
MATHEMATICAL MODELING OF FLEX CURVES IN CUSTOMIZED ALPINE SKIS USING ISOGEOMETRIC ANALYSIS

BJØRN HARALD SNERSRUD,
FREDRIK HILDRUM,
HERMAN GALTELAND,
JON VEGARD VENÅS
AND
MAGNUS NORUM

Spring 2014



Faculty of Information Technology, Mathematics and Electrical Engineering
NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

PREFACE

This booklet represents the outcome of a teamwork project for students in the course «TMA4850 - Experts in Teamwork - Mathematics in Applications» at the Norwegian University of Science and Technology (NTNU) during the Spring 2014 semester. The course agenda is to set up a larger group project with practical applications, where students from various disciplines come together and contribute with their academic skills. Moreover, a special focus is given to how groups operate and function, and facilitation during the teamwork stimulates continuous evaluation and implementation of concepts with regards to group effectivity.

With a quite ambitious project given the time constraints and previous knowledge, there have been several challenges during the course. But we feel the results are considerably higher than what was realistically expected initially.

We want to thank our supervisor Trond Kvamsdal both for outlining a project description on modeling flex curves in alpine skis, developed by ski producer Endre Hals, and support during the course. Also, a special thanks goes to Endre Hals for inviting us to his production facility, providing data during the modeling and giving us a pair of skis for testing purposes. In addition, we received great help from both Arne Morten Kvarving from SINTEF and Ph. D fellow Kjetil André Johannessen with regards to generating useful data structures for the ski model. At last we would like to mention our group facilitators Mai Sharoni and Sebastian Jørgensen.

Trondheim, May 2, 2014

Bjørn Harald Snersrud

Fredrik Hildrum

Herman Galteland

Jon Vegard Venås

Magnus Norum

TEAM STRUCTURE

In this section we present some details of the team structure and how the group's academic competences impacted the working process.

THE MAC-MODEL

Our team has worked under the governing village theme «Mathematics in Applications», and as a foundational philosophy the so-called MAC-model has been used. Introduced to us by the village supervisor Trond Kvamsdal, the *Mathematics, Applications and Computer science* model highlights the effective interactions between different scientific groups. Powerful modern teams within the areas of applied mathematics need to cooperate with people from the specific physical application, and implementations of numerical procedures benefit highly from expertise from the computer science world.

In Figure 1, our team structure according to the MAC-model is presented. Even though there are no experts in computer science, both Jon Vegard and Fredrik have solid experience with implementation of numerical models. Nonetheless, a dedicated member with skills in computer science could have managed the technicalities in applying for example parallel algorithms for better performance.

ACADEMIC COMPOSITION

With «mathematical modeling of flex curves in customized alpine skis using isogeometric analysis» as the project title, the necessity of both mathematical- and applicational skills is clear. Since Bjørn Harald's expertise lies in industrial mechanics, it was quite natural that he studied some of the underlying structural mechanics and designed the CAD ski model. Moreover, as Magnus

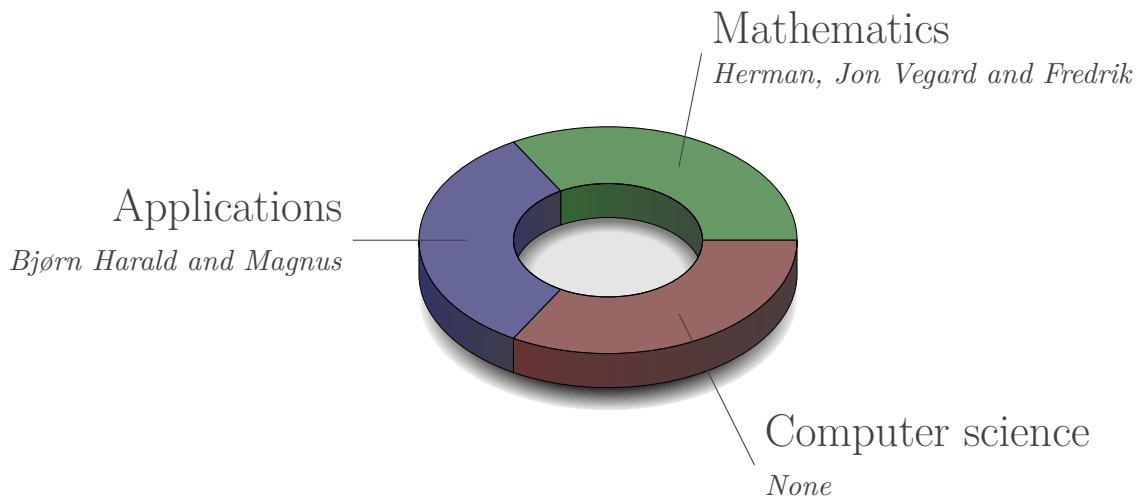


Figure 1: Team structure according to the MAC-model.

have knowledge on material properties and practical experience as a carpenter, he focused in particular on testing the materials in the ski core.

When it came to the necessary mathematical basis, both Jon Vegard and Fredrik had previously worked together with the finite element method. With this foundation, the topic of isogeometric analysis stood out as exciting and interesting. Since neither of them had knowledge on this subject, the task was quite ambitious. During the project it became evident, due to Jon Vegard's extensive contribution and gained knowledge from a specific course, that he operated the main mathematical analysis, both theoretically and numerically. Also, as Jon Vegard will continue with a master study on this topic, and thus saw this project as very useful practical preparation.

Herman is a master student in mathematics, with specialization in abstract algebra and cryptography. It was not easy to obtain a project description that could fit his skills with the other members' capabilities. But when the idea of generating a web presentation displaying details of the project, Herman decided that he could contribute on this development. Since Fredrik had some past familiarity and personal enthusiasm with making websites, they cooperated the process throughout the semester.

As the isogeometric analysis approach establishes direct interaction between CAD and the finite element method, Bjørn Harald and Jon Vegard interacted repeatedly to obtain a useful model. Also, on the structural mechanics side, both Magnus and Bjørn Harald worked side by side on theory and applications.

