theta$	thickness coordinate
$\mathbf{a}_3$	normal vector
$\mathbf{a}_{\alpha\beta}$	metric coefficients
$\mathbf{a}_\alpha$	base vectors
$\mathbf{a}_i$	base vectors on the middle surface
$A_{min}$	smallest area ration between trimmed area and element area
$\mathbf{a}_n$	in-plane normal vector
$\mathbf{a}_t$	tangent vector
$b_{\alpha\beta}$	curvature coefficients
$C$	parametric curve
CAD	Computer Aided Design
$D$	material matrix
$d$	distance vector
deg	polynomial degree of surface
$E$	Young's modulus
$E_{\alpha\beta}$	strain coefficients
FEA	Finite Element Analysis

---

FEM	Finite Element Method
GP	number of Gauss points
H	knot vector
i.e.	that is to say
M	basis function
m	stresses for bending moments
N	basis function
n	stress resultans for normal forces
NURBS	Non-Uniform Rational B-Splines
P	control point
p	polynomial degree
Pl	point where the load is acting on (parametric space)
q	polynomial degree
$r_{in}$	inner radius
$r_{out}$	outer radius
t	thickness of the shell
v	tangential vector of the trimming curve



## 1 Introduction

In the marine industry exist many geometries which are in one dimension much smaller than in the other two dimensions, such as the ship hull or plates for hatches. To account for its curved nature, the ship hull can be modelled as a shell structure. Shell structures carry their load ‘through its shape’ and the curved surface results in an interaction between membrane and bending moments. Similar to the beam theory, two different shell theories exist. One is called Reissner-Mindlin theory or thick shell theory. It can be compared to the Timoshenko-beam theory and takes into account transverse shear deformations [17]. The Reissner-Mindlin theory is valid for shells, where the plate thickness  $p$  is divided by length  $l$  to the power of three or less than one. If  $\frac{p}{l} \ll 1$  the thin plate theory can be used. This theory is also called Kirchhoff-Love theory and is based on the assumption of the ‘classical plate theory’ of Gustav R. Kirchhoff. The Reissner-Mindlin theory is used although most of the shells are thin shells, in the finite element method. Furthermore the Kirchhoff-Love theory could be applied.

The finite element method is the standard in Computer Aided Engineering CAE and is widely used in structural analysis. The geometric model is described by many small elements. These elements are defined by nodal points, which are connected by basis function, normally linear polynomials. In Computer Aided Design CAD it is treated differently. Here the geometric model is described by non-uniform rational B-Splines NURBS. They are based also on nodal points, but the basis function are polynomials of higher order, which is why they can represent the geometry in a more detailed way. The geometric model is the foundation for the analysis model and causes problems due to the different basis functions. The transition from the CAD model to the CAE model is called meshing and it is presented as the most time-consuming step and is presented as the bottleneck of the finite element method. Lovadina et al. [15] states that the mesh generation accounts for more than 80% of overall analysis time. For simple geometries, automatic meshing works fine but for more complex geometries, such as ship hulls, wind blades etc. it can be done only semi-automatically and is rather time consuming. This is the reason why there are new approaches to avoid the meshing, such as Isogeometric analysis IGA.

Hughes et al. [9] presented the idea of IGA, which uses the same geometric model, based on NURBS for the analytical model. The result is that the analysis can work with the exact geometrical model and can be applied to all different fields of engineering. First tests of IGA show great advantages in comparison to the traditional FEM, such as high accuracy and robustness. This is a result of higher polynomial degree, which provides higher continuity.

During the design of complex structures, even for smaller geometries a common tool is the trimming. The trimming tool cuts off surfaces or curves in the model. This is a quiet powerful tool because adjustments can be done easily. In addition the trimming tool does not change any description of the surface or curve. It makes the part invisible, which is trimmed. Whereas the knot vectors, control points and basis functions are still

the same. This is a big challenge for IGA, because it is only working with the description of control points, knot vectors and basis functions. In order to solve this problem two different solutions exist. First, avoiding trimming during the design phase, but this would be too time consuming and the main advantage of IGA would be neglected, the other option is that IGA figures out the trimmed patches and computes them in a special way. S. Schönfeldt [19] wrote an implementation to the existing research code of NTNU, which can import IGES, categorizes the element into the possible trimming cases and calculates the integration points for the analysis.

A novel idea is to combine IGA with a digital twin. The digital twin is an ‘integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin’ [20]. This will be one of the key concepts of the Industry 4.0, which is coming closer and closer. In that way information can be exchanged between the physical system and the digital twin and vice versa. This can help to understand how the physical system works and how to improve the product lifecycle of that system. Here the bottleneck of FEM results in spending a lot of time in meshing. For every change at the digital twin a new mesh has to be generated, due to updates on the structure, etc. With IGA the efficiency of this workflow will significantly increase. In this thesis ‘Gunnerus’ the research vessel of NTNU will be used as a digital twin. It was built during a summer job by five students of NTNU. The digital twin contains a structural model, which is used to perform IGA on a complex structure with trimmed surfaces.

The thesis is outlined as follows:

- *Chapter 2*: The geometrical basics, which is needed for this thesis, with respect to geometric modelling of NURBS in 1D and 2D is represented, as well the refinement of NURBS. Further a small outline of IGES and trimming in CAD is presented to understand the workflow between CAD and IGA in a better way.
- *Chapter 3*: This chapter provides an overview of IGA. It explains the basics of NURBS-based IGA and the actual development of the in-house research code with special focus of the trimming option.
- *Chapter 4*: The revision of the implementation for analysis on trimmed surfaces is outlined, with some benchmark test. The aim is to improve the code in regards of efficiency.
- *Chapter 5*: In this chapter, IGA is performed on the structural model of Gunnerus. Also, the upcoming challenges are presented and discussed.
- *Chapter 6*: A conclusion and outlook on the basis of the thesis is presented.

## 2 Geometrical Description

To describe geometric shapes mathematically, it is important to use functions which are extremely flexible and have independent parameters to represent a wide range of geometric shapes. Parametric functions offer almost all these requirements. With parametric representation, it is possible to describe even complex curves as an archimedean spiral. This is why it is the most common way of representing curves and surfaces in CAD. All the following methods are based on parametric representation.

### 2.1 NURBS Curves and Surfaces

NURBS curves and surfaces are based on Béziers and B-Splines [18]. Only B-Splines are described shortly in the next part. For a more detailed description of Béziers and B-Splines the reader is referred to [18].

#### 2.1.1 Knot vector

The knot vector  $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$  defines the parametric space.  $\xi_i$  are the parametric coordinates or knots and divide the B-Spline curve in different sections. The parametric coordinates  $\xi_i$  have to be ordered in the way of  $\xi_i \leq \xi_{i+1}$ , that is the only requirement for knot vectors [18]. Otherwise, the knot vector is divided into two types: periodic and open. If the knot vector has a multiplicity  $p + 1$  at the first and last knot it is called open knot vector, otherwise it is a periodic knot vector. This gives the open knot vector the possibility to describe the exact start and end point of a curve, which is why the open knot vector is state of the art in CAD[16].

The two types of knot vectors can be specified more precise with the two classes non-uniform and uniform. If all knots have the same distance to each other, they are called uniform otherwise non-uniform.

#### 2.1.2 Basis function

The B-Spline basis functions are based on the knot vector and the polynomial degree and are computed by the Cox-deBoor recursion formula [4]. The start of computing the first basis function is with  $p = 0$  and is defined as:

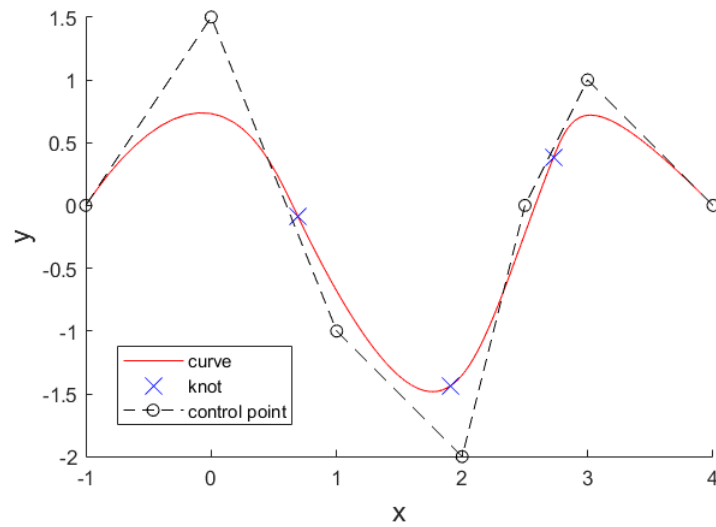
$$N_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1} \\ 0 & \textit{otherwise} \end{cases} \quad (1)$$

for  $p \geq 1$  it is

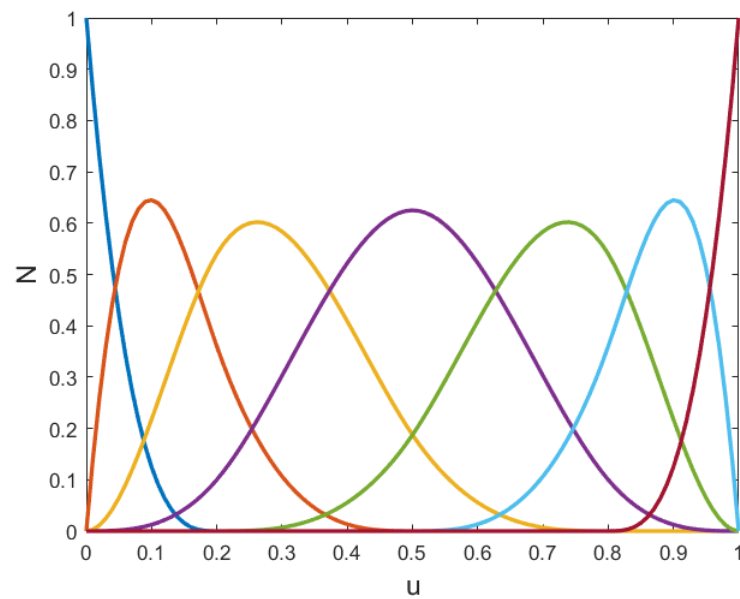
$$N_{i,p}(\xi) = \frac{(\xi - \xi_i)N_{i,p-1}(\xi)}{\xi_{i+p} - \xi_i} + \frac{(\xi_{i+p+1} - \xi)N_{i+1,p-1}(\xi)}{\xi_{i+p+1} - \xi_{i+1}} \quad (2)$$

From this equation some important properties of B-Splines can be deduced [9, 10]:

- local support, i.e. a basis function  $N_{i,p}(\xi)$  is non-zero only in the interval  $[\xi_i, \xi_{i+p+1}]$



(a) B-Spline curve with its control point (dashed line) and knot position on the curve



(b) Corresponding basis functions defined in parameter space

Figure 1: Cubic B-Spline curve with the clamped knot vector  $\Xi = [0, 0, 0, 0, 0.2, 0.5, 0.8, 1, 1, 1, 1]$  and its basis functions

- partition of unity, for  $\forall \xi \sum_{i=1}^n N_{i,p}(\xi) = 1$
- each basis function is non-negative, i.e.  $N_{i,p}(\xi) \geq 0$
- linear independence, i.e.  $\sum_{i=1}^n \alpha_i N_{i,p}(\xi) = 0 \iff \alpha_i = 0, i = 1, 2, \dots, n$

The knot vector and the resulting basis functions for a random B-Spline curve is shown in Figure 1.

### 2.1.3 B-Spline curves

B-Splines are non-interpolating, piecewise defined polynomial curves. They are defined by the following entities:

- set of control points,  $P_i, i = 1, \dots, n$
- set of basis functions  $N_i, i = 1, \dots, n$

Yields to

$$C(\xi) = \sum_{i=1}^n N_{i,p}(\xi) P_i \quad (3)$$

as a formula for a B-Spline curve.

An important aspect of the B-Spline curves is the continuity,  $C$ . It describes the ‘smoothness’ of the curves and it exists three main different discontinuities. If the inner knot has a multiplicity of  $k = p$ , the continuity will be  $C^0$ . This will lead to a change in position. The other two discontinuities are:

- $C^1$  change in tangent, i.e. a kink in the curve
- $C^2$  change in acceleration

Figure 2 shows a B-Spline curve with different continuities at different knots.

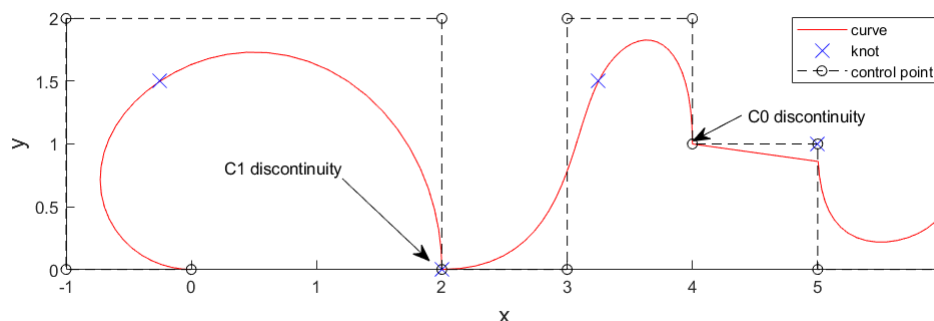


Figure 2: B-Spline curve with  $C^1$  and  $C^2$  continuity

Hughes et al. [9] summarizes the important properties of B-Spline curves with the respect to IGA as follows:

- the curve has continuous derivatives of order  $p - 1$ , but not at repeated knots or control points
- the curve is  $C^\infty$  continuous between two knots and is  $C^{p-k}$  continuous at a knot of multiplicity of  $k$
- An affine transformation of a B-Spline curve is obtained by applying the transformation to the control points

#### 2.1.4 B-Spline surfaces

A B-Spline surface is calculated by the tensor product of B-Spline basis functions in two parametric dimensions  $\xi$  and  $\eta$ . It is expressed by Rogers [18] as:

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) M_{j,q}(\eta) P_{i,j} \quad (4)$$

where  $N_{i,p}(\xi)$  and  $M_{j,q}(\eta)$  are the basis functions corresponding to the control points in s-direction  $n$  and t-direction  $m$ , two knot vectors  $\Xi$  and  $H$  and the polynomial degrees  $p$  and  $q$ , which do not have to be equal.

The Figure 3 shows an example of a quadratic B-Spline surface, with its control net (dashed lines).

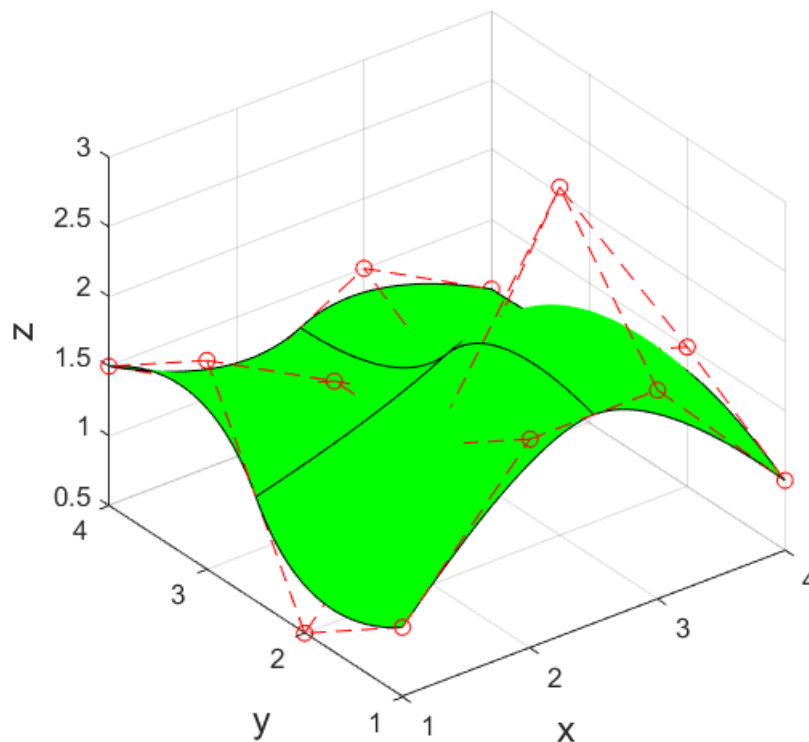


Figure 3: B-Spline surface

### 2.1.5 NURBS

‘NURBS’ stands for Non-Uniform Rational B-Splines. ‘Non-uniform’ means that the knot vector is not uniform in general. ‘Rational’ refers to the basis function. For NURBS the basis functions are piecewise rational polynomials. For B-Splines, the basis functions are piecewise polynomials.

The main difference of NURBS regarding B-Splines is that NURBS have an additional parameter. Each coordinate has an individual weight  $w_i$ . This yields that the control point  $P_i = (x_i, y_i, z_i, w_i)$  can be represented in a projective  $\mathbb{R}^4$  space. A NURBS curve is the projection of a B-Spline in  $\mathbb{R}^4$  with homogeneous control points onto  $\mathbb{R}^3$  [16]:

$$C(\xi) = \frac{\sum_{i=1}^n N_{i,p}(\xi)w_i P_i}{\sum_{k=1}^n N_{k,p}(\xi)w_k} \quad (5)$$

The NURBS basis function is:

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{i=1}^n N_{i,p}(\xi)w_i} \quad (6)$$

Yielding to the more common form:

$$C(\xi) = \sum_{i=1}^n R_{i,p}(\xi)P_i \quad (7)$$

Equation 5 shows that B-Splines are a special case of NURBS. NURBS with equal control weights are B-Splines. This is also the biggest advantage of NURBS. With the control weight, there are almost no limitations regarding modelling complex shapes, which is why NURBS are established as a standard in CAD modelling.

NURBS surfaces are defined similar to the B-Spline surfaces, as [16]:

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}^{p,q}(\xi, \eta)P_{i,j} \quad (8)$$

with

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}} \quad (9)$$

Equation 8 is not a tensor product of one-dimensional NURBS basis functions. The equation is based on the weighted ratio of tensor products of B-Spline basis functions.