All in all, the academic composition fitted the project description quite well. Some additional competence in computer science may have resulted in a more advanced implementation of the numerics, and that the link between CAD software and the finite element codes could have been easier to establish. Though not of crucial importance, the group, however, also recognizes that presentational skills, such as a creating website, is useful for public recognition of the project, and thus the work of Herman and Fredrik suited the team well.

CONTENTS

Preface	iii
Team structure	iv
1 Introduction	1
1.1 Ski history and the introduction of flex curves	1
1.1.1 The ski flex index	1
1.2 The problem description	3
1.3 Overview of the project	3
2 Isogeometric analysis	5
2.1 NURBS	6
2.2 Linear elasticity	10
2.3 The weak form and Galerkin's method	12
2.4 Assembly	15
2.5 Post-processing	20
2.6 Error analysis	20
3 Materials and their properties	27
3.1 Wood	28
3.1.1 Calculations	30
3.1.2 Test results	32
3.2 Composites	33
3.3 Edges	34
3.4 Base	34
3.5 Protectiv plastic	35
4 CAD and Geometry	37
4.1 Geometry definition	37
4.1.1 Cross section	38
4.1.2 Side view definition	38
4.1.3 Top view definition	40
4.1.4 Sweeping of cross section along guides	40
4.1.5 Simplifications in NX model	40
4.2 MATLAB ski geometry	42

5 The Flex Computer	43
6 Results and discussion	49
7 Web presentation	53
7.1 Setting up a content management system	53
7.2 Design and compatibility across devices	53
7.3 A 3D model via the x3dom framework	54
8 Conclusions and suggested further work	57
A Parameter values for displayed ski	59
B Laboratory tests	61
C Laboratory tests of finished skis	69
References	71

Chapter 1

INTRODUCTION

1.1 SKI HISTORY AND THE INTRODUCTION OF FLEX CURVES

Although the concept of skiing dates long back, commercial production of skis became huge first in the 20th century. From the 1950s and onward, the introduction of material structures with metal, fiberglass or carbon fiber, resulted in more complex products. Today, the construction methods enable fine-tuning of ski characteristics when it comes to quality, user preferences and durability.

As mentioned by the Norwegian ski producer Endre Hals [1], the last 10 years have seen a dramatic change in public awareness of ski shapes. This has forced ski companies to modify their construction facilities. Moreover, with the increased demand for personally adapted products comes the need for a measure of the various qualities.

Important in this discussion is the notion of *ski stiffness*, or perhaps its more common version known as the *ski flexibility*, or *flex* for short. The geometry and choice of materials impact the physical properties of alpine skis, and the complexity of today's models may just increase the stiffness variability. Endre Hals notes that there have been much debate about ski stiffness, but the arguments lack an objective language.

1.1.1 THE SKI FLEX INDEX

Essential in this context is how the flex should be calculated. In [2] longitudinal deflection is measured via placing weights on the binding mounting center of the ski. But, as Hals argues, this simple process does not capture a satisfactory amount of information. As a remedy, he proposed so-called *flex curves* and a new standard called the *ski flex index* (SFI) in his diploma thesis.

Cooperating with the Norwegian sports magazine FRI FLYT, he measured selected freestyle- and freeride skis for the 2008 season. With the middle of a ski as a reference point and origin, typically 8 deflection tests on both sides of the ski were used for calculating displacement. Moreover, each such measurement was separated by 100 mm. He then combined the results into flex curves, which draw the distribution of ski stiffness. At the same time some recalculations have been done to make skis of different lengths comparable. An example of a flex curve can be seen in Figure 1.1 with measurements from Table 1.1.

In order to ease the understanding of the data for the broader public, the ski flex index consists of five numbers corresponding to the back tip, back ski, mid ski, front ski and front tip,

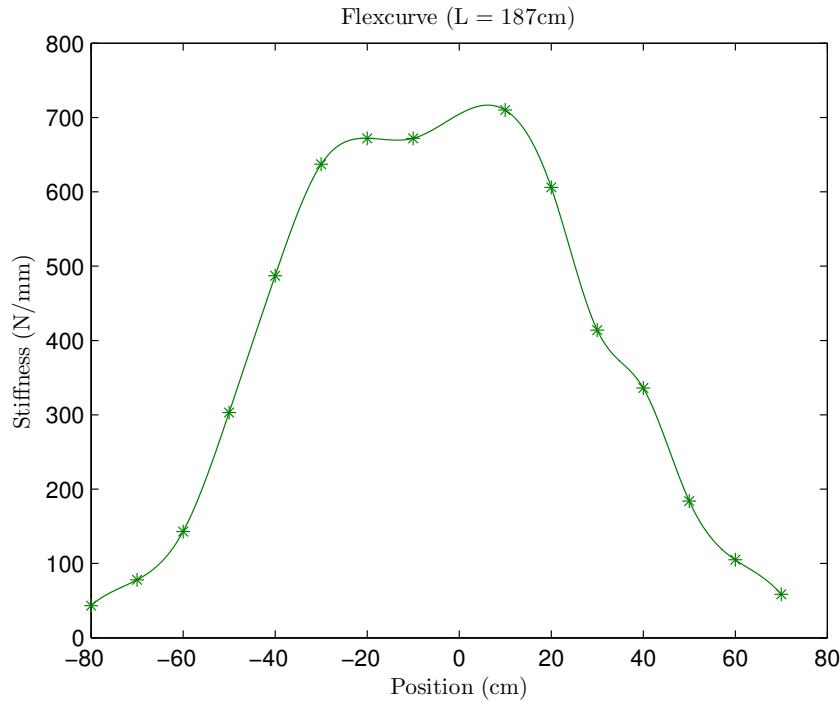


Figure 1.1: Flex curve based on experimental data from Table 1.1 generated by Endre Hals.

respectively. The numbers, ranging from 1 through 9, are based upon a formula via average flex curves. Soft skis get low numbers, while large numbers correspond to stiff models, and a completely average ski gets the SFI 5 5 5 5 5. The ski from Table 1.1 results in the SFI 3 5 6 4 4, which suggest an overall flex index of 4. As Hals comments, it is still open what values may be considered optimal, but they may anyway serve as a better foundation for further discussions.

Table 1.1: Data set from experiments made by Endre Hals. The data from the location 80 cm was not added due to the large displacement of this measurement.

Location (cm)	Displacement (mm)	Stiffness (N/mm)	Flex number	Average
-80	5.74	43.3	3.1	3
-70	3.19	77.9	3.2	
-60	1.74	143	3.2	
-50	0.82	303	4.5	5
-40	0.51	487	5.4	
-30	0.39	637	6.1	
-20	0.37	672	6.0	6
-10	0.37	672	5.9	
10	0.35	710	6.4	
20	0.41	606	5.7	
30	0.6	336	4.5	4
40	0.74	414	4.7	
50	1.35	184	3.8	
60	2.37	105	3.6	4
70	4.25	58.5	3.5	

1.2 THE PROBLEM DESCRIPTION

In this report we have tried to set up a complete model of a customized alpine ski in order to estimate its associated flex curve. Our goal has been to computationally replicate the entire measurement procedure done by Hals, and then compare the estimated flex curve with real data. Another objective was to make some details of the process, methods and results available to the broader public and upcoming students at NTNU via a web presentation.

1.3 OVERVIEW OF THE PROJECT

We do not focus on the SFI, which we believe to be an unstable standard. Since it is based on average flex curves, the ski flex index will actually float during the coming years. The flex curves are, however, not affected by this.

The analyzed ski is designed by Endre Hals, and in close cooperation with him, we have tried to create a 3D CAD model which captures the essential features of the ski. Several extensive laboratory tests have been carried out for the estimation of key material properties.

As a mathematical framework, the so-called isogeometric analysis (IGA) approach has been used. Building on the finite element method (FEM), IGA establishes a direct link between CAD design tools and FEM software which enables exact transfer of the geometry. Our reason for choosing IGA was twofold. First, it is a very fresh and promising technology, being introduced in 2005 by Hughes *et al.* [3]. Second, having acquired some experience with FEM, extending to IGA seemed like an interesting challenge. In addition our working team consisted of members capable of designing CAD models.

The governing physical equations throughout the project are based on linear elasticity, and the underlying assumptions from linear elasticity fit well in this context since the measured displacements are small. Also, the resulting partial differential equations give numerical procedures adequate for the level of the working team.

In chapter 2 we begin outlining details of IGA, along with error analysis of several numerical examples. chapter 3 then discusses the structural components of the ski and results from the laboratory tests. An overview of the design of the 3D model can be found in chapter 4. Here also details of the ski geometry are given.

The numerical replication of the measurement procedure is named «The Flex Computer» and is the content of chapter 5. From then on chapter 6 gives the main results of the project combined with thorough discussions. In chapter 7 details of the construction of the web presentation associated with the project are outlined. Lastly, conclusions and suggestions for further work can be found in chapter 8.

The intended readership is somewhat mixed, and we try to meet several audiences. Mostly, we target advanced undergraduates with interest in mathematical modeling using isogeometric analysis (IGA). However, also a more general public user group such as alpine ski producers

or ski enthusiasts may benefit from our methods. Indeed, the text may be looked upon as a technical report used in the enhancement of ski products.

Much of the content require experience with industrial mechanics and numerical mathematics. In particular, familiarity with the finite element method is of great help for understanding the IGA in chapter 2, and may be the most difficult component to digest in this booklet.

Chapter 2

ISOGEOMETRIC ANALYSIS

Our main mathematical tool to model alpine ski is known as isogeometric analysis (IGA), which is basically an extension of the finite element method (FEM) using non-uniform rational B-splines (NURBS) as basis functions. Being introduced in 2005 by Hughes *et al.* [3], followed by the book [4] in 2009, IGA tries to bridge the gap between finite element analysis and CAD design tools. The important future of IGA is that it uses the same basis as CAD software for describing the given geometry, and thus exact representation of the model is possible. It is therefore natural to include a section considering this basis in the beginning before we set up the IGA. Note that only the relevant parts of the IGA will be presented, and even here we shall be brief. This chapter assumes the reader is somewhat familiar with the finite element method.

The underlying assumption for our analysis is linear elasticity. The resulting equations will be the fundamental equations to which we apply IGA. The notation will be presented and the complete partial differential equation will be presented.

Initially the IGA implementation was entirely based on the library given in [5]. This library have implemented NURBS calculations in C, and have a lot of other artillery which we will not need. We have rewritten all code implemented in C into MATLAB, and many more adjustments have been done, even some optimizations concerning how the sparse global stiffness matrix is built. In this regard it must be noted that the original code written in C is faster than the corresponding rewritten code in MATLAB. It was not necessary to rewrite all parts of the library, so only the parts which are edited will be presented in what follows. For the remaining part of the code, we refer to [5]. When presenting the theory, we shall consecutively present the corresponding implementation in MATLAB. In this regard we present explanations for the most important variables used, and writes the corresponding variable names used in MATLAB, in Table 2.1.

Variable	MATLAB	Description
ξ, η, ζ	xi, eta, zeta	Free variables in the parametric space
$\Xi, \mathcal{H}, \mathcal{Z}$	Xi, Eta, Zeta	Knot vectors in ξ, η and ζ direction respectively.
p, q, r	p, q, r	NURBS order in ξ, η and ζ direction respectively.
n, m, l	n, m, l	Number of control points in ξ, η and ζ direction respectively.
ν, E	nu, E	Poisson's ratio and Youngs Modulus
C, k^e, K	C, k_e, K	Elacitity matrix, element- and global stiffness matrix
$n_{eq} = 3nml$	noDofs	Number of degrees of freedom/equations
n_{en}	n_en	Number of non zero basis functions at (ξ, η, ζ)

Table 2.1: Description of the different variables used in this report and corresponding variable names used in MATLAB.