### 2.1.6 Refinement

There are two different methods to refine NURBS curves and surfaces:

- knot insertion, corresponding to FEM h-refinement
- order elevation, corresponding to FEM p-refinement

Both methods do not change the shape of the NURBS curve and they both enter more control points to the curve as a refinement, which results in a more accurate analysis. The

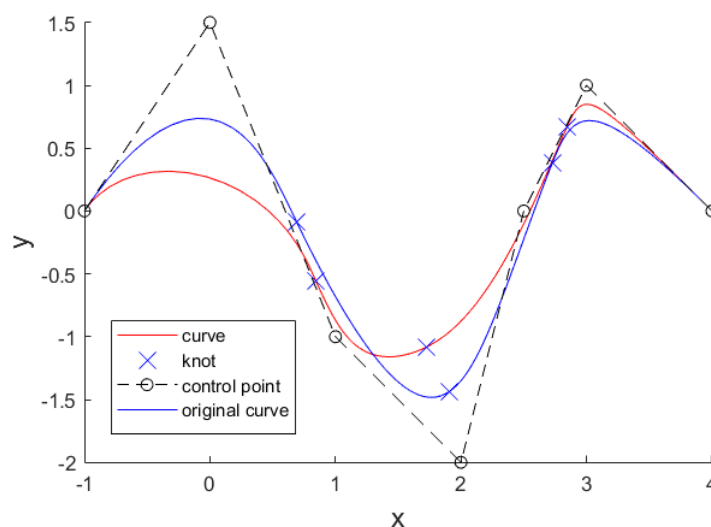


Figure 4: NURBS curve, same control points as in Figure 1a but with different weights

knot insertion [9] divides the curve into smaller knot spans by adding an extra knot. As a consequence, the continuity of this point will be reduced by one. The order elevation increases the polynomial degree of the basis function, while keeping the knot interval constant before and after the refinement.

Hughes et al. [9] introduces the k-refinement. The k-refinement combines the knot insertion and the order elevation by combining the two advantages of them. But it is important to mention that the two refinement methods are not interchangeable, as shown in Figure 5. In the k-refinement first the order elevation is done before the knot insertion. This is the common procedure in all existing IGA codes.

## 2.2 IGES code

As mentioned before, NURBS are the basics for CAD and CAD is common tool in the engineering world and a lot of different CAD software exist. Every software uses their one file format and the other software can not read them. Therefore a neutral file format was needed. Nowadays there exists two the IGES file format and the STEP file format.

In the internal research code is a script implemented, which can read IGES files and gives out the number of patches, the control points, knot vectors, polynomial degree of the surface and existing curves.

IGES stands for Initial Graphics Exchange Specification and is used to transfer geometry data and CAD models from one program to another one. It is a vendor-neutral file format, which was initiative of the United States Air Force Integrated Computer Aided Manufacturing program in 1979 [22].

The file is an ASCII text based format and every line has exactly 80 characters. Also the file is split into five sections [1]:

1. Start (Sender comments)



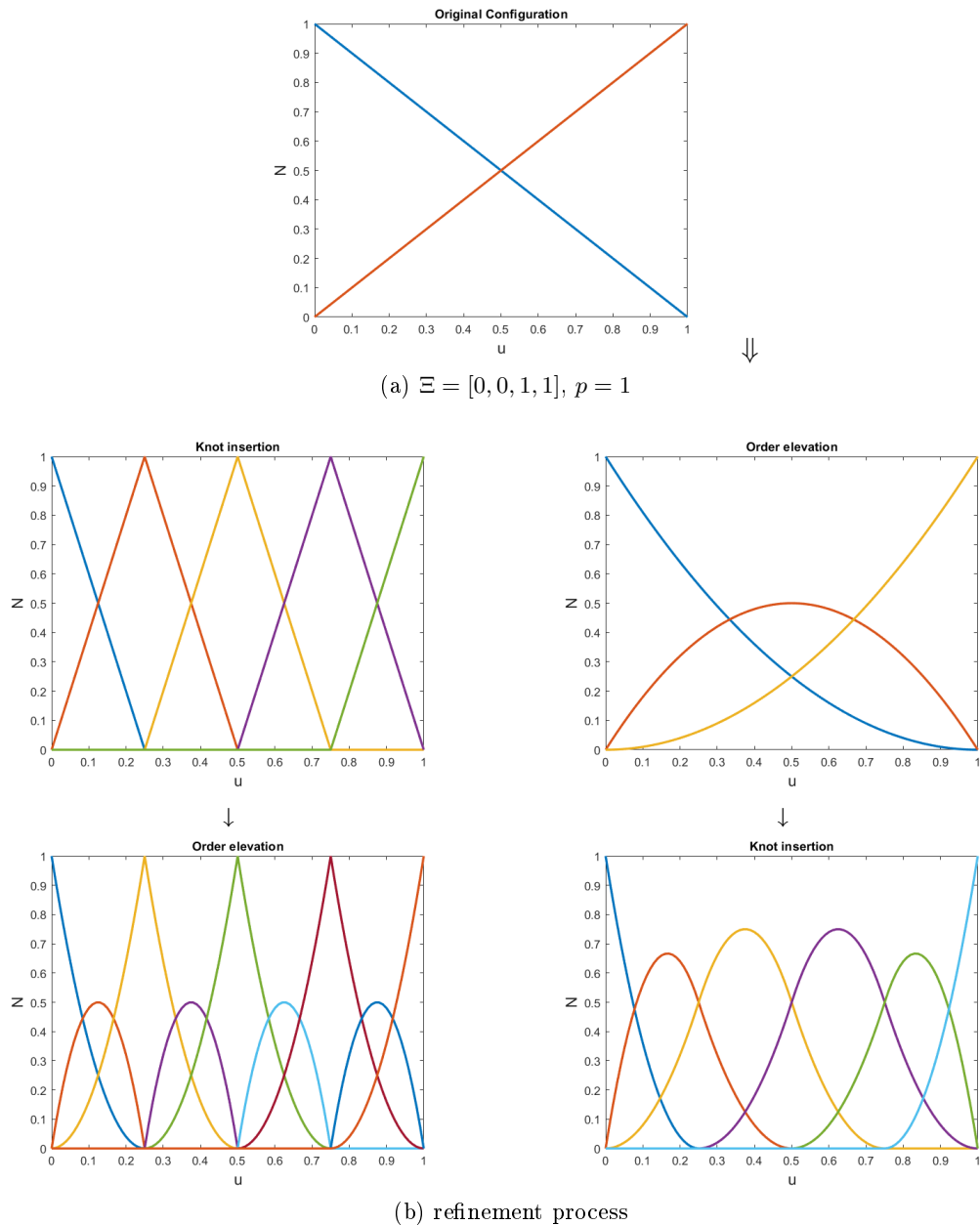


Figure 5: two different orders of applying knot insertion and order elevation. Figure (a) shows the original configuration, which will be refined. On the left side first knot insertion and then order elevation has been done. On the right side first order elevation and then knot insertion took place. This is the new  $k$ -refinement [9]

2. Global (general file characteristics)
3. Directory Entry (Entity index and common attributes)
4. Parameter Data (Entity Data)
5. Terminate (Control totals)

In the appendix D is a generic IGES file of a untrimmed trapezium. The important sections of the IGES file for IGA are the sections Directory Entry and Parameter data. Both of them contain entities, which can be geometry and non-geometry entities. The geometry entities define the physical shape of a product and include points, curves, surfaces, solids. The non-geometry entities specify annotation, definition, structure, colours etc. It exists around 150 different entities, but the important ones for IGA are: 128, 143 and 126.

- 128 describes Rational B-Spline surfaces
- 143 describes bounded surfaces
- 126 describes Rational B-Spline curves

An entity, for example 128, is described in the Directory Entry as follows:

1	128	4	0	0	1	0	0	000010000D
		7						
2	128	0	-1	4	8	0	0 TrimSrf	0D
		8						

Each line is separated into ten character fields, which contain eight characters and every entity gets two lines. That ends in 20 characters fields for every entity. These fields map to following properties:

Table 1: format of the Directory Entry section [1]

<i>col</i> 1-8	<i>col</i> 9-16	<i>col</i> 17-24	<i>col</i> 25-32	<i>col</i> 33-40	<i>col</i> 41-48	<i>col</i> 49-56	<i>col</i> 57-64	<i>col</i> 65-72	<i>col</i> 73-80
Entity type	parameter data	Structure	Line Font pattern	Level	view	Transformation Matrix	Label Display Assoc.	Status number	Sequence number
Entity type	line weight number	colour number	Parameter line count	Form number	Reserved	Reserved	Entity label	Entity subscript number	Sequence number

These fields define the following properties for every entity:

**Entity type** Identifies the entity type

**Parameter structure** Pointer to the first line of the parameter data for the entity

**Structure** zero or pointer to definition entity or zero

**Line font patter** number to line font pattern entity (Type 304) or zero

**Level** specifies levels to be associated with this entity or zero

**View** defines viewing options, View Entity (Type 410) or zero

**Transformation matrix** references a transformation entity (Type 124) or zero

**Label Display Associativity** pointer to the Directory Entry of a label Display Associativity (Type 402) or zero

**Status number** contains four sections of two numbers. 1-2: Blank status; 3-4: Subordinate Entity Switch; 5-6: Entity Use Flag; 7-8: Hierarchy

**Sequence number** physical count of this line from the beginning of the Directory section

**Line weight number** System display thickness

**Colour number** defines the colour of the entity

**Parameter line count** Quantity of lines in parameter section which are related to this entity

**Form number** describes the form or the representation of the entity or zero

**Entity Label** right justifies or NULL

**Entity subscript number** digit unsigned number related to the entity label

The Parameter Data Section comes after the Directory Entry and lists the data for each respective entry. For the entity type 128 the entry can look like this:

```

1 128,1,1,1,1,0,0,1,0,0,0.0D0,0.0D0,9000.0D0,9000.0D0,0.0D0,0.0D0,0000007P
   4 8000.0D0,8000.0D0,1.0D0,1.0D0,1.0D0,1.0D0,250.0D0,28.0D0,0.0D0,
   0000007P      5 9250.0D0,28.0D0,0.0D0,250.0D0,8028.0D0,0.0D0,9250.0D0
,8028.0D0,0000007P      6
2 0.0D0,0.0D0,9000.0D0,0.0D0,8000.0D0,0,1,5;                                0000007P
   7

```

The commas separate the parameters and the semi-colon ends the listings. The columns from 4-64 contain parameter data, 65-72 the data entry pointer number (needs to be odd) and 73 to 80 the Sequence number similar to the one from the Directory Entry. In the following the parameter data for a Rational B-Spline curve (Type 126) are explained, because that contains the information, which are needed for IGA.

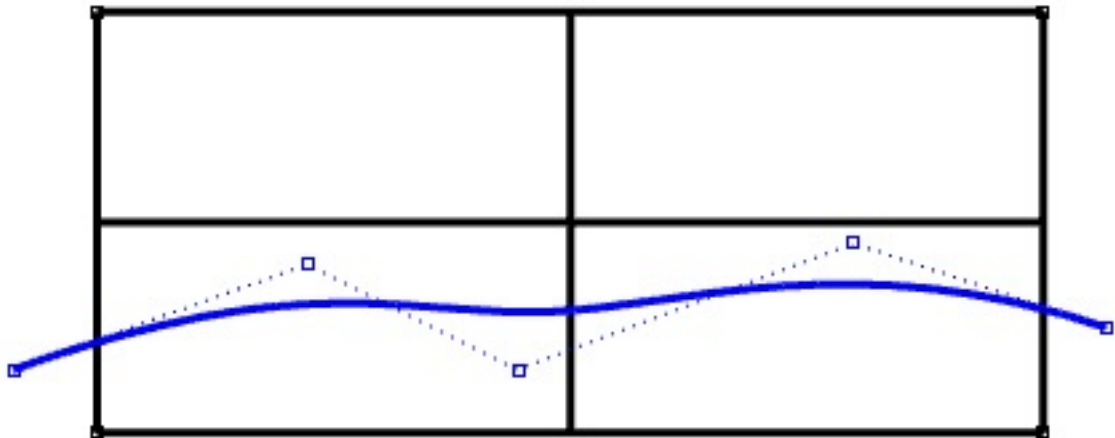
Table 2: parameter data for Type 126 [1]

<i>Index in list</i>	<i>Type of data</i>	<i>Name</i>	<i>Description</i>
1	Integer	K	Upper index of sum
2	Integer	M	Degree of basis function
3	Integer	Prop1	0=nonplanar, 1=planar
4	Integer	Prop2	0=open curve, 1=closed curve
5	Integer	Prop3	0=rational, 1=polynomial
6	Integer	Prop4	0=nonperiodic, 1=periodic
7	Real	T(1)	First value of knot sequence
8+K+M	Real	T(1+K+M)	Last value of knot sequence
9+K+M	Real	W(0)	First weight
9+2K+M	Real	W(K)	Last weight
10+2K+M	Real	X(0)	x-coordinate of first CP
11+5*K+M	Real	Z(K)	y-coordinate of last CP
12+5*K+M	Real	V(0)	Start parameter value
13+5*K+M	Real	V(1)	Last parameter value
14+5*K+M	Real	Xnorm	Unit normal x (if planar)
15+5*K+M	Real	Ynorm	Unit normal y (if planar)
16+5*K+M	Real	Znorm	Unit normal z (if planar)

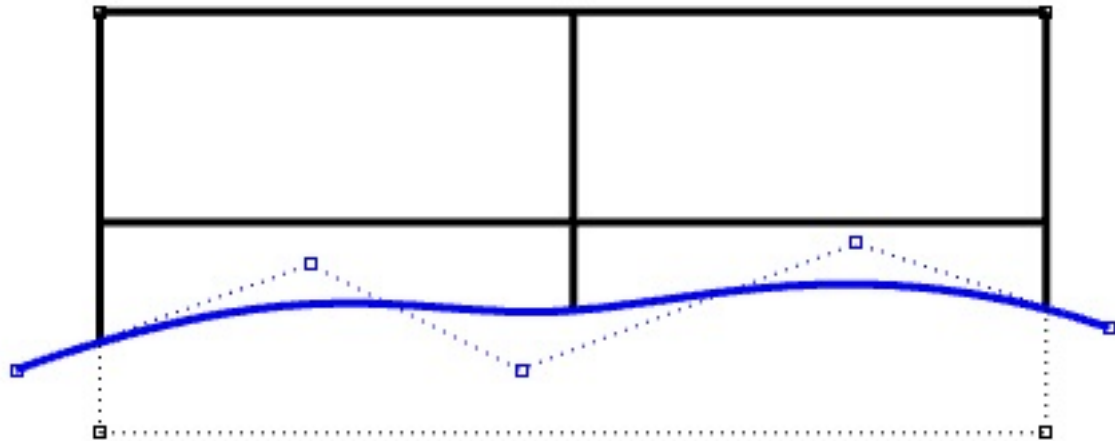
With these information all the necessary information can be selected from the IGES file for IGA. The research code will split the Directory Entry in the number of different patches. A trimmed patch will start with type 128 and the last entry for that patch is ending with type 143. In between of these lines the information of the curves type 126 are stored. A untrimmed patch will only contains type 128 without any bounded surfaces or curves depending on how it was modelled.

### 2.3 Trimming in CAD

A common tool in CAD is the trimming. Trimming makes parts of structures or patches invisible without changing the mathematical description. Figure 6 shows a untrimmed rectangular with a line (blue line) through the rectangular. The line will be the trimming line for the trimmed case. For both cases the mathematical description with the control points is shown (small squares in the corners). It can be seen that for both cases the mathematical description does not changed only the visible surface. This tool is really helpful for modelling complex structures because it is not always possible to model individual parts as precise as their are needed and then trimming is the solution.



(a) untrimmed



(b) trimmed

Figure 6: Trimming of a patch in Rhino

### 3 Isogeometric Analysis

Isogeometric Analysis is a new approach, which uses the same mathematical description for the analysis model and geometrical model. This means in practice that the geometrical model from a Computer Aided Design (CAD) program can be used directly for the analysis with Finite Element Analysis (FEA). The advantage of IGA will be that the whole process of meshing can be omitted. It is estimated that the whole meshing process takes 80% of the overall analysis time [15]. The reason why it is necessary to mesh from CAD to FEM is that these two approaches use different basis functions. CAD uses normally basis functions with higher polynomial degree. In comparison FEM is only using linear basis function, called Lagrange polynomials. This problem is solved with IGA, because it uses the same basis functions as CAD. IGA was introduced by Hughes et al. [9].

#### 3.1 NURBS-based Isogeometric analysis

In general, all functions, which are used in CAD can be used for IGA, as well. They have to fulfil the necessary conditions for basis functions, such as linear independence and partition of unity. The reason why NURBS are chosen is that they are the most common used in CAD and they fulfil the requirements [9]. Other alternatives to NURBS are subdivision surfaces and T-Splines [10].

There are some similarities between FEA and IGA, e.g. both analyses work with elements. The elements can be defined in two different ways in NURBS-based IGA. One approach is that the NURBS elements are defined by the knot spans of the knot vectors. In other words, one NURBS patch is subdivided into elements by the knot vector. The other approach is that one NURBS patch is one element. Each approach has its own advantages and disadvantages. In the end everyone has to decide for oneself which approach is preferred.

There are more similarities and differences between classical FEA and IGA, which Hughes et al.[9] summarise as shown in Table 3. For example FEA works with nodal points and IGA uses instead control points or the elements in FEA are defined by the mesh and in IGA the element boundaries are defined by the knots.

Table 3: comparison of FEA and IGA based on NURBS [9]

<i>FEA</i>	<i>IGA</i>	<i>Similarities</i>
Nodal points	Control points	Compact support
Nodal variables	Control variables	Partition of unity
Mesh	Knots	Isoparametric concept
Basis interpolates nodal points and variables	Basis does not interpolate nodal points and variables	Affine covariance
Approximate geometry	Exact geometry	Patch tests satisfied
Polynomial basis	NURBS basis	
Gibbs phenomena	Variation diminishing	
Subdomains	Patches	

### 3.2 NURBS-based Kirchhoff-Love shell

In his doctoral-thesis J. Kiendl [10] developed a shell element on the Kirchhoff-Love shell theory with NURBS as a basis function. The purpose was to use the Kirchhoff-Love shell elements for IGA. Kirchhoff-Love shells need a  $C^1$ -continuity for fulfilling the compatibility conditions. The reason for that is that the curvature is expressed with the second derivative of the geometry description. This is not possible with Lagrange polynomials as basis functions, without modification. Lagrange polynomials are common in FEM, which is why Reissner-Mindlin shell formulation is used more often. Reissner-Mindlin formulation takes into account the transverse shear strain, which ends up with variational index one and therefore only a  $C^0$ -continuity is required. With NURBS as basis functions, the Kirchhoff-Love shell formulation can be implemented without further modifications.