2.1 NURBS

The NURBS basis is built by B-splines. So an understanding of B-splines is crucial to understanding NURBS. Let p be the polynomial order, let n be the number of functions and let $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ be an ordered vector with non decreasing elements (referred to as a knot vector). Then, the B-spline $N_{i,p}$ are recursively defined by (for $p = 0$)

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

and for $p = 1, 2, 3, \dots$ we have

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

This formula is referred to as Cox-deBoor formula, and the derivative of a spline may be computed by

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

Throughout the project we shall use *open knot vectors*. That is, the first and last element in the vector is repeated $p + 1$ times. Moreover, we shall use normalized knot vectors, which simply spans from 0 to 1. We shall build upon this information to optimize algorithms. When we want to evaluate a B-spline at a fixed ξ it is important to note that it is very redundant to evaluate all n basis function. This is due to the small support of each function. In fact, it turns out that only (at most) $p + 1$ basis functions are non-zero at ξ . It is important to then only use these function to have an efficient code. One typically then implements a function which find the span corresponding to a given ξ . Due to the ordering of the knot vector, we may use a binary search algorithm for finding this span. The span will be defined by the index i corresponding to the first basis function which is non-zero at ξ . The following listing represents the algorithm called `findKnotSpan.m`.

```

if xi == 1
    i = n;
else
    i = floor((n+1)/2) + p;
    while true
        if xi >= Xi(i)
            if xi < Xi(i+1)
                return
            end
            i = floor((n+1-i)/2) + i;
        else
            if xi >= Xi(i-1)
                i = i - 1;
                return
            end
        end
    end

```

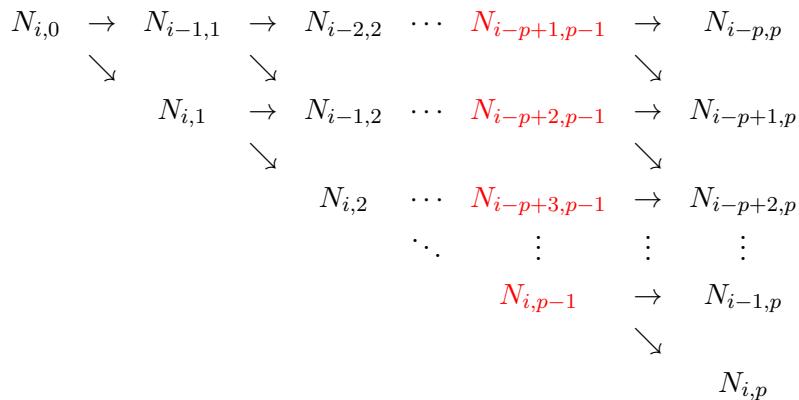
```

    i = floor((i-p)/2) + p;
end
end
end

```

As an example, if $p = 2$ and $n = 8$ and we want to evaluate a B-spline with the knot vector $\Xi = \{0, 0, 0, 0.1, 0.3, 0.5, 0.8, 0.9, 1, 1, 1\}$, we get the index $i = 3$ if $\xi = 0.09$, $i = 6$ if $\xi = 0.5$ and $i = n = 8$ if $\xi = 0.9$ or $\xi = 1.0$.

We are now ready to implement a program which evaluates B-splines using the previous routine. When the recursion formula is used to evaluate the $p + 1$ functions which are non-zero at ξ , the function $N_{i,0}$ is the only function of order zero which is non-zero at ξ . Everything is thus build from this function. We then have the following graph



where the values colored red is used to calculate the nonzero derivatives at the same ξ . It is then clear that we need two loops. The outer loop should iterate over the columns of this graph and the inner loop should iterate over the rows. The output of the function `Bspline_basis` is simply an array `N` which contains the $p + 1$ functions which are evaluated at ξ . To save memory one should also store the intermediate values $N_{i-j+k,j-1}$ in this same array. Thus, we need to store `N(k-1)` to compute `N(k)` (which we store in the variable `saved`). Here, j is the loop index of the outer loop, while k is the loop index of the inner loop. The first iteration (when $p = 0$) should be done separately with (2.1). Finally, we use the convention that $0/0 = 0$.

```

N = zeros(1,p+1);
N(1) = 1;
saved = 1;

for j = 2:p+1
    % For k = 1 there is no dependence on N(k-1) of the previous run.
    for k = 1:j
        % Compute N_{i-j+k, j-1} according to the Cox-deBoor formula
        temp = 0;
        if k ~= j
            temp = (Xi(i+k)-xi)/(Xi(i+k)-Xi(i-j+k+1))*N(k);
        end
        if k ~= 1
            temp = temp + (xi-Xi(i-j+k))/(Xi(i+k-1)-Xi(i-j+k))*saved;
        end
    end
end

```

```

    saved = N(k);
    N(k) = temp;
end
end

```

A way to verify this implementation is to check that the nonnegativity and the partition of unity holds,

$$N_{i,p}(\xi) \geq 0 \quad \text{and} \quad \sum_{i=1}^n N_{i,p}(\xi) = 1 \quad \forall \xi, p,$$

which is important properties of B-splines. We will finally need to compute the corresponding derivatives. Once again, only $p + 1$ derivatives will be non zero at ξ . Let `N_tilde` be an array containing the red values in the previous graph. We may then compute these as in the following listing

```

dNdxi = zeros(1,p+1);
dNdxi(1:p) = p*N_tilde./(Xi(i+1:i+p)-Xi(i-p+1:i));
dNdxi(2:p+1) = dNdxi(2:p+1) - p*N_tilde./(Xi(i+1:i+p)-Xi(i+1-p:i));

```

Since we only need these derivatives also when the corresponding values for the basis function is needed, we extend the function `Bspline_basis` into a new function called `Bspline_basisDers`. We then avoid redundant calculations.

A B-spline volume may now be represented by

$$\mathbf{V}(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) \mathbf{P}_{i,j,k} \quad (2.2)$$

where \mathbf{P} contains the $n \cdot m \cdot l$ controlpoint of the volume. For a given (ξ, η, ζ) , this formula does a lot of redundant computations as discussed so far. Namely, given a (ξ, η, ζ) , only $(p + 1) \cdot (q + 1) \cdot (r + 1)$ basis functions will contribute. If these nonzero basis functions evaluated at a given point (ξ, η, ζ) are stored in \mathbf{M} , \mathbf{N} and \mathbf{L} , we may rewrite (2.2) as

$$\mathbf{V}(\xi, \eta, \zeta) = \sum_{k_1=1}^{p+1} \sum_{k_2=1}^{q+1} \sum_{k_3=1}^{r+1} \underbrace{N_{A_1,p}(\xi)}_{\mathbf{N}(k_1)} \underbrace{M_{A_2,q}(\eta)}_{\mathbf{M}(k_2)} \underbrace{L_{A_3,r}(\zeta)}_{\mathbf{L}(k_3)} \mathbf{P}_{A_1,A_2,A_3}$$

where

$$\begin{aligned} A_1 &= i_1 - p + k_1 - 1 \\ A_2 &= i_2 - q + k_2 - 1 \\ A_3 &= i_3 - r + k_3 - 1 \end{aligned}$$

and i_1 , i_2 and i_3 are knot span index corresponding to ξ , η and ζ , respectively. This observation will be used frequently when dealing with corresponding sums in the NURBS basis.

Given a set of weights w corresponding to each basis function the NURBS basis functions in 3D may be expressed by

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) w_{i,j,k}}{W(\xi, \eta, \zeta)} \quad (2.3)$$

where

$$W(\xi, \eta, \zeta) = \sum_{k_1=1}^{p+1} \sum_{k_2=1}^{q+1} \sum_{k_3=1}^{r+1} \underbrace{N_{A_1,p}(\xi)}_{N(k1)} \underbrace{M_{A_2,q}(\eta)}_{M(k2)} \underbrace{L_{A_3,r}(\zeta)}_{L(k3)} w_{A_1,A_2,A_3}.$$

The partial derivatives of these functions is then given by the quotient rule

$$\frac{\partial R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta)}{\partial \xi} = \frac{W(\xi, \eta, \zeta) N'_{i,p}(\xi) - W_\xi(\xi, \eta, \zeta) N_{i,p}(\xi)}{(W(\xi, \eta, \zeta))^2} M_{j,q}(\eta) L_{k,r}(\zeta) w_{i,j,k} \quad (2.4)$$

where

$$W_\xi(\xi, \eta, \zeta) = \sum_{k_1=1}^{p+1} \sum_{k_2=1}^{q+1} \sum_{k_3=1}^{r+1} \underbrace{N'_{A_1,p}(\xi)}_{dNdx_i(k1)} \underbrace{M_{A_2,q}(\eta)}_{M(k2)} \underbrace{L_{A_3,r}(\zeta)}_{L(k3)} w_{A_1,A_2,A_3}$$

and

$$N'_{i,p}(\xi) = \frac{dN_{i,p}(\xi)}{d\xi}.$$

Similar expressions may be found for the partial derivatives with respect to η and ζ . We may now insert all of this into a routine which computes the nonzero basis functions at a given point (ξ, η, ζ) and the corresponding nonzero derivatives. This function is called `NURBS3DBasisDers`. After finding the nonzero B-spline functions and their corresponding derivatives as discussed above, it continues by first finding the function W and it's corresponding derivatives. This is illustrated in the following listing.

```

for k3 = 1:r+1
    A3 = i3 - r + k3 - 1;
    for k2 = 1:q+1
        A2 = i2 - q + k2 - 1;
        for k1 = 1:p+1
            A1 = i1 - p + k1 - 1;
            A = (m*n)*(A3-1) + n*(A2-1) + A1;

            W = W + N(k1)*M(k2)*L(k3)*weights(A);
            dWdx1 = dWdx1 + dNdx1(k1)*M(k2)*L(k3)*weights(A);
            dWdeta = dWdeta + dMdeta(k2)*N(k1)*L(k3)*weights(A);
            dWdzeta = dWdzeta + dLdzeta(k3)*N(k1)*M(k2)*weights(A);
        end
    end
end

```

Note the use of the index A which is convenient when `weights` is located in an long array (which is the format we shall use when doing the IGA analysis). Finally, the function can now compute the NURBS basis alongside the corresponding derivatives as shown in the following listing.

```

for k3 = 1:r+1
    A3 = i3 - r + k3 - 1;
    for k2 = 1:q+1
        A2 = i2 - q + k2 - 1;
        for k1 = 1:p+1
            A1 = i1 - p + k1 - 1;
            A = (m*n)*(A3-1) + n*(A2-1) + A1;
            a = 9*(k3-1) + 3*(k2-1) + k1;
            fac = weights(A)/W^2;

            NML = N(k1)*M(k2)*L(k3);

            R(a) = NML*weights(A);
            dRdx1(a) = (dNdx1(k1)*M(k2)*L(k3)*W - NML*dWdx1) * fac;
            dRdeta(a) = (dMdeta(k2)*N(k1)*L(k3)*W - NML*dWdeta) * fac;
            dRdzeta(a) = (dLdzeta(k3)*N(k1)*M(k2)*W - NML*dWdzeta) * fac;
        end
    end
end

```

Note the use of the intermediate variables `fac` and `NML` which is used to avoid redundant computations. The index `a` loops over the $n_{en} = (p + 1) \cdot (q + 1) \cdot (r + 1)$ nonzero functions.

2.2 LINEAR ELASTICITY

One important assumption for using linear elasticity is that only small deformations of the material occurs. This is due to the *linearized* quantities. We shall here not go in detail on how to derive the needed equations from *Hooke's* law, but we present the notation used in the formulation of the isogeometric analysis.