This thesis provides only a small outline of the NURBS-based Kirchhoff-Love shell formulation. For a more detailed version, the reader is referred to [10] and [11]. A general point is that the following formulas are valid for a displacement-based Kirchhoff-Love shell formulation and not only for the discretization with NURBS.

In Kirchhoff-Love shell theory, it is assumed that the cross sections, which are normal to the middle surface, remain normal after bending/deformation. This means that the transverse shear strains are neglected. For thin structures as for most ship hull constructions, it is a reasonable assumption. As a result of this assumption, each point of the shell can be described by its normal vector and the middle surface.

In the following derivation, for non-linear cases, the Einstein's summation convention is used and indices in Greek letters take the values  $\{1, 2\}$ , whereas indices in Latin letters take the values  $\{1, 2, 3\}$ .

The base vectors  $a_i$  on the middle surface are obtained by:

$$a_\alpha = x_{,\alpha} \quad (10)$$

$$a_3 = \frac{a_1 \times a_2}{|a_1 \times a_2|} \quad (11)$$

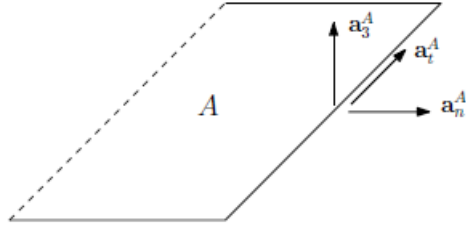
The metric and curvature coefficients of the middle surface are defined as:

$$a_{\alpha\beta} = a_\alpha \cdot a_\beta \quad (12)$$

$$b_{\alpha\beta} = -a_\alpha \cdot a_{3,\beta} = -a_\beta \cdot a_{3,\alpha} = a_{\alpha,\beta} \cdot a_3 \quad (13)$$

Corresponding to the first fundamental form of surfaces and second fundamental form of surfaces. The strain coefficients  $E_{\alpha\beta}$  for thin and moderate thick shells are defined by:

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \theta^3 \kappa_{\alpha\beta} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta}) + \theta^3(B_{\alpha\beta} - b_{\alpha\beta}) \quad (14)$$

Figure 7: vectors along the patch coupling line of patch  $S^A$  [8]

where  $\varepsilon_{\alpha\beta}$  is the membrane strain and  $\kappa_{\alpha\beta}$  represents the changes in curvature due to bending.

Corresponding to the strain coefficients, the stresses are split into two parts the stresses due to bending moments  $m$  and stress resultants  $n$  for normal forces. Because of symmetry, the tensors  $m$  and  $n$  only have three independent components  $[m^{11}, m^{22}, m^{12}]$  and  $[n^{11}, n^{22}, n^{12}]$  and can be computed with the material matrix  $D$  and Voigt notation:

$$\begin{bmatrix} \bar{n}^{11} \\ \bar{n}^{22} \\ \bar{n}^{12} \end{bmatrix} = t \cdot D \cdot \begin{bmatrix} \bar{\varepsilon}^{11} \\ \bar{\varepsilon}^{22} \\ 2\bar{\varepsilon}^{12} \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} \bar{m}^{11} \\ \bar{m}^{22} \\ \bar{m}^{12} \end{bmatrix} = \frac{t^3}{12} \cdot D \cdot \begin{bmatrix} \bar{\kappa}^{11} \\ \bar{\kappa}^{22} \\ 2\bar{\kappa}^{12} \end{bmatrix} \quad (16)$$

The bar over the coefficients indicates them as coefficient to a Cartesian basis. This transformation is necessary, because the material matrix  $D$  has physical material parameters referring to the local Cartesian coordinate system. With all these formulas, everything is set to calculate the internal virtual work:

$$\delta W_{int} = - \int_{\Omega} (S : \delta E) d\Omega = - \int_A (n : \delta \varepsilon + m : \delta \kappa) dA \quad (17)$$

Further the stiffness matrix can be derived from the second derivatives of the virtual work. For the exact formulations, the reader is referred to Kiendl [10].

Another advantage of the NURBS-based Kirchhoff-Love shell is that it needs neither rotational degrees of freedom nor nodal directors. Nevertheless, in some cases, such as coupling of NURBS patches or clamped boundary conditions, the rotations have to be described at the shell boundary.

### 3.3 Patch coupling

NURBS patches can represent a lot of different geometrical models, but sometimes it is better to use more than one NURBS patch to model complex structures, due to practical or technical limitations in geometry modelling [8]. If more patches are in use, they have to be coupled somehow. The challenge is to provide the  $C^1$  continuity from the Kirchhoff-



Love shell element, because the basis functions at the patch boundaries only provide  $C^0$  continuity, which is why the rotation-free IGA elements need a special treatment.

State of the art is the penalty method, because of its simplicity and flexibility. The main drawback of this method is the choice of the penalty parameter. If the value of the penalty parameter is too small, the result will not be precise enough and with a too high value the matrix may become ill-conditioned and the equation will have large numerical errors.

In others works, different approaches of penalty methods are shown for the use of patch coupling. All these approaches have some limitations or are not easy to implement for a general purpose analysis code. For example the approach of Breitenberger et al. [2] is limited for rotational deformation less than  $90^\circ$ .

### 3.3.1 Penalty coupling of non-matching isogeometric Kirchhoff-Love shell patches

Herrema et al. [8] developed a new penalty method sharing similarities with Breitenberger et al. [2] and Duong et al. [6], where only one penalty coefficient is needed for displacement and rotation. The new method is proofed for smooth and non-smooth, matching and non-matching patch interfaces.

In the following a small summary of the method is given. For a more detailed version, the interest reader is referred to the paper [8].

The coupling between two surfaces ( $S^A$  and  $S^B$ ) along a curve  $\mathcal{L}$  have to fulfil the displacement continuity and rotation continuity. The curve can be an interface curve or a trimmed curve. To fulfil the requirements, the coordinates of the integrations points along the line and the normal vector  $a_3$ , tangent vector  $a_t$  and in-plane normal vector  $a_n$  need to be known. Afterwards, the penalty virtual work for the displacement continuity :

$$\delta W^{pd} = \int_{\mathcal{L}} \alpha_d (u^A - u^B) * (\delta u^A - \delta u^B) d\mathcal{L} \quad (18)$$

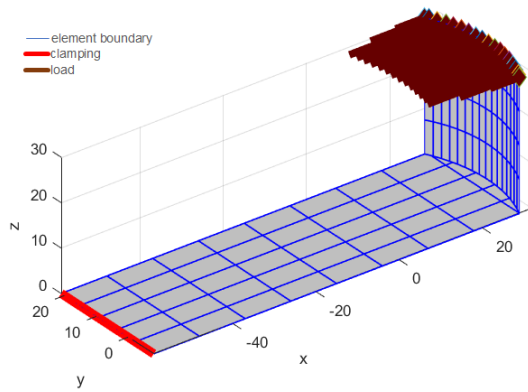
and for the rotation continuity

$$\delta W^{pr} = \int_{\mathcal{L}} \alpha_r ((a_3^A * a_3^B - \dot{a}_3^A * \dot{a}_3^B) (\delta a_3^A * \delta a_3^B - \delta \dot{a}_3^A * \delta \dot{a}_3^B)) \quad (19)$$

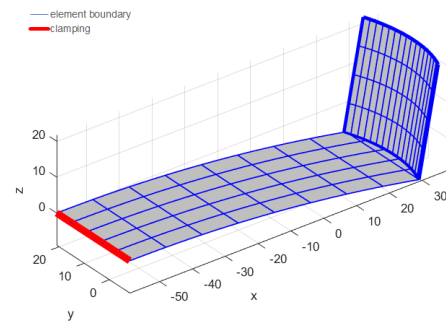
$$+ (a_n^A * a_3^B - \dot{a}_n^A * \dot{a}_3^B) (\delta a_n^A * \delta a_3^B - \delta \dot{a}_n^A * \delta \dot{a}_3^B) d\mathcal{L} \quad (20)$$

can be computed. The superscripts A and B indicate the two different patches. At the end the formulation of virtual work, Equation 21, has to be fulfilled at any arbitrary variation of the displacement variable  $\delta u_r$ .

$$\delta W = \delta W^{int} + \delta W^{pd} + \delta W^{pr} - \delta W^{ext} \quad (21)$$

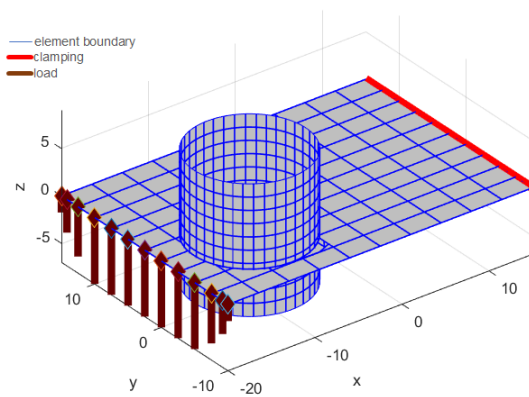


(b) undeformed, with boundary conditions and load

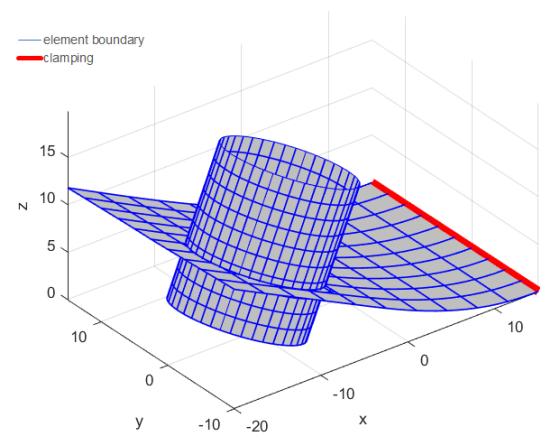


(c) deformed

(c) rectangular with an arc



(e) undeformed, with boundary conditions and load



(f) deformed

(f) plate with a cylinder

Figure 8: Patch coupling along a trimming line

### 3.3.2 Patch coupling for trimmed surfaces along a trimming line

The scope from the work of [8] was to find a penalty formulation for an interface curve. This formulation is extended for trimming curve. The in-house research code provides information about points on the trimming curve, the tangential vector at each point and mapping between the parameter and physical space. With these information, the coupling between trimmed patches along a trimming line is working.

In the following two models with two patches each are presented. The two patches are coupled along a trimming line.

The first model is a rectangular plate, which is trimmed by an arc (s. Figure 8a). The rectangular is clamped on the left side (red line) and the load act in positive x-direction on the upper end of the arc (arrows).

Figure 8d shows a plate with a cylinder. The cylinder is modelled through the plate. The plate gets trimmed by the cylinder and a hole is generated. The whole model consists of one untrimmed patch, the cylinder, and one trimmed patch, the plate with the hole. The two patches are coupled along the trimming line of the hole. The plate is clamped on the right side (red line) and the load is a line load in z-direction, acting on the left side of the plate. Figure 8 shows the coupled and deformed results for both models.

## 3.4 Trimming in IGA

During the set-up of IGA as a fully functional tool, some challenges came up. Some are solved and some still need some refinement. In the following the challenge ‘trimming’ will be presented, which is still under development for the IGA code of NTNU Marine Technology [19].

### 3.4.1 State of the art

Trimming is a widely used method in CAD. The visible part of geometries like surfaces or curves will be modified by cutting parts. The important point with respect to FEA or IGA is that the parameters of the curve or surface do not change in a mathematical way. In other words, IGA will use the original geometry instead of the trimmed surface for calculation of integration points. This ends up in the following problems which have to be solved for a sufficient use of IGA:

- to detect which element is trimmed
- mapping of trimmed elements

For FEA the problem with trimming is solved during the meshing part. It is one of the reasons why meshing takes approximately 80% of the analysis time.

S. Schönfeldt wrote her master thesis about the problems of detecting and mapping of trimmed elements and tried to solve them. In the following, a small outline of her work is presented. For more details, the reader is referred to her work [19].

The challenge is to figure out if an element is trimmed or not. For that two searching steps were implemented. Both searching steps are done in the parametric domain to have always rectangular elements. The first searching step compares the distance vector  $d$  with the inner radius  $r_{in}$  and outer radius  $r_{out}$ . The distance vector starts from the midpoint of the element to the closest point on the trimming curve (see Figure 9). For that the trimming curve of every element is divided into different parts with start- and endpoint. Afterwards the distance between every point and the midpoint is computed and the vector with the shortest distance is the distance vector. The inner and outer radius are defined as:

$$r_{in} = \min\left\{\frac{s_{i+1} - s_i}{2}, \frac{t_{j+1} - t_j}{2}\right\} \quad (22)$$

$$r_{out} = \frac{1}{2} * \sqrt{(s_{i+1} - s_i)^2 + (t_{j+1} - t_j)^2} \quad (23)$$

Then the elements can be classified as

- The element is untrimmed, if  $r_{out} < |d|$
- The element is trimmed, if  $r_{in} > |d|$
- more information is needed, if  $r_{in} \leq |d| < r_{out}$

If the element is untrimmed, it has to be checked whether it is in the trimmed domain or not. For that the tangential vector  $v$  of the trimming curve is needed. The direction of the trimming curve is always clockwise, whenever the inside region of the trimming curve is in void region. This means if the cross product of the distance vector is positive the element will be in the trimmed domain (active) and contribute to the stiffness matrix.

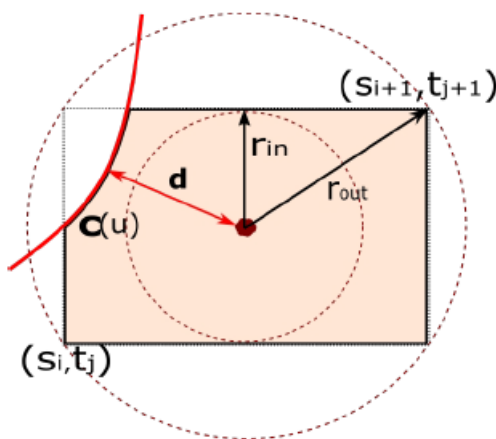
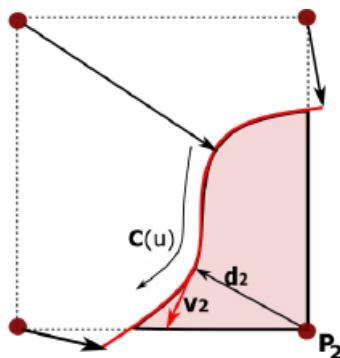


Figure 9: 1<sup>st</sup> searching step [19]

With the second searching step, the trimmed or unknown element is specified in a more precise way. A discrimination of all vertex points of the element is performed. If one or more vertex points are not active the element is trimmed.

Figure 10: 2<sup>nd</sup> searching step [19]

The two searching steps divide the trimmed element in three cases (see Figure 11):

1. three vertex-points are active
2. two vertex-points are active
3. one vertex-point is active

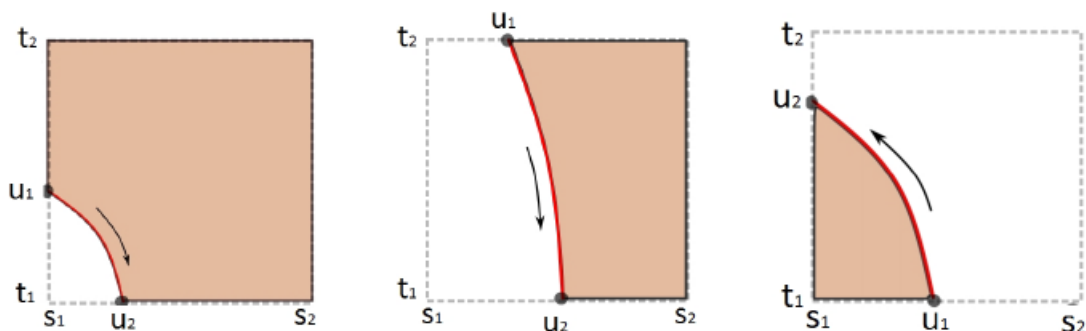


Figure 11: the three cases of trimmed elements: three vertex points active (left), two vertex-points are active (middle), one vertex-points are active (right) [19]

These two searching steps have some constraints regarding the elements.

- the sign of the tangential vector at any point on the trimming curve needs to be equal
- no closed curves inside an element
- one trimming curve per element
- trimming curve needs to cross at least one vertex
- the trimmed element has to be one of the three cases (see Figure 11)

If one of these constraints is violated refinement will mostly solve the problem.

The next problem of trimmed elements is the mapping. As mentioned before, the mathematical description of the element does not change after the trimming. This means

the Gaussian quadrature points are at the same place as before the trimming. If these Gaussian quadrature points are used, an area which does not exist any more contributes to the stiffness matrix and the element gets stiffer than it is. This is the reason why the Gaussian quadrature points have to be rearranged for the trimmed elements.

There are several different methods to do that, but the three most important ones are the NURBS enhanced triangles method by Kim et al. [12, 13], Adaptive Gaussian integration procedure by Breitenberger [3] and blending function method by Guo et al. [7]. The three methods use different approaches to map the Gaussian quadrature points. NURBS enhanced triangles method divides the element into normal triangles and NURBS curved triangles. The normal triangles were integrated with Gauss quadrature points for triangles. The NURBS curved triangles use several mappings to transform the Gaussian space to the parametric space. The Adaptive Gaussian integration procedure tries to remodel the trimmed element by an untrimmed element in the Gaussian space. The blending function method reconstructs the physical part of the trimmed element by dividing the trimming curve at all intersectional points. The interested reader is referred to Guo et al. All three methods have drawbacks and advantages. In the end Schönfeldt decides to use the blending function method.