The notations used, takes inspirations from [4]. In this section, the indices i, j, k and l will denote a specific spatial direction. We shall model everything in three dimensions, such that $i, j, k, l = 1, 2, 3$. Moreover, u_i shall denote the i^{th} component of the vector \mathbf{u} and differentiation is denoted with a comma such that

$$u_{i,j} = u_{i,x_j} = \frac{\partial u_i}{\partial x_j}.$$

Finally, we use the convention that if an index is repeated, it imply summation. That is,

$$\sigma_{ij}n_j = \sigma_{i,1}n_1 + \sigma_{i,2}n_2 + \sigma_{i,3}n_3$$

and

$$\sigma_{ij,j} + f_i = \frac{\partial \sigma_{i1}}{\partial x_1} + \frac{\partial \sigma_{i2}}{\partial x_2} + \frac{\partial \sigma_{i3}}{\partial x_3} + f_i.$$

Note that we do not sum over i in the latter example since the quantities are separated by a

plus sign. Define now the *symmetric part* of a general tensor $\mathbf{A} = [A_{ij}]$ to be

$$A_{(ij)} = A_{(ji)} := \frac{A_{ij} + A_{ji}}{2}$$

and note that if $\mathbf{B} = [B_{ij}] = [B_{(ij)}]$ is a symmetric tensor, then

$$A_{ij}B_{ij} = A_{(ij)}B_{ij}. \quad (2.5)$$

That is, we can combine the components of \mathbf{B} which are equal to reduce redundant computations.

Let now σ_{ij} denote the Cartesian components of the Cauchy *stress tensor* and let ε_{ij} denote the *infinitesimal strain tensor* which is defined by

$$\varepsilon_{ij} = u_{(i,j)} = \frac{u_{i,j} + u_{j,i}}{2}.$$

We can now state the relation between ε_{ij} as σ_{ij} using the generalized Hooke's law as

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl}$$

where c_{ijkl} are *elastic coefficients*. In the case of isotropic material, these coefficients are given by

$$c_{ijkl} = \frac{\nu E}{(1+\nu)(1-2\nu)}\delta_{ij}\delta_{kl} + \frac{E}{2(1+\nu)}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$$

where the Kronecker delta function is given by

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

and the parameters E and ν are *Young's modulus* and *Poisson's ratio* respectively. We are now ready to state the *strong form* of the linear elasticity problem in three dimensions.

Let $\Omega \subset \mathbb{R}^3$ be the domain with a boundary $\partial\Omega$ which is composed of two parts; Γ_{D_i} and Γ_{N_i} . These are called *Dirichlet* and *Neumann* boundary conditions, respectively, and satisfies $\bigcup_{i=1}^3 \Gamma_{D_i} \cup \Gamma_{N_i} = \partial\Omega$ and $\Gamma_{D_i} \cap \Gamma_{N_i} = \emptyset$. Moreover, let the functions $f_i : \Omega \rightarrow \mathbb{R}$, $g_i : \Gamma_{D_i} \rightarrow \mathbb{R}$ and $h_i : \Gamma_{N_i} \rightarrow \mathbb{R}$ be given. Then, find $u_i : \bar{\Omega} \rightarrow \mathbb{R}$ such that

$$\sigma_{ij,j} + f_i = 0 \quad \text{in } \Omega, \quad (2.6)$$

$$u_i = g_i \quad \text{on } \Gamma_{D_i}, \quad (2.7)$$

$$\sigma_{ij}n_j = h_i \quad \text{on } \Gamma_{N_i}, \quad (2.8)$$

for $i = 1, 2, 3$. In this project, we will only consider boundaries which are entirely Neumann boundaries or Dirichlet boundaries, respectively. That is, $\Gamma_{D_i} = \Gamma_D$ and $\Gamma_{N_i} = \Gamma_N$ for all i .

In this project, we neglect any body forces such as gravity, such that $f_i = 0$. Moreover, we shall only need homogeneous Dirichlet conditions such that $g_i = 0$.

2.3 THE WEAK FORM AND GALERKIN'S METHOD

The *weak form* of the problem is derived from the strong form. Typically, one defines two classes of functions: \mathcal{S}_i denotes the *solution space* and \mathcal{V}_i denotes the *weighting space* for a given spatial component i . These spaces are made in order to handle non homogeneous Dirichlet boundary conditions, but since we have homogeneous Dirichlet boundary conditions, these spaces will be the same. That is, $\mathcal{S}_i = \mathcal{V}_i$ for $i = 1, 2, 3$. Typically, \mathcal{S}_i is a subspace of the Sobolev space $H^1(\Omega)$ (which consist of all functions which have square-integrable derivatives) with a condition such that the Dirichlet boundary condition is satisfied.

We now multiply each of the equations in (2.6) by a corresponding test functions $v_i \in \mathcal{S}_i$ and sum these three equations into one single equation given by

$$v_i \sigma_{ij,j} = -v_i f_i$$

which can be written as

$$v_i \nabla \cdot \bar{\boldsymbol{\sigma}}_i = -v_i f_i$$

where

$$\bar{\boldsymbol{\sigma}}_i = \begin{bmatrix} \sigma_{i1} \\ \sigma_{i2} \\ \sigma_{i3} \end{bmatrix}.$$

Integrating over the domain yields

$$\int_{\Omega} v_i \nabla \cdot \bar{\boldsymbol{\sigma}}_i \, d\Omega = - \int_{\Omega} v_i f_i \, d\Omega. \quad (2.9)$$

Consider the divergence theorem

$$\int_{\Omega} \nabla \cdot \boldsymbol{\Psi} \, d\Omega = \int_{\partial\Omega} \boldsymbol{\Psi} \cdot \mathbf{n} \, d\Omega$$

with $\boldsymbol{\Psi} = v_i \bar{\boldsymbol{\sigma}}_i$. Note that

$$\nabla \cdot \boldsymbol{\Psi} = \nabla \cdot (v_i \bar{\boldsymbol{\sigma}}_i) = \nabla v_i \cdot \bar{\boldsymbol{\sigma}}_i + v_i \nabla \cdot \bar{\boldsymbol{\sigma}}_i$$

which inserted in the divergence theorem yields

$$\int_{\Omega} v_i \nabla \cdot \bar{\boldsymbol{\sigma}}_i \, d\Omega = - \int_{\Omega} \nabla v_i \cdot \bar{\boldsymbol{\sigma}}_i \, d\Omega + \int_{\partial\Omega} v_i \bar{\boldsymbol{\sigma}}_i \cdot \mathbf{n} \, d\Omega.$$