For the blending function method the elements need to be described by quadrilateral or triangular elements. If this is not the case the elements have to be divided into more elements until every element is represented by a quadrilateral or triangular element. Also every element case has to be defined in this way it is orientated in the parametric space, because the element case can be rotated by  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  or  $270^\circ$  (see Figure 12).

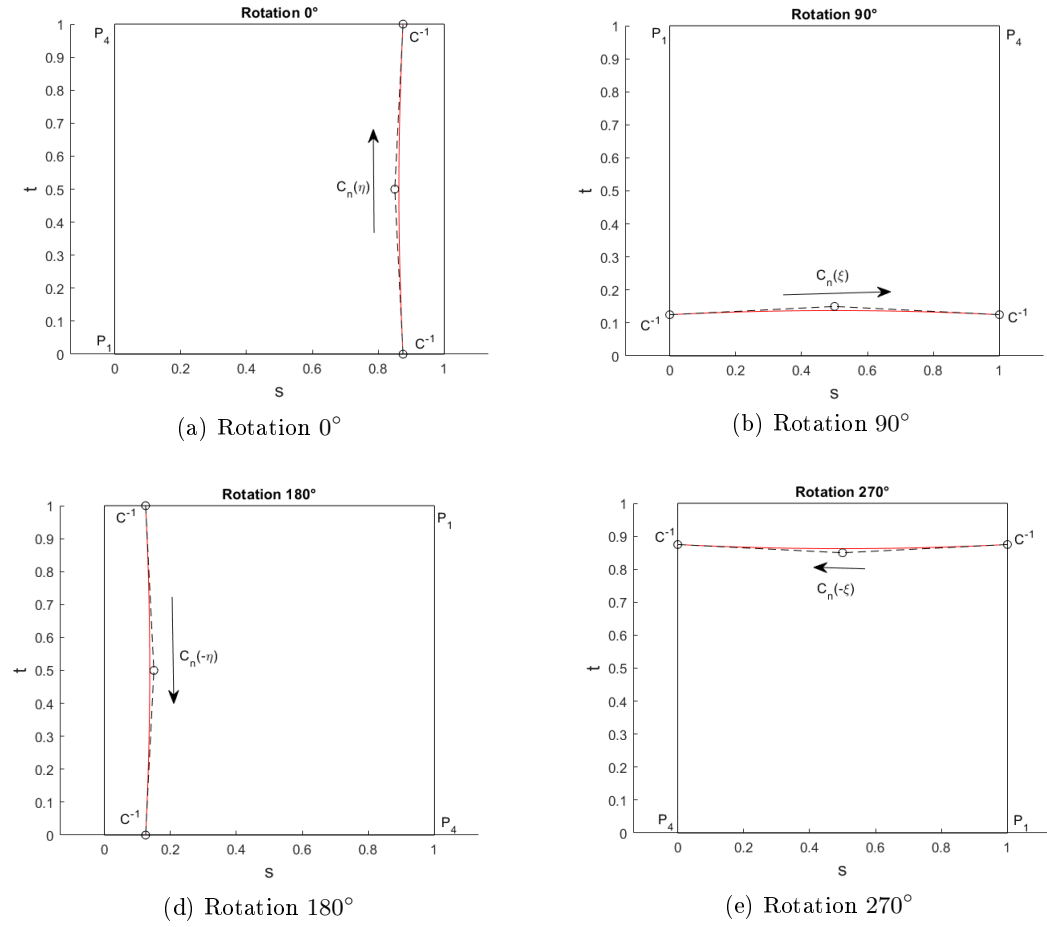


Figure 12: Rotation of quadrilateral trimmed elements

For every rotation a different mapping is needed. In the following the four mappings for the quadrilateral element are shown:

0° rotation:

$$\begin{bmatrix} s \\ t \end{bmatrix} = \frac{1}{4}((1 - \xi)(1 - \eta)P_1 + (1 - \xi)(1 + \eta)P_4) + \frac{1}{2}C_n(\eta)(1 + \xi) \quad (24)$$

90° rotation:

$$\begin{bmatrix} s \\ t \end{bmatrix} = \frac{1}{4}((1 - \xi)(1 + \eta)P_1 + (1 + \xi)(1 + \eta)P_4) + \frac{1}{2}C_n(\xi)(1 - \eta) \quad (25)$$

180° rotation:

$$\begin{bmatrix} s \\ t \end{bmatrix} = \frac{1}{4}((1 + \xi)(1 + \eta)P_1 + (1 + \xi)(1 - \eta)P_4) + \frac{1}{2}C_n(-\eta)(1 - \xi) \quad (26)$$

270° rotation:

$$\begin{bmatrix} s \\ t \end{bmatrix} = \frac{1}{4}((1 + \xi)(1 - \eta)P_1 + (1 - \xi)(1 - \eta)P_4) + \frac{1}{2}C_n(-\xi)(1 + \eta) \quad (27)$$

The Jacobian matrix for the quadrilateral element is as follows:

$$J_R = \begin{bmatrix} \frac{\partial s}{\partial \xi} & \frac{\partial t}{\partial \xi} \\ \frac{\partial s}{\partial \eta} & \frac{\partial t}{\partial \eta} \end{bmatrix} \quad (28)$$

The mapping for triangular elements can be found in Kudela [14].



## 4 Improvements of IGA and the internal research code

For the upcoming task to model parts of the R/V ‘Gunnerus’ the existing IGA code for trimmed surfaces needs some improvements in regard of efficiency. The improvements will be based on the existing code because writing an own code will be too complex and is not part of this master thesis.

At the moment the implementation for analysis of trimmed surfaces works only for trimmed surfaces and there is another implementation which is working for non trimmed surfaces. This second code is a lot more complex and works with a higher efficiency, which is why the idea is to implement the trimmed surface analysis into the second code. After merging the two codes, the implementation for analysis of trimmed surfaces reads in trimmed and untrimmed surfaces and commits for untrimmed surfaces the geometrical data of the refined patch, such as control points, knot vector etc. For trimmed patches the implementation stores the coordinates of the Gauss integration points, plotting points and further information to couple patches along the trimming curve and the geometrical description.

After converting the two implementations, different approaches, such as neglecting small trimmed elements, knot removal and polynomial degree for the Gauss integration points were implemented and discussed to improve the efficiency of the code.

### 4.1 Validation of the existing IGA trimming code

S. Schönfeldt programmed her own IGA code for trimmed surfaces. Furthermore there is another in-house research code, which is used for non trimmed analysis. After coupling these two codes, some tests have been done to check if the merged version works in the correct way. The first test is performed with a section of the deck of ‘Gunnerus’. The deck contains two trimming curves and consists of one patch. It is fixed on the two edges with no trimming curve and loaded with a point load in the middle of the deck. The expected result is a symmetric deformation. As Figure 13 shows, the deformation is not symmetric. Therefore mistakes exist in the code.

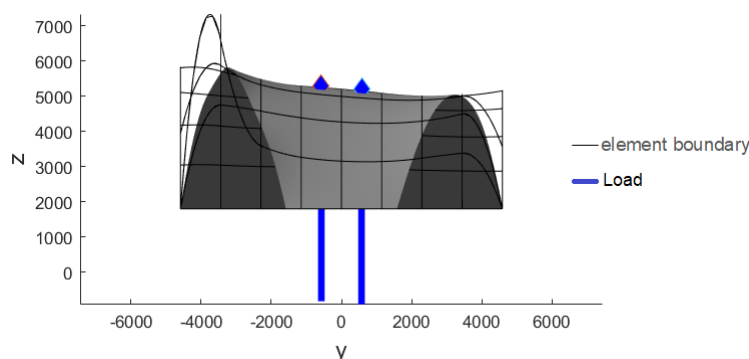


Figure 13: Deck-1 wrong deformation

Investigations with comparison of the stiffness matrix and mapping show that the

problem lies in the code of trimmed surfaces. The mapping of special element cases are not correct. These element cases are 7, 8, 9 and 10. They have two active vortex points and one trimming curve, but they are orientated differently. That means that the trimming curve lies in element case 7 on the left side in the parameter space and in element case 8 on the right side. The same is valid for element case 9 and 10. Element case 9 has the trimming curve at the upper side and element case 10 at the lower side. For more precise investigations, every element case is modelled by a patch containing one element. Then the patch is loaded in the middle with a point load. The results are shown in Figure 14. It can be seen that the deformation for element case 7 and 10 is wrong. The deformation cannot be positive and negative at the same time for one element. The Figure 14a shows the two element cases 7 and 10 within one plot, because due to rotation of the plot the patch can have the same orientation. The same is valid for Figure 14b, which shows the correct deformation. In both figures the surface is the trimmed area, the black line shows the element boundaries and the triangle the boundary conditions. All four patches are fixed on the two sides next to the trimming line. The problem was a wrong derivative in the mapping script for quadrilateral and triangular trimmed elements, which belong to the element cases 7 and 10.

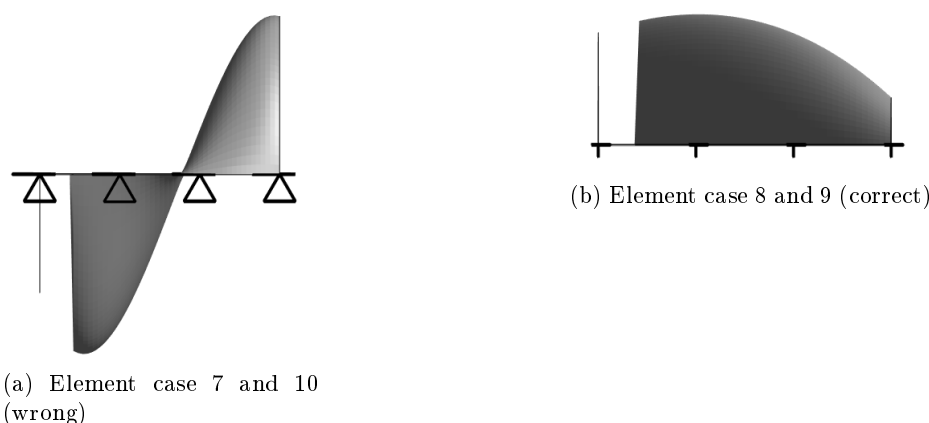


Figure 14: deformation of element cases 7, 8, 9, 10

Further tests with a symmetric trapezium and a plate with a hole are repeated to validate the changes in the code. These two geometries are selected because they can be modelled in an easy way without trimming curve. For that the two geometries are modelled with trimmed curves and then computed with the original code and the modified version, as well without trimmed curves. The untrimmed model is used as reference, because previous tests have shown that they are correct. For all analyses the same input parameters are used, which are shown in Table 4. The values for the load and thickness do not have any realistic character. They are defined in a way to have a result which is not unrealistic. The tables in the Appendix 10 and 11 show the result for the analysis. The column called ‘displacement’ describes the position in parametric space of the measurement points of the displacement. The Figures 15 and 16 show the result in a plot. It can be seen that the

results for both geometries for the untrimmed and trimmed modified case convert fast to the exact result. The original code shows for all investigated points no convergence.

Table 4: input parameters for validation analysis

<i>Trapezium</i>		<i>1/4 plate with hole</i>	
<i>Parameter</i>	<i>Value</i>	<i>Parameter</i>	<i>Value</i>
E	$210000 \frac{N}{mm^2}$	E	$210000 \frac{N}{mm^2}$
$\nu$	0,3	$\nu$	0,3
Thickness	100mm	Thickness	1mm
Point load	2000N	Point load	200N
P1	(0,5/0,5)	P1	(0,5/0,5)
degree of surface	2	degree of surface	2

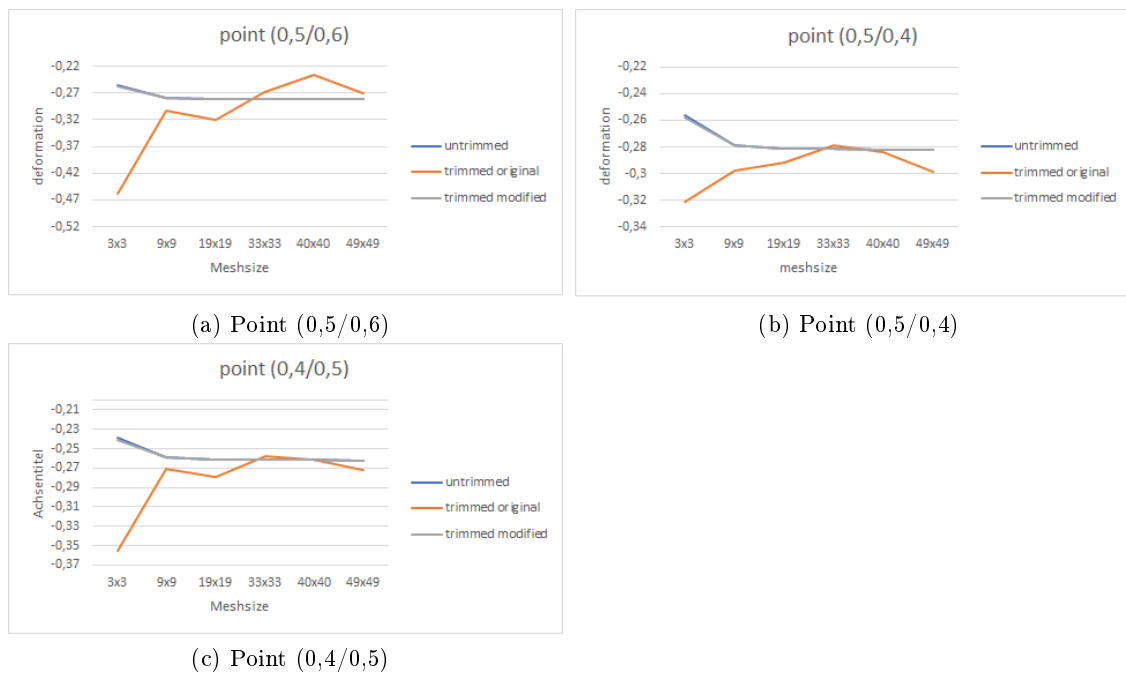


Figure 15: deformation on three different points on 'Trapezium'



Figure 16: deformation of 'plate with hole'

## 4.2 Flying Nodes

Control points with a very small influence on the trimmed surface can change their position during analysis. That means that they can be far away from the deformed trimmed surface. Figure 17 shows a plot of control points (red circles), where one control point is far away from the surface. These control points are called ‘flying nodes’ [3]. They do not influence the solution of the analysis but they influence the condition of the stiffness matrix. In consequence the stiffness matrix can be singular and the solution is not valid anymore. A solution is to exclude these elements from the calculation of Gauss integration points and stiffness matrix. These elements get inactive and the control points are added to the supports.

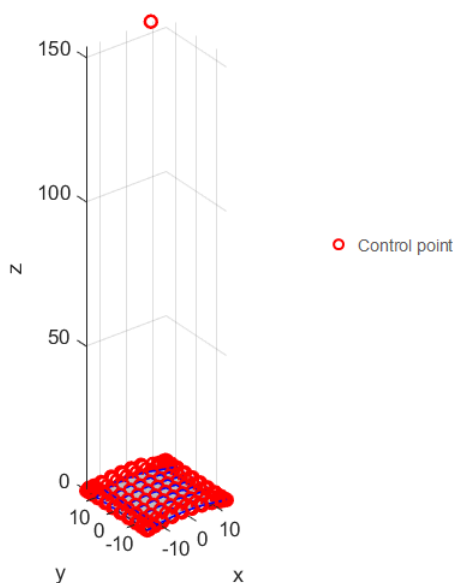


Figure 17: Flying nodes

The idea is to find a ratio between the element size and the trimmed area size. If this size is smaller than a specific value, this element will be inactive and no Gauss integration points will be computed. Instead the control point (‘Flying node’) will be added to the supports. To have a more efficient code only the area ratio for element cases with one or two active vertex points are calculated. The element cases with three active vertex points will never have a small area ratio so that flying nodes will appear. The area of the element and the area of the trimmed area are calculated in the parametric space. As a simplification the trimmed curve is assumed to be straight. For element cases with one active vertex point is the area a triangle and for two active vortex points a rectangular. Another simplification for the trimming curve in the case of two active vortex points is that the trimming curve enters and exits the element on the same height. This can result in big deviation between the calculated area and the real area. For instance if the shape of the real area is almost a triangle, on the one side the curve enters the element quite high

and on the other side it exits the element quite low, the calculated area will be around two times bigger or half than the real area. The area of the rectangular is computed by the length of the element and the height of one of the two intersection points, without comparing them or calculating the average height of the intersection points. Nevertheless, if the variation between the height of the two intersection points is high, the area ration between the trimmed domain and the element area will not be small and therefore these elements will be always active. If the element have different sub elements, due to inner knots, the area of every subelement will be added up to the total area.

Another problem showed up with the elements with one vortex point but with different sub elements due to inner knots. If an element has sub elements only one sub element will have a triangular shape, the other ones will have a rectangular shape (s. Figure 18). The problem is that the sub element is still specified as a element with one vortex and the area will be calculated as a triangle, which makes the area smaller than it is. At the end all the areas of the sub element are summed up to the total trimmed area and compared with the total element area. This simplification can influence the decision if the element should be active or non active, but in the following benchmark tests this simplification do not show big influence on the result. Perhaps on more complex structure and more inner knots, this can affect the result and has to be investigated in more detail.

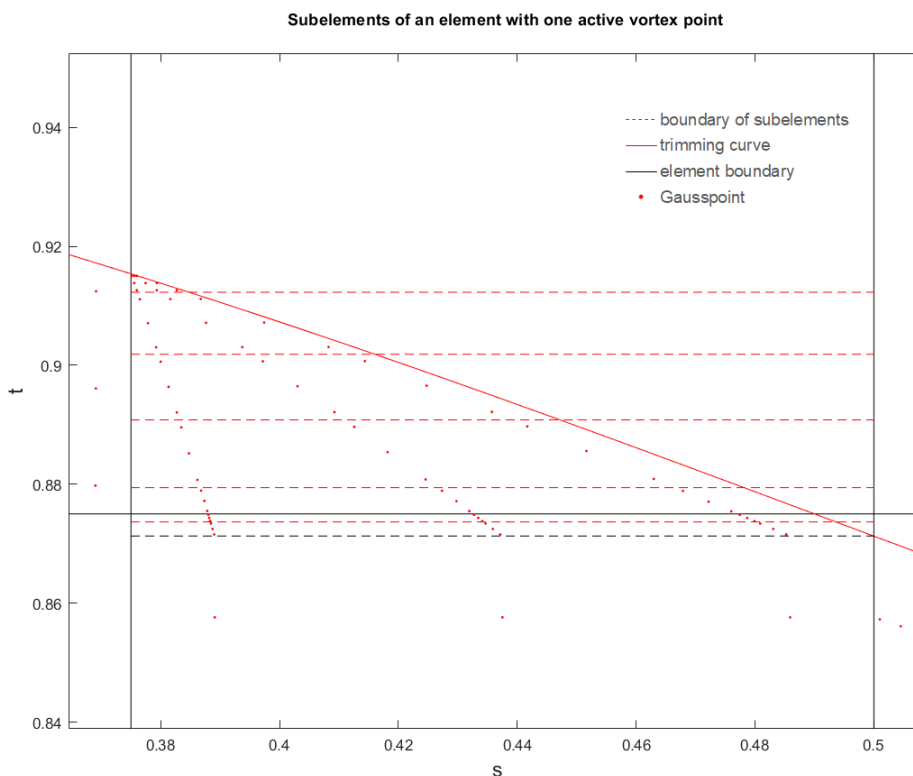


Figure 18: Sub elements of elements with one active vortex point

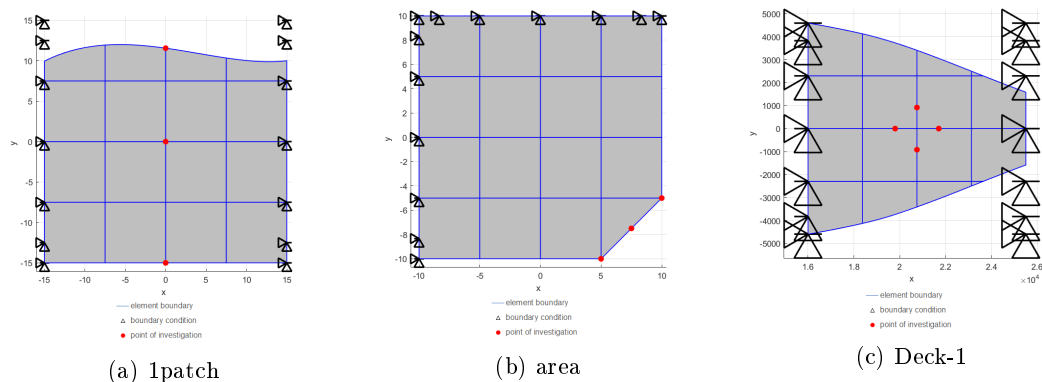


Figure 19: geometries for benchmark test flying nodes

#### 4.2.1 Benchmark test for flying nodes

The investigation of which area ratio is small enough to influence the condition of the matrix in a good way is done with a simple benchmark test with three different geometries. The first investigation is related to which area size should be neglected. Therefore all three geometries are simulated with different meshes to create elements with small trimmed areas. Afterwards, the condition of the stiffness matrix is checked in regards to singularity under which area ratio. The first geometry is a rectangular with a trimming curve on the upper side called ‘1patch’ (s. Figure 19a). The second geometry is a square with a trimmed edge called ‘area’ (s. Figure 19b). These two geometries are exemplarily modelled geometries and therefore the third geometry is from ‘Gunnerus’ and is called Deck-1 (s. Figure 19c). All geometries are fixed on two sides in all three directions and are loaded with an area load. Again the values for the load and thickness are chosen freely. The values are not based on any realistic case.

Table 5 shows the condition of the stiffness matrix for the case that all trimmed elements are used to calculate the global stiffness matrix. After analysing the result it is decided to test what the condition of the global stiffness matrix will be if trimmed elements under 1% and 3% are neglected. Likewise, the deformation of different locations on the surface will be checked to evaluate the influence on the result due to neglecting areas. The positions of measurement are indicated by red dots in Figure 19.

The exact results for the deformation and condition of stiffness matrix are shown in the appendix B. The condition of the stiffness matrix improves for neglecting elements with less than 1%. For the comparison of the deformation only the result with non singular matrices are allowed to take into account. The results do not change for the different mesh sizes between the results of all elements are active and only elements with an area ratio bigger than 1% . For higher polynomial degrees, such as four the stiffness matrix is in some cases still singular. If  $eps = 0,03$  the condition of the stiffness matrix is even better but for smaller meshes the accuracy is worse than for 1%. For example for ‘1patch’ result for the deformation at point (0,5/0,5) converge to 1,393mm. With the mesh size 8x8 and polynomial degree 2 the result for 1% is 1,3741mm and for 3% it is 1,3738mm. With

Table 5: results of the convergence study

(a) condition of stiffness matrix for Deck-1

$A_{min}$	$Mesh$	$deg$	$matrix$
0,035	8x8	2	ok
		3	ok
		4	ok
0,0261	16x16	2	ok
		3	ok
		4	singular
0,0053	10x10	2	ok
		3	singular
		4	singular
7,67e-4	25x25	2	ok
		3	singular
		4	singular

(b) condition of stiffness matrix for area

$A_{min}$	$Mesh$	$deg$	$matrix$
0,0312	7x7	2	ok
		3	ok
		4	ok
0,0312	15x15	2	ok
		3	ok
		4	singular
0,001	33x23	2	singular
		3	singular
		4	singular
2,98e-4	21x45	2	singular
		3	singular
		4	singular

(c) condition of stiffness matrix for 1patch

$A_{min}$	$Mesh$	$deg$	$matrix$
0,0106	25x25	2	ok
		3	singular
		4	ok
0,0015	8x8	2	ok
		3	ok
		4	singular
3,41e-4	16x16	2	singular
		3	singular
		4	singular
3,47e-5	10x10	2	singular
		3	singular
		4	singular

Table 6: input parameter for benchmark test (flying nodes)

$parameter$	$geometry$	$value$
E	all	$210000 \frac{N}{mm^2}$
$\nu$	all	0,3
Thickness	1patch	2mm
	area	2mm
	Deck-1	100mm
area load	1patch	20N
	area	20N
Point load	Deck-1	2000N
Pl	Deck-1	(0,5/0,5)
degree of surface	all	3
eps for small areas	all	0,01

the increase of mesh size this difference is getting less but it shows the influence of setting small elements without to inactive. Therefore for further simulation all elements with a smaller area ratio of 1% is neglected.

### 4.3 Knot removal

One big disadvantage of IGA is that in some cases there are more Gauss integration points than necessary to have a correct result. Figure 20 shows that in some elements exist more Gauss points than in others. The reason for the extra Gauss integration points are internal knots in the element. The trimming curve is described by a knot vector. This knot vector can contain internal knots, which lead to extra Gauss points in an element.

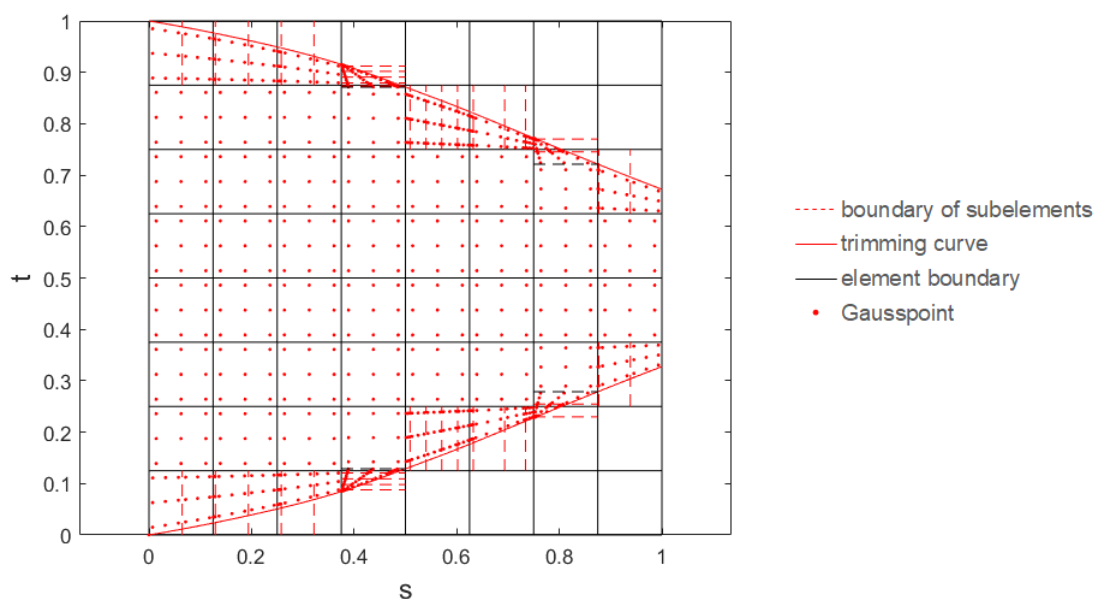


Figure 20: Gauss points

To compute the Gauss integration point the internal research code first adds knots at the intersection points and than splits the trimming curve in smaller curves. The new curves are starting from one knot and ending at the next knot. For every curve the Gauss integration points are computed, which means if there are two sections of the trimming curve in one element there will be a double amount of Gauss integration points. In Figure 20 the red dashed lines shows the position of the knot, where the curve of one element is split in sub-curves. The idea is to remove the internal knot of one element after adding the knots on the intersection points.

The  $p$ -th degree NURBS curve defined on a knot vector  $U = (u_1, \dots, u_m)$ , control points  $P = (P_1, \dots, P_n)$ , weights  $w = (w_1, \dots, w_n)$ , has a knot which can be removed  $u_r$ . This knot cannot be a boundary knot, which is why we have  $p + 1 < r < m - p$ . Due to the removal of one knot the number of control points will be decreased by one. The new control points  $Q = (Q_1, \dots, Q_{n-1})$  and weights can be derived as follows [16]:



$$Q_i^w = P_i^w, i = 1, \dots, r - p - 1 \quad (29)$$

$$Q_{i-1}^w = P_i^w, i = r, \dots, n \quad (30)$$

For  $r - p \leq i \leq r - 1 - p/2$

$$Q_i^w = \frac{P_i^w - (1 - \alpha_i)P_{i-1}^w}{\alpha_i} \quad (31)$$

with

$$\alpha_i = \frac{u_r - u_{i+1}}{u_{i+p+2} - u_{i+1}} \quad (32)$$

For  $r - 1 - p/2 \leq j \leq r - 2$

$$Q_j^w = \frac{P_j^w - (1 - \alpha_j)P_{j+2}^w}{1 - \alpha_j} \quad (33)$$

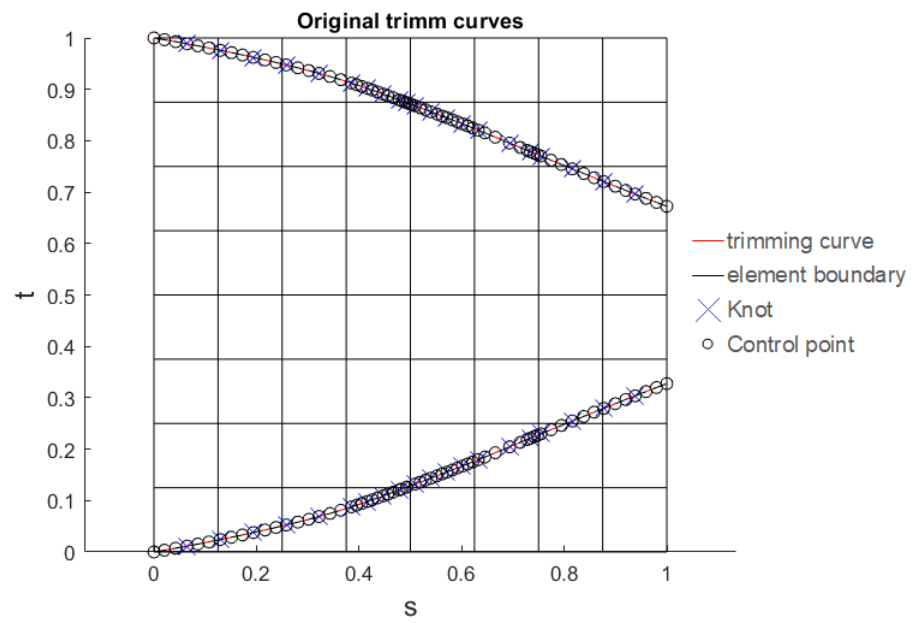
with

$$\alpha_j = \frac{u_r - u_{j+1}}{u_{j+p+2} - u_{j+1}} \quad (34)$$

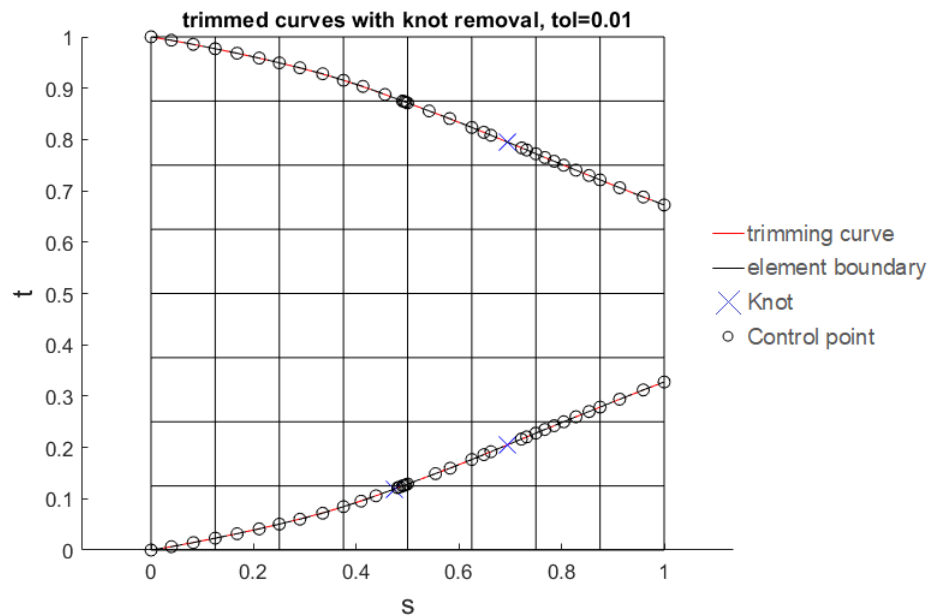
With the calculation of the new control points and weights the new NURBS curve is set.

In comparison to adding knots to a knot vector the removal of knots can change the shape of the curve. Some complicated shapes need at least three control points, which is why a tolerance need to be specified. In the internal research code the distance between the old and new NURBS curve is calculated. If the distance is smaller than the tolerance the internal knot will be removed.

Figure 21a shows the original trimmed curves for the patch (Deck-1) from Gunnerus. The black circles represent the control points and the blue crosses mark the inner knots. The knots on the intersection points are not plotted. The Figure 21b plots the trimmed curve with removed inner knots. The lower trimming curve has only two knots left and the upper one only one. Also the control points are a lot less. This knot removal is done with a tolerance of 0,01 and shows how the knot removal works on NURBS.



(a) original trimming curves



(b) removed knots, with tolerance 0,01

Figure 21: Knot removal for Deck-1

After computing the Gauss integration points for the new trimming curves, Figure 22 shows less Gauss integration points than the original Gauss integration points in Figure 20.

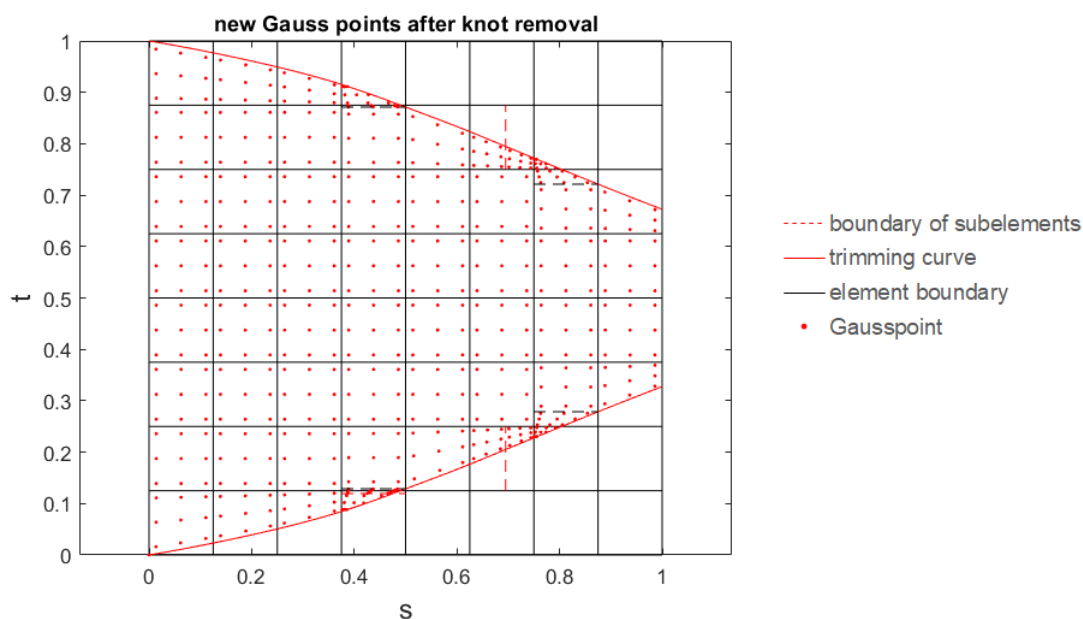


Figure 22: new Gauss points after knot removal, Tol 0,01

The new Matlab code for removing knots is implemented in the existing research code and the value for the tolerance of the changes between original and new NURBS-curve must be set.

#### 4.3.1 Benchmark test for knot removal

To figure out a good value the geometry of ‘Deck-1’ is used. That geometry has a lot of inner knots and is therefore suitable to test out the knot removal function. Table 7 shows the input parameter and Figure 23 shows the boundary conditions and the points where the displacements are measured for references (blue dots). Four points are selected close to the trimming curve, because at these points most changes are expected to happen due to slope changes.