We may now insert this into (2.9) such that

$$\int_{\Omega} \nabla v_i \cdot \bar{\boldsymbol{\sigma}}_i \, d\Omega = \int_{\Gamma_N} v_i \bar{\boldsymbol{\sigma}}_i \cdot \mathbf{n} \, d\Omega + \int_{\Omega} v_i f_i \, d\Omega. \quad (2.10)$$

Note that we only integrate over the Neumann part of the boundary since the integral over the vanishes (the test function v_i is zero at this boundary). Also note that the domain of integration for the boundary integral is depending on i . Returning to the index summation convention we

may rewrite (2.10) as

$$\int_{\Omega} v_{i,j} \sigma_{ij} d\Omega = \int_{\Gamma_N} v_i (\sigma_{ij} n_j) d\Gamma + \int_{\Omega} v_i f_i d\Omega$$

which using the boundary conditions, and the fact that $f_i = 0$ for our case, may be written as

$$\int_{\Omega} v_{(i,j)} \sigma_{ij} d\Omega = \int_{\Gamma_N} v_i h_j d\Gamma.$$

Note that since $[\sigma_{ij}]$ is a symmetric tensor, we have used (2.5) to only write the symmetric part of $v_{i,j}$. If we now define the space $\mathcal{S} = \{\mathbf{u} \mid u_i \in \mathcal{S}_i\}$ we can state the weak formulation in a concise form: Find $\mathbf{u} \in \mathcal{S}$ such that for all $\mathbf{v} \in \mathcal{S}$ we have

$$a(\mathbf{v}, \mathbf{u}) = L(\mathbf{v})$$

where

$$a(\mathbf{v}, \mathbf{u}) = \int_{\Omega} v_{(i,j)} c_{ijkl} u_{(k,l)} d\Omega \quad (2.11)$$

and

$$L(\mathbf{v}) = \int_{\Gamma_N} v_i h_j d\Gamma.$$

Here we have used the relation $\sigma_{ij} = c_{ijkl} \varepsilon_{kl} = c_{ijkl} u_{(k,l)}$.

We now want to transform this weak statement into a system of algebraic equations. We here apply Galerkin's method and now turn to a finite-dimensional subspace $\mathcal{S}^h \subset \mathcal{S}$. The basis for this subspace is the presented NURBS basis. But note that we will have vector valued control variables. The Galerkin approximation of the weak form is now given by: Find $\mathbf{u}^h \in \mathcal{S}^h$ such that

$$a(\mathbf{v}^h, \mathbf{u}^h) = L(\mathbf{v}^h) \quad (2.12)$$

for all $\mathbf{v}^h \in \mathcal{S}^h$.

To find the system of algebraic equations we need to write \mathbf{u}^h as a linear combination of the basis functions. First, let $\boldsymbol{\eta} = \{1, \dots, n_{np}\}$ (where n_{np} is the number of basis functions) be the set containing the indices of all the functions in the NURBS basis defining the geometry and let $\boldsymbol{\eta}_{g_i} \subset \boldsymbol{\eta}$ be the set containing the indices of all the basis functions that are non-zero on Γ_D . Due to the homogeneous Dirichlet boundary condition, we may write the i^{th} component of \mathbf{v}^h and the j^{th} component of \mathbf{u}^h as

$$v_i^h = \sum_{A \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}} R_A c_{iA} \quad \text{and} \quad u_j^h = \sum_{B \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_j}} R_B d_{jB} \quad (2.13)$$

respectively, where $\boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}$ denotes set subtraction. Using the same index summation convention we may now write

$$\mathbf{v}^h = v_i^h \mathbf{e}_i \quad \text{and} \quad \mathbf{u}^h = u_j^h \mathbf{e}_j \quad (2.14)$$

where the unit vectors \mathbf{e}_i are given by

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The finale step is now to insert (2.14) (using (2.13)) into (2.12) such that we obtain a matrix formulation of the problem. Insertion yields

$$a \left(\sum_{A \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_j}} R_A c_{iA} \mathbf{e}_i, \sum_{B \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}} R_B d_{jB} \mathbf{e}_j \right) - L \left(\sum_{A \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}} R_A c_{iA} \mathbf{e}_i \right) = 0$$

which using the bilinearity of a and the linearity of L may be written as

$$\sum_{A \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}} c_{iA} \left(\sum_{B \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}} a(R_A \mathbf{e}_i, R_B \mathbf{e}_j) d_{jB} - L(R_A \mathbf{e}_i) \right) = 0.$$

Since the coefficients c_{iA} is arbitrary (the relation should hold for all $\mathbf{v}^h \in \mathcal{S}^h$) the term in the parentheses must vanish. That is, for all $A \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}$ and $i = 1, 2, 3$ we have

$$\sum_{B \in \boldsymbol{\eta} - \boldsymbol{\eta}_{g_i}} a(R_A \mathbf{e}_i, R_B \mathbf{e}_j) d_{jB} = L(R_A \mathbf{e}_i).$$

One should typically make a system of the ordering of these equations. That is, one should create a function ID which collapse the indices i and A into a single index. A given equation then has the index $P = \text{ID}(i, A)$ and the index over all unknown components of the displacement vectors are called $Q = \text{ID}(j, B)$. The resulting system of equation may then be written as

$$\mathbf{K} \mathbf{U} = \mathbf{F}$$

where

$$\begin{aligned} \mathbf{K} &= [K_{PQ}], \\ \mathbf{U} &= \{d_Q\}, \\ \mathbf{F} &= \{F_P\}, \end{aligned}$$

and

$$\begin{aligned} K_{PQ} &= a(R_A \mathbf{e}_i, R_B \mathbf{e}_j), \\ F_P &= L(R_A \mathbf{e}_j), \\ d_Q &= d_{jB}. \end{aligned}$$