Table 7: input parameter for benchmark test of knot removal on Deck-1

<i>parameter</i>	<i>value</i>
E	$210000 \frac{N}{mm^2}$
$\nu$	0,3
Thickness	100mm
Point load	2000N
P1	(0,5/0,5)
degree of surface	3
eps for small areas	0,01

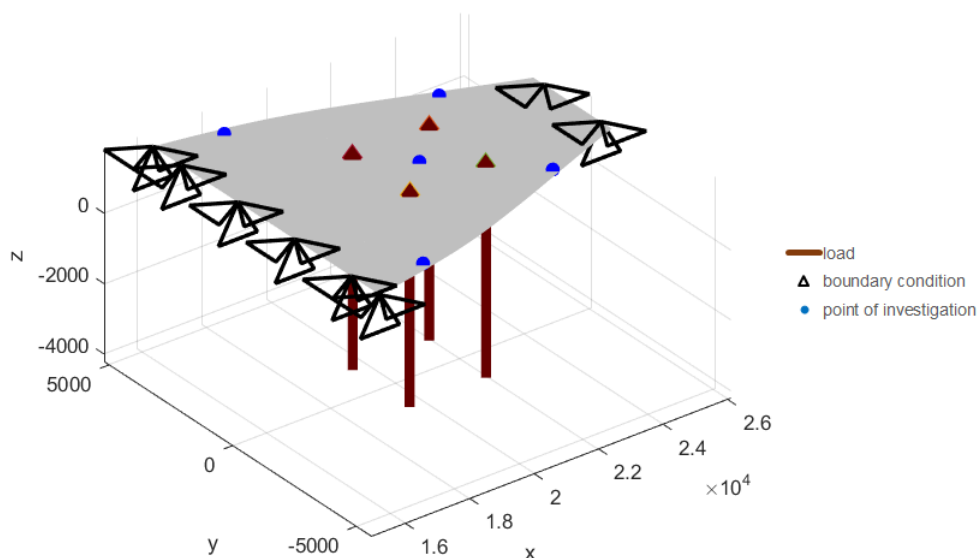


Figure 23: Points of measurements

Two analyses are performed: one with 0 tolerance, which means no knot is removed and one with 0,1. At this value there are no inner knots for the elements. Both analyses run with different mesh sizes until convergence appears. The exact values can be looked up in the Appendix C and Figure 24 shows the convergence of the test. The plot shows only three of the five investigation points, because two points are symmetric and have the same result. It can be seen that the results between all inner knots and without inner knots are the same for meshes finer than 8x8 and the results for mesh 4x4 varies a little bit. Therefore a tolerance of 0,1 seems good for this specific case.

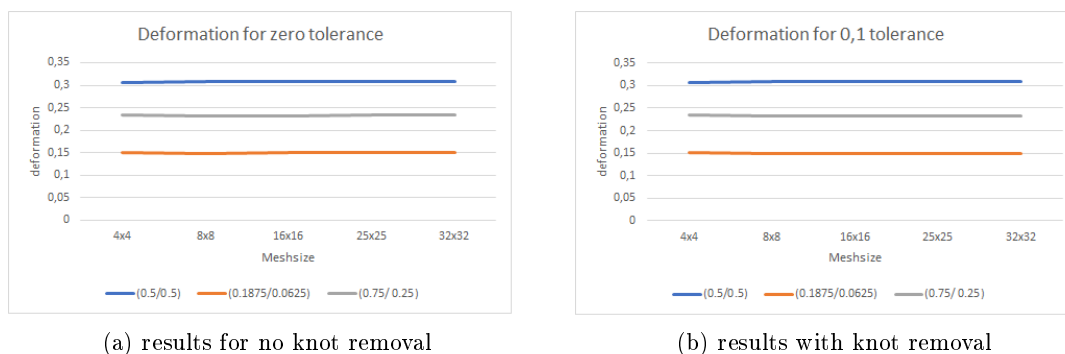
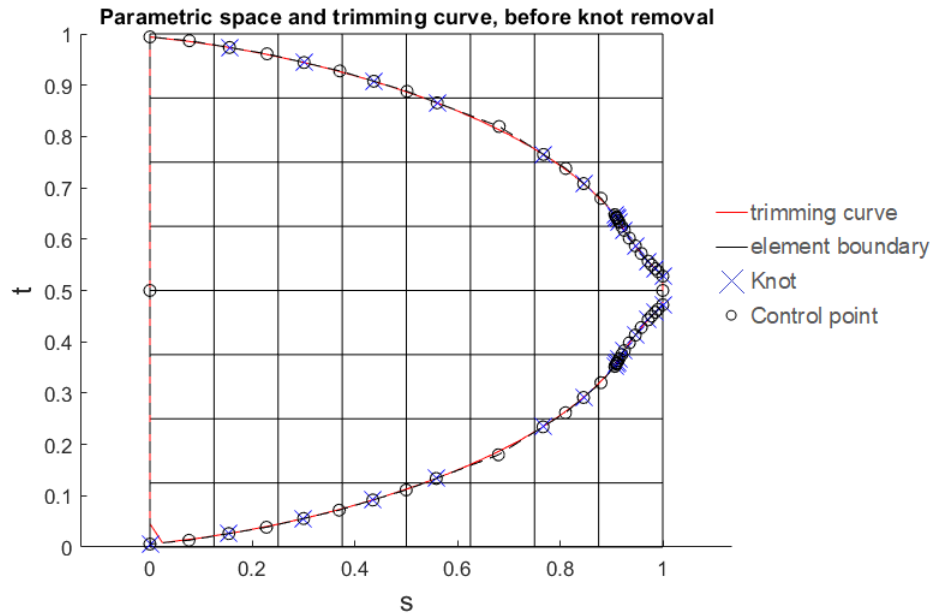


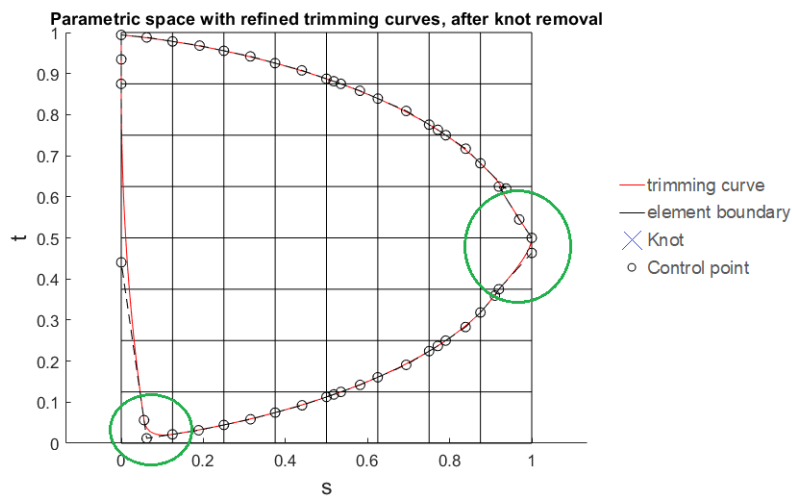
Figure 24: results of benchmark test for knot removal

In cases of one trimming curve around the whole patch, which can be often found on the structural model of Gunnerus, the slope of the trimming curve changes. Figure 25 shows the frame 50. Figure 25a shows the original trimming curve with all knots and Figure 25 shows the removal of all inner knots with a tolerance of 0,1. In the two areas marked with a green circle the shape of the trimming curve changes a lot. In that areas there are kinks of the trimming curve and without internal knot the kinks cannot represent

in a correct way. As described later on in 5.2 one trimming curve around a patch causes problems with the existing code this will not be investigated in this thesis. However it will be investigated in more detail in the future. For the simulation later on, the tolerance for knot removal will be set to 0,1 because for the cases which are analysed in this thesis, this number shows satisfactory accuracy.



(a) original shape



(b) shape with removed knots

Figure 25: changes on trimming curve due to knot removal

#### 4.4 Degree for Integration

As described in 4.3 IGA has sometimes too many Gauss integration points, which makes the computation inefficient. In the research code the polynomial degree of the surface will be compared with the polynomial degree of the trimming curve and then the maximum

value will be chosen. The polynomial degree decides over the amount of Gauss integration points. If the polynomial degree is three, there will be four gauss points for full integration and three for reduced integration. In general  $GP = p + 1$  for full and  $GP = p$  for reduced integration, with  $p$  polynomial degree and  $GP$  number of Gauss integration points.

The geometry ‘Deck-1’ which was used in the benchmark test for knot removal contains trimming curves with polynomial degree of 3 and the NURBS surface has a polynomial degree of 1. The tests which are performed in the benchmark test for knot removal show that there are differences between polynomial degree 2 or 3, which is why the existing code will not be changed due to having a more precise solution with less elements.

#### 4.5 Plotting of trimmed surface

The implementation of the trimming script into the in-house research code demands changes in the script of plotting trimmed surfaces in physical space. The modified version works with the function of computing the Gauss integration points. The function computes instead of the coordinates of Gauss integration points the coordinates of points which can be used to plot the surface later on. The points will be symmetric all over the element and the number of points per element is defined by the number of element and the resolution. The resolution specifies the total number of points in a row over the whole patch. In other words for the same resolution the number of plotting points is proportional to the number of elements and vice versa.

The surface is plotted element wise with the plotting surface function in Matlab. In this way the mesh for the trimmed surface can be plotted. The mesh shows the boundaries of every element which is why a line around the element should be plotted. A challenge is to sort the points on the element boundary in the correct way to plot a line in the correct order from one point to the other. For single elements without subelements, the plotting of the mesh works in a good way. For elements with subelements due to inner knots, the function needs further improvements.

## 5 Gunnerus



Figure 26: ‘Gunnerus’ [21]

Until now the internal research code was only tested for simple trimming cases and maximum one patch. These are special cases and not practical for investigating a whole system. Therefore the implementation is extended and improved to combine multiple trimmed surfaces and investigated on a complex system. In this case, it will be the research vessel of NTNU ‘Gunnerus’ (Figure 26). The research vessel Gunnerus was built in 2006 by Larsnes Mekansike Verksted for the Norwegian University of Science and Technology (NTNU). The main dimensions of the ship are listed in Table 8.

Table 8: main dimensions of Gunnerus

Length over all	31,25m
Length between pp	28,90m
Length in waterline	29,90m
Breath middle	9,60m
Breath extreme	9,90m
Depth	4,20m
Draught	2,70m
Dead weight	107t

The following chapter presents the work with a digital twin with respect to IGA as well the problem which occurs during that work. These problems will be discussed, too.

## 5.1 Digital twin Gunnerus and structural model

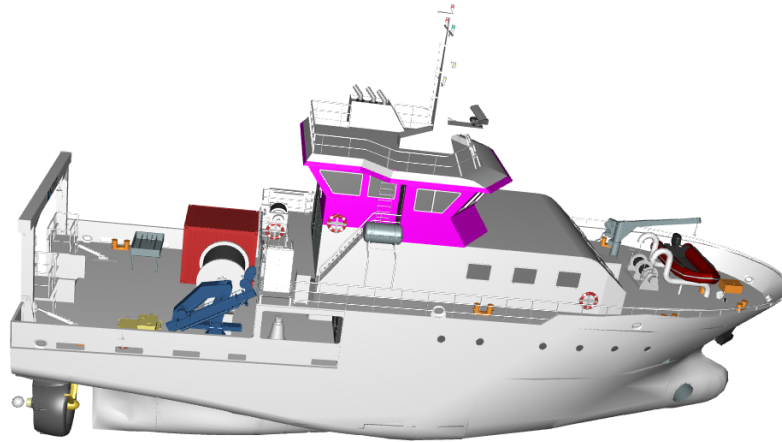


Figure 27: Gunnerus as a digital twin

In the summer 2018 five students of NTNU built a digital twin of Gunnerus as a cooperation between DNV-GL and NTNU (s. Figure 27). Both parties have plans to use the digital twin in the future. NTNU plans to use the digital twin for teaching. Then the students have an object on which they can work for example the effects of different sea states acting on the digital twin/ship. For DNV-GL the main focus lies on supporting their project ‘Approved’. The project contains a new digital standard for the exchange of information and digital models between shipyards, designers and classification societies. In order to achieve this, they need information from tests, which the digital twin of Gunnerus can provide [5]. The result of the work of the students is a digital twin in the basics and needs further investigation and work to create a fully operational digital twin. This is not part of this master thesis and the information which are needed to apply IGA to Gunnerus as a digital twin are included in the existing digital twin.

The digital twin contains also a structural model of Gunnerus based on CAD drawing. This CAD drawing (s. Figure 28) is used for testing and investigating the internal research code.



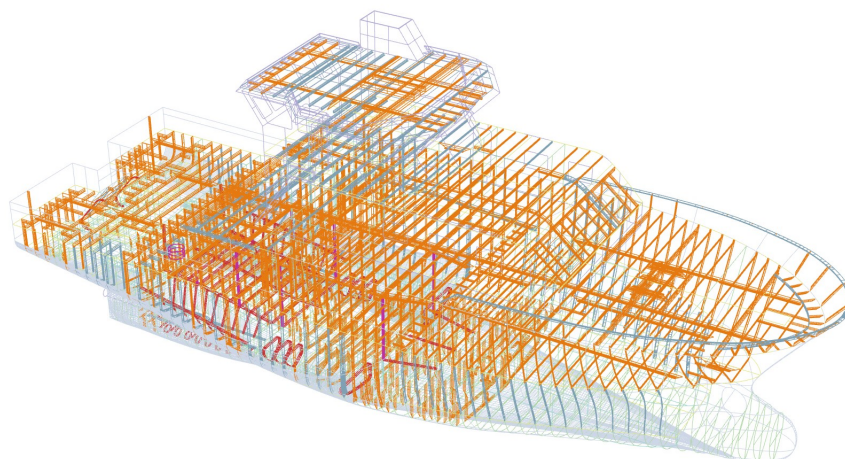


Figure 28: Gunnerus as a CAD drawing with all structural elements

The structural model contains 11888 single patches due to the high accuracy of modeling, for example every stiffener is modelled. Another reason why the amount of patches is so high is the usage of shell elements, every stiffener is modelled out of 6 patches. Further the structural model do not have any information about the thicknesses. To analyse them, every thickness has to be added by hand to the IGA code by studying the classification drawings. Therefore it is not suitable to analyse the whole ship because it will take too much time to couple every single patch to each other and adding every specific thickness to the specific patch. That will be out of the range of this master thesis. Thus only small areas of Gunnerus will be investigated with the IGA code.

## 5.2 Problems with structures of Gunnerus

As mentioned before there exists too many patches and the lack of information regarding thickness in the structural model, which is why the plan is to investigate only small areas of Gunnerus. During the search of suitable areas another problem shows up. Almost all patches are modelled with only one trimming line around the whole patch. Figure 29 shows generic the frame 50. The red line shows the trimming curve around the patch in the parameter space. The resulting Gauss integration points are plotted in Figure 30. The Gauss integration points are not correct for the elements 1, 32, 40 and 57. The element 1 is detected as element case 1, that means this element is not available for analysis; element 57 is detected as -1, which means it is an inactive element and element 32 and 40 are detected as element case 8, which means element with 3 active vortex points. The correct element cases would be for element 1 case 7, element 32 case 8 (two active vortex points), element 40 case 6 and element 57 case 5 (3 active vortex points).

These problems of finding the correct element cases are caused by the trimming curve, which lies on the boundary. In the following the problem is explained on element 1. The

element has two vortex points on the trimming line. On these points the trimming line is straight, therefore the crossproduct is 0. This means that the tangential vector on the trimming line and the distance vector from the vortex point to the trimming line are collinear/parallel. This result gains no information in regards to inactive or active vortex points. The vortex point in the right bottom corner has a negative crossproduct and is therefore inactive. The vortex point in the right bottom corner has a positive crossproduct and is therefore active. This special trimming case is not covered by the implemented element cases in the research code neither at any other research ([7, 12]).

In the previous test this was not a problem because the import-IGES script deletes all straight lines. The import-IGES script assumes all boundaries as trimming lines. The result without deleting the straight lines is that corner elements will have more than one trimming curve, which the code cannot handle. Boundary lines are in general straight lines and trimming lines are most of the time not straight. This is the reason why straight lines are deleted. This simplification causes an other limitation of the existing code. If the trimming line is straight and is not on the boundary, the trimming line will be deleted and is not available for further analysis.

In the case of frame 50 this solution does not work because it is modelled by only one curve. This curve has straight parts but also curved parts, which is why it is not deleted and causes problems to find the correct element case.

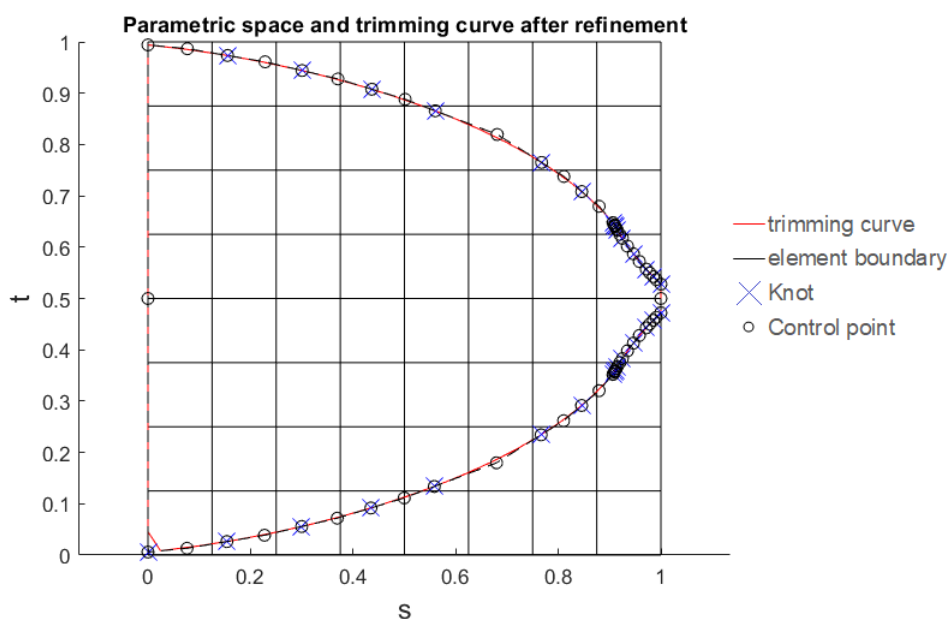


Figure 29: frame 50 of Gunnerus in parametric space

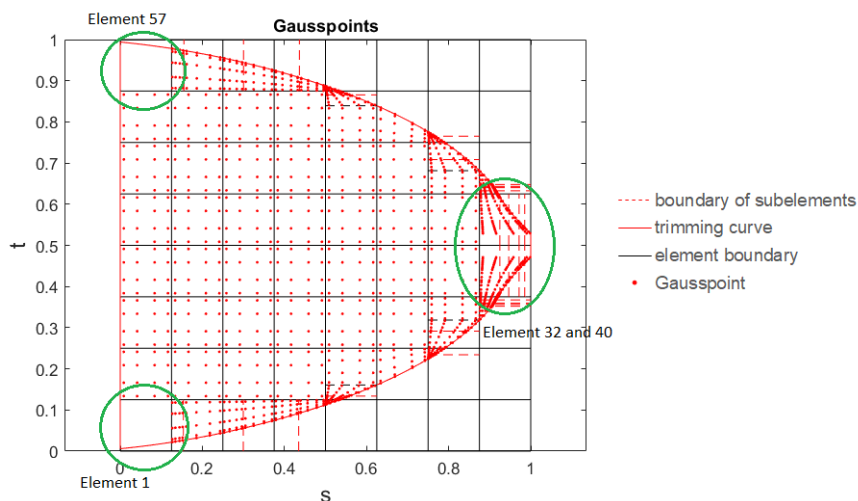


Figure 30: Gauss integration points for frame 50 in original shape

A possible solution to fix this problem is to *remodel* the patch in a patch with two trimming curves. That could be a solution for really small structures but is not feasible for bigger structures, such as ‘Gunnerus’. The result would be remodelling the whole ship and than the advantage of IGA would have been disappeared in regards to the classical FEM.

The frame 50 is remodelled in an exemplary manner, to show that remodelling fix this problem (Figure 31). The remodelled frame is not as precise in the tip as the original one, but the element cases are the correct ones, same as the Gauss integration points. The orientation of the remodelled frame is different in parametric space but the physical space is still the same. This does not have any influence on the analysis except for applying the load or possible patch coupling. The coordinates should switch from  $u$  to  $v$  and  $v$  to  $u$  to have the same result if the original geometry would work.

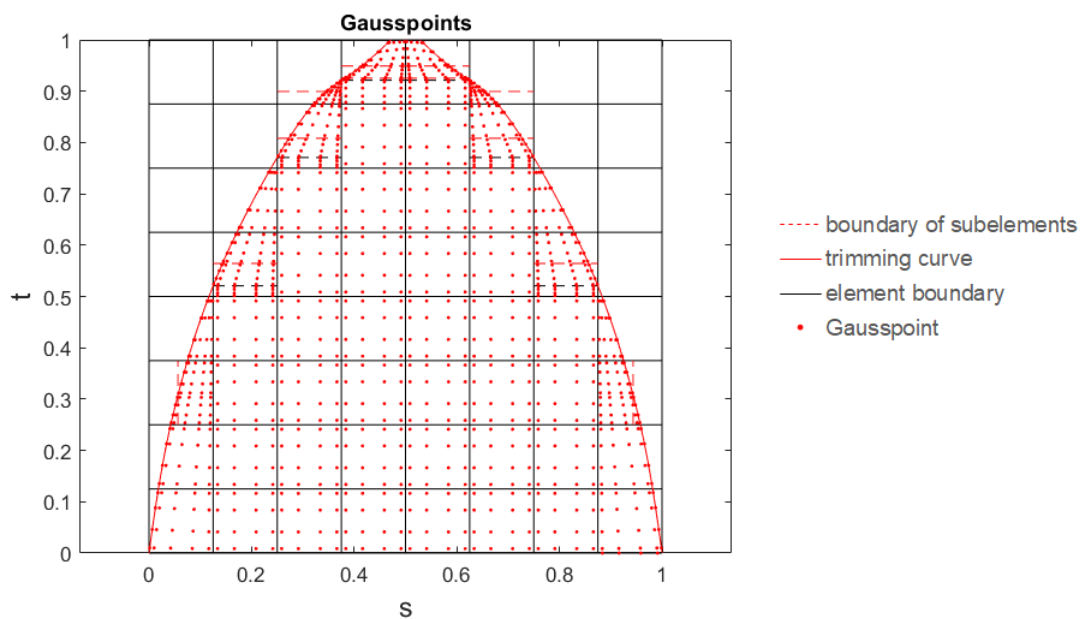
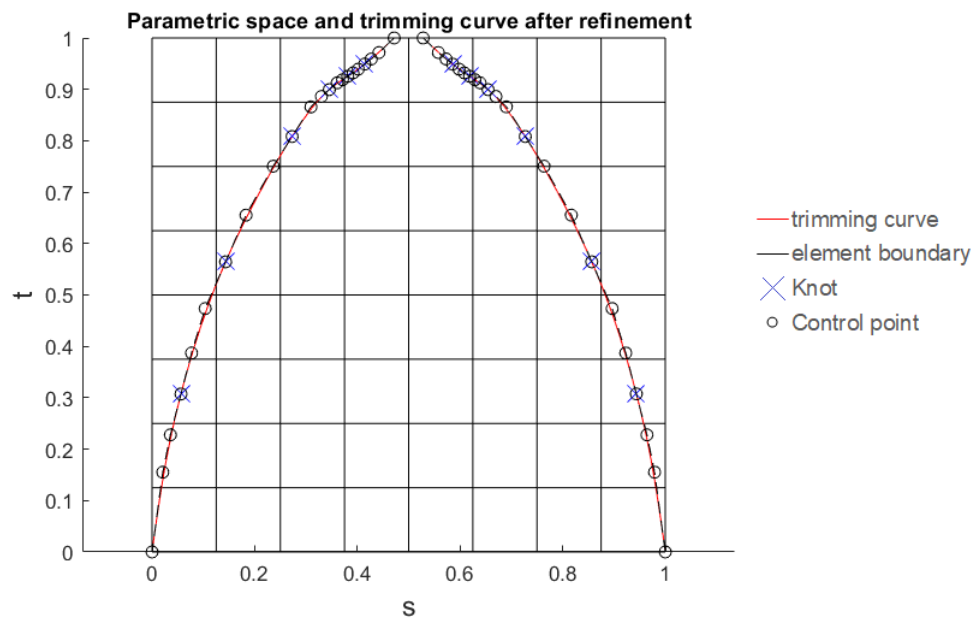


Figure 31: remodelled frame 50

### 5.3 Simulation with parts of Gunnerus

In Chapter 4 the patch ‘Deck-1’ was often used for different benchmark tests. It is a patch from ‘Gunnerus’, which originally has two holes inside the patch. For simplification, These are deleted and the trimming curve around the patch was split in four parts. Therefore the problem with the trimming curves along the boundary is solved and the patch could be analysed. The result for ‘Deck-1’ can be seen in Chapter 4. Figure 32 shows the original

geometry with two holes and the deformed shape. The deformation of the simplified model is not representative. Normally the deck is clamped on all four edges and has stiffeners underneath. Due to the explained problems and handicaps with the digital twin and IGA-code, the boundary condition and load case is chosen to show the potential of IGA.

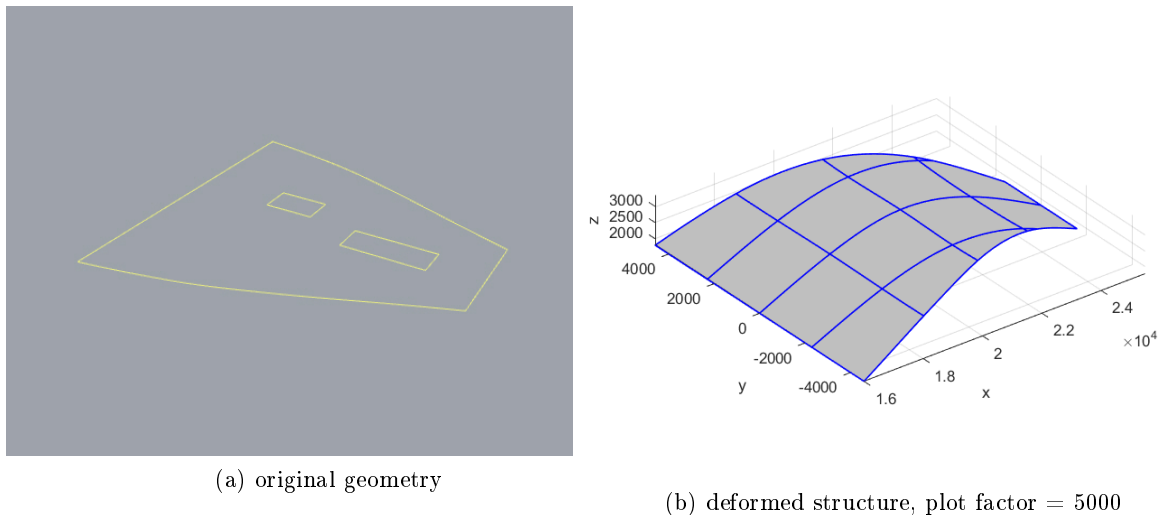


Figure 32: Deck-1

The other geometry, which is analysed, is a part of the upper deck and contains of two patches. Figure 33 shows the geometry and the position at ‘Gunnerus’. The geometry is modelled out of two patches. The upper curved patch is an untrimmed patch. The other patch is trimmed. One trimming curve goes around the whole patch, which is why this patch is remodelled. Afterwards the outer trimming line is split in six curves. Two of the six are trimming lines. The three windows are modelled in that way, that one trimming line is one window. The small square, which can be seen on Figure 33 is not a structural element, which is why it is not modelled.

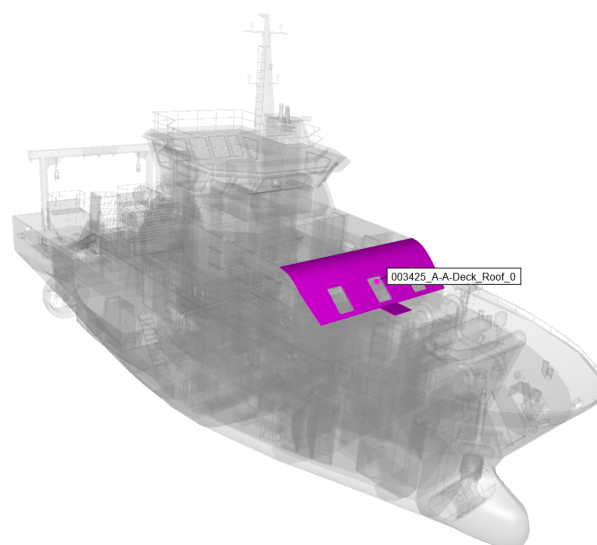


Figure 33: second geometry of Gunnerus

To perform the analysis the load case and input parameter have to be defined. The input parameters are shown in Table 9. The thickness is the real one of the that structure but the stiffeners are excluded, the boundary conditions and line load are not inspired by reality. Therefore the deformation does not represent a possible deformation in real life. The geometry with boundary conditions and line load is plotted in Figure 35 the deformed structure, as well.

Table 9: input parameters

<i>parameter</i>	<i>value</i>
Young's modulus E	$210000 \frac{N}{mm^2}$
Poisson's ratio	0,3
Thickness	6mm
line load	-0,2N
degree of surface	3
eps for small areas	0,01

To prove the solution, the analysis is run with different mesh sizes and the deformation is measured at one point at every patch and is checked if it converges. The exact deformation for the convergence test are shown in Table 15 in the Appendix. The measurement point on the untrimmed patch is at 0,5/0,5 in parametric space and at the trimmed patch at 0,2/1.

The results in Figure 34 do not show convergence yet. The convergence test is stopped with the final mesh size of 80x70 for the trimmed patch and 30x130 for the untrimmed patch. Finer meshes cannot be simulated due to lack of computational power with the provided computers. The existing results show already a tendency of convergence. Therefore the results seem to be correct.

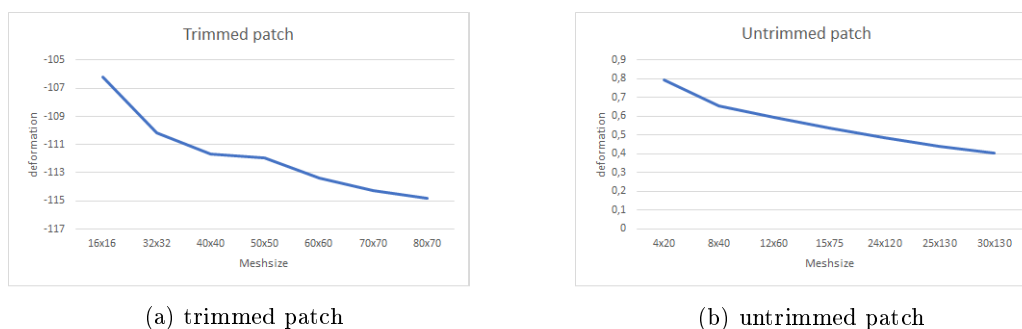
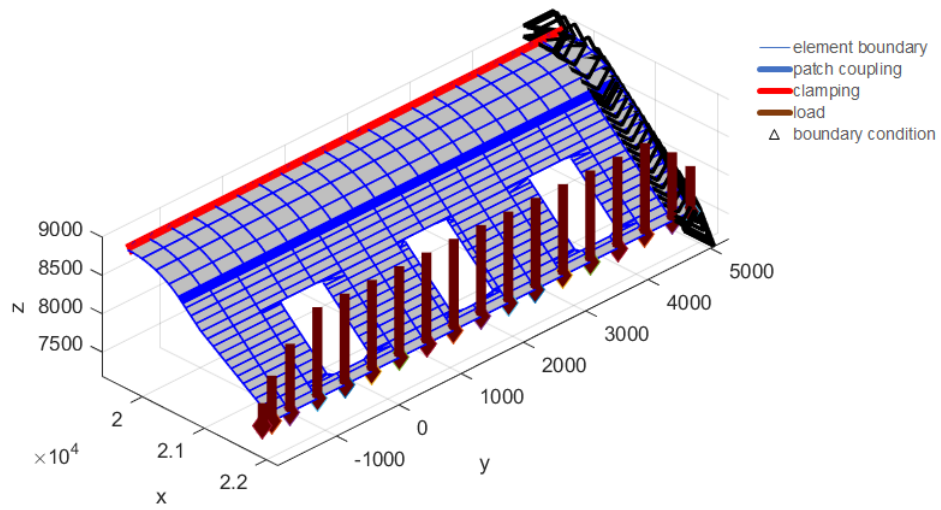
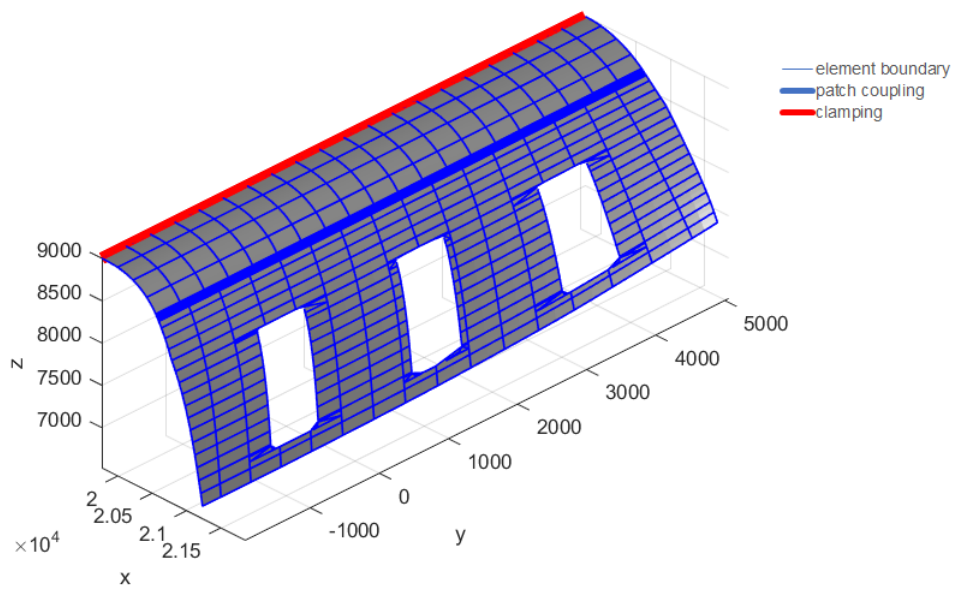


Figure 34: results of convergence study



(a) boundary conditions and load



(b) deformed structure, plot factor 10

Figure 35: Deck roof

## 6 Conclusions and Outlook

The scope of the master thesis is to improve the efficiency and extend the existing IGA in-house research code, so that the code can analyse the digital twin of the R/V ‘Gunnerus’.

To reach the goal of a more efficient in-house research code, the implementation for analysis on trimmed surfaces is merged into the implementation for analysis on untrimmed surfaces. The trimmed surface implementation works as a preprocessing step and calculates the Gauss integration points for computing the stiffness matrix, which is done in the untrimmed surface implementation and in the solving, as well. The merging decreases the amount of computational time and some tests show the reliability of the merging.

Further the code was extended with different implementations, such as knot removal or neglecting small trimmed elements. The neglecting of small trimmed elements under 1% of the untrimmed element size seems to be a good estimate. For the investigated geometries, the condition of the stiffness matrix changed in a positive direction and for most cases the stiffness matrix is not singular anymore. The knot removal works in his basics. For trimming curves without a kink the tolerance of 0,1 seems to be a good value, but for trimming curves with a kink the shape of the trimming curve changes a lot. Therefore more investigation in regards of a suitable tolerance needs to be performed in the future.

Besides the improvements for the analysis functions, the IGES-import functions needed some upgrades. The script can read single and multiple patches as well as untrimmed and trimmed ones, which is why it can be used for more complex structures now.

The digital twin of ‘Gunnerus’ contains all structural elements of the ships, which ends up in a huge number of surfaces. This ends up in a lot of work, because the existing in-house research code needs information of every patch coupling and thicknesses of every surfaces, which has to implemented by hand. Moreover, the structural model does not provide information about the thickness of individual surfaces. The information about the thicknesses provide the classification drawing. Therefore more work needs to be done in this direction. How can the structural contain the thickness of shell elements, how are they stored. The IGES-file contains information in regards to boundaries of a surface or curve. It would increase the efficiency of the IGA but further work is needed to extract that information and implementing it in the code.

Finally, the analysis of ‘Gunnerus’ failed on another point. The actual number of different trimmed element cases are not enough to analyse ‘Gunnerus’. Most surfaces are modelled with one trimming curve around it. This causes problems due to a trimming curve on more than one element edge. This element case is not covered by the existing element cases for trimmed surfaces. This is the reason why the cases for trimmed elements have to be extended or more detailed research needs to be done to solve that problem. By remodelling the part, the solution in this thesis is not suitable for big structures.

For the investigated cases, IGA showed great results and the advantage of using the modelled geometry directly without complex meshing was really handy. The world of engineering is ruled with simplification and for some cases it is useful to have a really



---

detailed model but it is suitable to have such a detailed model all the time. Sometimes only a rough analysis is enough. In that case a model with almost 12000 patches is not ideal due to computational effort and a simplified model would be better. It would be interesting to investigate how detailed the model needs to be to give a satisfactory result and how to construct a simplified model out of a really detailed CAD drawing.

---

## References

- [1] *Initial Graphics Exchange Specification IGES 5.3*. U.S. Product Data Association, 1996.
- [2] M Breitenberger, A Apostolatos, B Philipp, R Wüchner, and K-U Bletzinger. Analysis in computer aided design: Nonlinear isogeometric b-rep analysis of shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:401–457, 2015.
- [3] Michael Breitenberger. *CAD-integrated design and analysis of shell structures*. PhD thesis, Technische Universität München, 2016.
- [4] Carl De Boor, Carl De Boor, Etats-Unis Mathématicien, Carl De Boor, and Carl De Boor. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.
- [5] Digitread. Collaborating on building a digital twin of the research vessel r/v gunnerus, June 2018.
- [6] Thang X Duong, Farshad Roohbakhshan, and Roger A Sauer. A new rotation-free isogeometric thin shell formulation and a corresponding continuity constraint for patch boundaries. *Computer Methods in Applied Mechanics and Engineering*, 316:43–83, 2017.
- [7] Yujie Guo, Jason Heller, Thomas JR Hughes, Martin Ruess, and Dominik Schillinger. Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the step exchange format. *Computer Methods in Applied Mechanics and Engineering*, 336:39–79, 2018.
- [8] Austin J Herrema, Emily L Johnson, Davide Proserpio, Michael CH Wu, Josef Kiendl, and Ming-Chen Hsu. Penalty coupling of non-matching isogeometric kirchhoff–love shell patches with application to composite wind turbine blades. *Computer Methods in Applied Mechanics and Engineering*, 2018.
- [9] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39-41):4135–4195, 2005.
- [10] Josef Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Technische Universitaet Muenchen, 2010.
- [11] Josef Kiendl, K-U Bletzinger, J Linhard, and Roland Wüchner. Isogeometric shell analysis with kirchhoff–love elements. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52):3902–3914, 2009.
- [12] Hyun-Jung Kim, Yu-Deok Seo, and Sung-Kie Youn. Isogeometric analysis for trimmed cad surfaces. *Computer Methods in Applied Mechanics and Engineering*, 198(37-40):2982–2995, 2009.

- 
- [13] Hyun-Jung Kim, Yu-Deok Seo, and Sung-Kie Youn. Isogeometric analysis with trimming technique for problems of arbitrary complex topology. *Computer Methods in Applied Mechanics and Engineering*, 199(45-48):2796–2812, 2010.
- [14] László Kudela. Highly accurate subcell integration in the context of the finite cell method. *Master’s Thesis, Technical University Munich*, 2013.
- [15] C Lovadina, A Reali, and G Sangalli. What is isogeometric analysis. 2014.
- [16] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012.
- [17] Eric Reissner. The effect of transverse shear deformation on the bending of elastic plates. *J. appl. Mech.*, pages A69–A77, 1945.
- [18] David F Rogers. *An introduction to NURBS: with historical perspective*. Elsevier, 2000.
- [19] Sofie Schønfelddt-Borchgrevink. Isogeometric analysis with trimmed geometries applied to ship hulls. Master’s thesis, NTNU, 2018.
- [20] Mike Shafto, Mike Conroy, Rich Doyle, Ed Glaessgen, Chris Kemp, Jacqueline LeMoigne, and Lui Wang. Draft modeling, simulation, information, technology & processing roadmap, technology area 11. 2010.
- [21] Fredrik Skoglund. Pictures of gunnerus.
- [22] Bradford M Smith and Joan Wellington. Iges, a key interface specification for cad/calm systems integration. *Computer-Aided Geometry Modeling*, page 279, 1983.

## Appendix

## A Validation of merging the two codes

<i>mesh</i>	<i>displacement</i>	<i>untrimmed</i>	<i>trimmed original</i>	<i>trimmed modified</i>
3x3	0,4/0,5	-0,2385	-0,3547	-0,2410
	0,6/0,5	-0,2431	-0,3587	-0,2451
	0,5/1(0,5/0,125)	-0,2563	-0,4586	-0,2578
	0,5/0,6(0,5/0,875)	-0,2563	-0,3213	-0,2578
9x9	0,4/0,5	-0,2587	-0,2706	-0,2589
	0,6/0,5	-0,2655	-0,2719	-0,2656
	0,5/1(0,5/0,125)	-0,2785	-0,3027	-0,2787
	0,5/0,6(0,5/0,875)	-0,2785	-0,2977	-0,2787
19x19	0,4/0,5	-0,2613	-0,2785	-0,2613
	0,6/0,5	-0,2681	-0,2876	-0,2681
	0,5/1(0,5/0,125)	-0,2812	-0,3198	-0,2813
	0,5/0,6(0,5/0,875)	-0,2812	-0,2918	-0,2813
33x33	0,4/0,5	-0,2617	-0,2579	-0,2617
	0,6/0,5	-0,2686	-0,2633	-0,2686
	0,5/1(0,5/0,125)	-0,2817	-0,2681	-0,2817
	0,5/0,6(0,5/0,875)	-0,2817	-0,2787	-0,2817
40x40	0,4/0,5	-0,2618	-0,2607	-0,2618
	0,6/0,5	-0,2687	-0,2684	-0,2687
	0,5/1(0,5/0,125)	-0,2818	-0,2370	-0,2818
	0,5/0,6(0,5/0,875)	-0,2818	-0,2836	-0,2818
49x49	0,4/0,5	-0,2620	-0,2718	-0,2620
	0,6/0,5	-0,2689	-0,2872	-0,2689
	0,5/1(0,5/0,125)	-0,2818	-0,2703	-0,2818
	0,5/0,6(0,5/0,875)	-0,2818	-0,2985	-0,2818

Table 10: trapezium

<i>mesh</i>	<i>displacement</i>	<i>untrimmed</i>	<i>trimmed original</i>	<i>trimmed modified</i>
3x3	0,2/0	-0,0357	-0,0357	-0,0325
9x9	0,2/0	-0,0388	-0,0361	-0,0389
19x19	0,2/0	-0,0393	-0,0444	-0,0393
33x33	0,2/0	-0,0394	-0,0401	-0,0394
40x40	0,2/0	-0,0394	-0,0420	-0,0394
49x49	0,2/0	-0,0394	-0,0412	-0,0394

Table 11: plate with hole (left,down)

## B Benchmark test flying nodes

<i>ipatch</i>				<i>area</i>				<i>Deck-1</i>						
Mesh	deg	u	Matrix	Mesh	deg	u	Matrix	Mesh	deg	u	Matrix			
8x8	2	1,3741		7x7	2	2,7105		8x8	2	0,2794				
		1,55				2,7565				0,2906				
		1,5702				2,7105				0,2995				
	3	1,3929			3	2,7116			3	0,2995		0,2833		
		1,5701				2,7578				0,2949				
		1,5903				2,7116				0,3039				
	4	1,393			4	2,7116			4	0,2834		0,295		
		1,5701				2,7579				0,3039				
		1,5905				2,7116				0,3039				
	10x10	2	1,3809			15x15	2		2,7114		10x10	4	0,2808	
			1,5572						2,7576				0,2921	
			1,5776						2,7114				0,3011	
3		1,393		3	2,7116			2	0,3011			0,2832		
		1,5701			2,758		singular		0,2949					
		1,5904			2,7116		singular		0,3039					
4		1,393		4	2,7116		singular	3	0,3039			0,2834		
		1,5701			2,758		singular		0,295					
		1,5906			2,7116		singular		0,304					
16x16		2	1,3883		33x23		2	2,7116		16x16		4	0,304	
			1,5651					2,7579					0,3027	
			1,5855					2,7115					0,3027	
	3	1,393		3		2,7116	singular	2	0,2822			0,2937		
		1,5701				2,758	singular		0,3027					
		1,5906				2,7116	singular		0,3027					
	4	1,393	singular	4		2,7116	singular	3	0,2833			0,2948		
		1,5701	singular			2,758	singular		0,3038					
		1,5907	singular			2,7116	singular		0,3038					
	25x25	2	1,391			21x45	2	2,7115			25x25	3	0,2831	singular
			1,5681					2,7578					0,2948	singular
			1,5885					2,7115					0,3038	singular
3		1,393	singular	3	2,7116		singular	4	0,3038			0,2832	singular	
		1,5701	singular		2,758		singular		0,2948	singular				
		1,5907	singular		2,7116		singular		0,3038	singular				
4		1,393	singular	4	2,7116		singular	4	0,3038	singular		0,3038	singular	
		1,5701	singular		2,758		singular		0,3038	singular				
		1,5905	singular		2,7116		singular		0,3039	singular				
		2					2					2	0,2828	
													0,2944	
													0,3034	
	3			3				3	0,3034			0,2832		
									0,2948					
									0,3039					
	4			4				4	0,3039			0,2832	singular	
									0,2949		singular			
									0,3039		singular			

Table 12: results of benchmark test  $eps = 0,01$

<i>ipatch</i>				<i>area</i>				<i>Deck-I</i>						
Mesh	deg	u	Matrix	Mesh	deg	u	Matrix	Mesh	deg	u	Matrix			
8x8	2	1,3738		7x7	2	2,7105		8x8	2	0,2794				
		1,5497				2,7565				0,2906				
		1,5699				2,7105				0,2995				
	3	1,393			3	2,7116			3	0,2995		0,2833		
		1,5701				2,7578				0,2949				
		1,5904				2,7116				0,3039				
	4	1,393			4	2,7116			4	0,3039		0,2834		
		1,5701				2,7579				0,295				
		1,5905				2,7116				0,3039				
	10x10	2	1,3809			15x15	2		2,7114		10x10	4	0,2808	
			1,5572						2,7576				0,2922	
			1,5776						2,7114				0,3012	
3		1,393		3	2,7116			2	0,3012			0,2833		
		1,5701			2,758		singular		0,295					
		1,5904			2,7116		singular		0,304					
4		1,393		4	2,7116		singular	3	0,304			0,2835		
		1,5701			2,758		singular		0,2952					
		1,5906			2,7116		singular		0,3041					
16x16		2	1,3883		33x23		2	2,7114		16x16		4	0,2822	
			1,5651					2,7577					0,2938	
			1,5855					2,7114					0,3028	
	3	1,393		3		2,7115		2	0,3028			0,2832		
		1,5701				2,7579			0,2949					
		1,5906				2,7115			0,3039					
	4	1,393	singular	4		2,7115	singular	3	0,3039			0,2832	singular	
		1,5701	singular			2,7579	singular		0,2949		singular			
		1,5907	singular			2,7115	singular		0,3039		singular			
	25x25	2	1,391			21x45	2	2,7093			25x25	2	0,2832	
			1,5681					2,756					0,2949	
			1,5885					2,7095					0,3038	
3		1,393		3	2,7094			3	0,3039			0,2835		
		1,5701			2,7561				0,2953					
		1,5907			2,7095				0,3042					
4		1,393	singular	4	2,7094		singular	4	0,3043			0,2836	singular	
		1,5701	singular		2,7561		singular		0,2953	singular				
		1,5905	singular		2,7095		singular		0,3042	singular				
									0,3044			0,3044	singular	

Table 13: results of benchmark test  $eps = 0,03$

## C Benchmark test knot removal

<i>tol = 0</i>			<i>tol = 0, 1</i>		
<i>Mesh</i>	<i>deformation</i>	<i>points</i>	<i>Mesh</i>	<i>deformation</i>	<i>points</i>
4x4	0,3078	(0.5/0.5)	4x4	0,3079	(0.5/0.5)
	0,1502	(0.1875/0.0625)		0,1503	(0.1875/0.0625)
	0,1502	(0.1875/0.9375)		0,1503	(0.1875/0.9375)
	0,2343	(0.75 / 0.25)		0,2345	(0.75 / 0.25)
	0,2343	(0.75 / 0.75)		0,2345	(0.75 / 0.75)
8x8	0,308	(0.5/0.5)	8x8	0,308	(0.5/0.5)
	0,1499	(0.1875/0.0625)		0,1499	(0.1875/0.0625)
	0,1499	(0.1875/0.9375)		0,1499	(0.1875/0.9375)
	0,2331	(0.75 / 0.25)		0,2331	(0.75 / 0.25)
	0,2331	(0.75 / 0.75)		0,2331	(0.75 / 0.75)
16x16	0,3084	(0.5/0.5)	16x16	0,3084	(0.5/0.5)
	0,15	(0.1875/0.0625)		0,15	(0.1875/0.0625)
	0,15	(0.1875/0.9375)		0,15	(0.1875/0.9375)
	0,2331	(0.75 / 0.25)		0,2331	(0.75 / 0.25)
	0,2331	(0.75 / 0.75)		0,2331	(0.75 / 0.75)
25x25	0,3083	(0.5/0.5)	25x25	0,3083	(0.5/0.5)
	0,15	(0.1875/0.0625)		0,15	(0.1875/0.0625)
	0,15	(0.1875/0.9375)		0,15	(0.1875/0.9375)
	0,2332	(0.75 / 0.25)		0,2332	(0.75 / 0.25)
	0,2332	(0.75 / 0.75)		0,2332	(0.75 / 0.75)
32x32	0,3084	(0.5/0.5)	32x32	0,3084	(0.5/0.5)
	0,15	(0.1875/0.0625)		0,15	(0.1875/0.0625)
	0,15	(0.1875/0.9375)		0,15	(0.1875/0.9375)
	0,2332	(0.75 / 0.25)		0,2332	(0.75 / 0.25)
	0,2332	(0.75 / 0.75)		0,2332	(0.75 / 0.75)

Table 14: results of benchmark test knot removal

## D IGES file

1		S
2	1H, ,1H; , ,	G
	1	
3	121HC:\Users\Gerrit\Documents\Schiffbaustudium\NTNU\3. Semester\Project	G
	2	
4	work\Sofie MATLAB\Sofie MATLAB\Trapez_ungetrimmed.igs,	G
	3	
5	26HRrhinoceros ( May 22 2017 ),31HTrout Lake IGES 012 May 22 2017,	G
	4	

```

6 32,38,6,308,15, G
   5
7 , G
   6
8 1.0D0,2,2HMM,1,0.254D0,13H190226.125408, G
   7
9 0.001D0, G
   8
10 9000D0, G
   9
11 , G
   10
12 , G
   11
13 10,0,13H190226.125408; G
   12
14 314 1 0 0 0 0 0 000000200D
   1
15 314 0 1 1 0 0 0 COLOR 0D
   2
16 406 2 0 0 1 0 0 000000300D
   3
17 406 0 -1 1 3 0 0 OLEVELDEF 0D
   4
18 128 3 0 0 1 0 0 000000000D
   5
19 128 0 -1 5 8 0 0 0 TrimSrf 0D
   6
20 116 8 0 0 1 0 0 000000000D
   7
21 116 0 -1 1 0 0 0 03d Point 0D
   8
22 116 9 0 0 1 0 0 000000000D
   9
23 116 0 -1 1 0 0 0 03d Point 0D
  10
24 116 10 0 0 1 0 0 000000000D
  11
25 116 0 -1 1 0 0 0 03d Point 0D
  12
26 116 11 0 0 1 0 0 000000000D
  13
27 116 0 -1 1 0 0 0 03d Point 0D
  14
28 314,0.0,0.0,0.0,20HRGB( 0, 0, 0 ); 0000001P
   1
29 406,2,1,8HStandard; 0000003P
   2 128,1,1,1,1,0,0,1,0,0,0.0D0,0.0D0,9219.544457292888D0,
      0000005P   3 9219.544457292888D0,0.0D0,0.0D0,8000.0D0
      ,8000.0D0,1.0D0,1.0D0, 0000005P   4 1.0D0,1.0D0,0.0D0,0.0D0,0.0D0

```



	,9000.0D0,2000.0D0,0.0D0,0.0D0,	0000005P	5
30	8000.0D0,0.0D0,9000.0D0,6000.0D0,0.0D0,0.0D0,		0000005P
	6		
31	9219.544457292888D0,0.0D0,8000.0D0;		0000005P
	7		
32	116,9000.0D0,6000.0D0,0.0D0,0;		0000007P
	8		
33	116,9000.0D0,2000.0D0,0.0D0,0;		0000009P
	9		
34	116,0.0D0,8000.0D0,0.0D0,0;		0000011P
	10		
35	116,0.0D0,0.0D0,0.0D0,0;		0000013P
	11		
36	S0000001G00000012D00000014P00000011		T
	1		

## E Convergence study for parts of Gunnerus

	<i>mesh size</i>	<i>deformation</i>
patch 1	16x16	-103,3503
patch 2	4x20	0,7330
patch 1	32x32	-108,5031
patch 2	8x40	0,7509
patch 1	40x40	-109,5090
patch 2	12x60	0,6948
patch 1	60x60	-110,9238
patch 2	24x120	0,6001
patch 1	70x70	-112,2072
patch 2	25x130	0,5553
patch 1	80x70	-112,8194
patch 2	30x130	0,5177

Table 15: results of convergence test