2.4 ASSEMBLY

As for the finite element method, one typically do not loop through the basis functions. Rather, we loop through the elements constructing local stiffness matrices and successfully place their element in the global stiffness matrix. Let us first introduce some notations. The elastic coefficients are typically inserted in a matrix \mathbf{C} called the elasticity matrix. It is defined by

$$\mathbf{C} = \begin{bmatrix} c_{1111} & c_{1122} & c_{1133} & c_{1123} & c_{1113} & c_{1112} \\ & c_{2222} & c_{2233} & c_{2223} & c_{2213} & c_{2212} \\ & & c_{3333} & c_{3323} & c_{3313} & c_{3312} \\ & & & c_{2323} & c_{2323} & c_{2312} \\ & & \text{symmetric} & & c_{1313} & c_{1312} \\ & & & & & c_{1212} \end{bmatrix},$$

or in our case, more explicitly by

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-2\nu)/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-2\nu)/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-2\nu)/2 \end{bmatrix}.$$

Moreover, define the *strain vector* and the *stress vector* to be

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \begin{bmatrix} u_{1,1} \\ u_{2,2} \\ u_{3,3} \\ u_{2,3} + u_{3,2} \\ u_{3,1} + u_{1,3} \\ u_{1,2} + u_{2,1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}.$$

Then

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}(\mathbf{u}),$$

such that we may write the bilinear form in (2.11) as

$$a(\mathbf{v}, \mathbf{u}) = \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{v})^{\top} \mathbf{C} \boldsymbol{\varepsilon}(\mathbf{u}) \, d\Omega.$$

Also note that

$$\boldsymbol{\varepsilon}(R_A \mathbf{e}_i) = \mathbf{B}_A \mathbf{e}_i,$$

where

$$\mathbf{B}_A = \begin{bmatrix} R_{A,1} & 0 & 0 \\ 0 & R_{A,2} & 0 \\ 0 & 0 & R_{A,3} \\ 0 & R_{A,3} & R_{A,2} \\ R_{A,3} & 0 & R_{A,1} \\ R_{A,2} & R_{A,1} & 0 \end{bmatrix}.$$

The entries in the global stiffness matrix may then be written as

$$K_{PQ} = a(R_A \mathbf{e}_i, R_B \mathbf{e}_j) = \mathbf{e}_i^\top \int_\Omega \mathbf{B}_A^\top C \mathbf{B}_B d\Omega \mathbf{e}_j.$$

Let Ω^e be the domain of a given element, where the index e loops over all elements. The support of the NURBS are highly localized. To reduce computations, we should only integrate over functions which have support in Ω^e . If we have n_{en} such *local shape functions*, and let a and b iterate over these functions, we may calculate the entries in the local stiffness matrix as

$$k_{pq}^e = \mathbf{e}_i^\top \int_{\Omega^e} \mathbf{B}_a^\top C \mathbf{B}_b d\Omega \mathbf{e}_j$$

where

$$p = n_{en}(i - 1) + a \quad \text{and} \quad q = n_{en}(j - 1) + b.$$

The local force vector may similarly be calculated by

$$f_p^e = \int_{\Gamma_N^e} R_A h_i d\Gamma. \quad (2.15)$$

The integration is done by quadrature formulas. One first maps to the parametric domain, and then map this domain to a parent domain. The element in the parametric domain, corresponding to Ω^e , is given by

$$\hat{\Omega}^e = [\xi_i, \xi_{i+1}] \times [\eta_i, \eta_{i+1}] \times [\zeta_i, \zeta_{i+1}].$$

In three dimension we want to map this domain into the parent domain given by

$$\tilde{\Omega}^e = [-1, 1] \times [-1, 1] \times [-1, 1].$$

So given $(\tilde{\xi}, \tilde{\eta}, \tilde{\zeta}) \in \tilde{\Omega}^e$, we calculate $(\xi, \eta, \zeta) \in \hat{\Omega}^e$ by

$$\begin{aligned} \xi &= \xi_i + (\tilde{\xi} + 1) \frac{\xi_{i+1} - \xi_i}{2}, \\ \eta &= \eta_i + (\tilde{\eta} + 1) \frac{\eta_{i+1} - \eta_i}{2}, \\ \zeta &= \zeta_i + (\tilde{\zeta} + 1) \frac{\zeta_{i+1} - \zeta_i}{2}. \end{aligned}$$

The Jacobian determinant for the parametric to parent mapping is thus given by

$$J_2 = \begin{vmatrix} \frac{\partial \xi}{\partial \xi} & \frac{\partial \xi}{\partial \eta} & \frac{\partial \xi}{\partial \zeta} \\ \frac{\partial \eta}{\partial \xi} & \frac{\partial \eta}{\partial \eta} & \frac{\partial \eta}{\partial \zeta} \\ \frac{\partial \zeta}{\partial \xi} & \frac{\partial \zeta}{\partial \eta} & \frac{\partial \zeta}{\partial \zeta} \end{vmatrix} = \frac{1}{8} (\xi_{i+1} - \xi_i)(\eta_{i+1} - \eta_i)(\zeta_{i+1} - \zeta_i).$$

Similarly, we need the Jacobian for the mapping from the physical domain into the parametric domain. Given $(\xi, \eta, \zeta) \in \hat{\Omega}^e$, we calculate $(x, y, z) = (x_1, x_2, x_3) \in \Omega^e$ by

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \mathbf{P}_{i,j,k} = \sum_{a=1}^{n_{en}} R_a(\xi, \eta, \zeta) \mathbf{P}_a$$

where $\mathbf{P}_{i,j,k}$ are the control points and $R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta)$ are the NURBS basis functions which are computed by (2.3). Note that the last equality again comes from the highly localized support of the NURBS basis. The Jacobian matrix is thus given by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi} & \frac{\partial x_1}{\partial \eta} & \frac{\partial x_1}{\partial \zeta} \\ \frac{\partial x_2}{\partial \xi} & \frac{\partial x_2}{\partial \eta} & \frac{\partial x_2}{\partial \zeta} \\ \frac{\partial x_3}{\partial \xi} & \frac{\partial x_3}{\partial \eta} & \frac{\partial x_3}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \dots & \mathbf{P}_{n_{en}} \end{bmatrix} \begin{bmatrix} \frac{\partial R_1}{\partial \xi} & \frac{\partial R_1}{\partial \eta} & \frac{\partial R_1}{\partial \zeta} \\ \frac{\partial R_2}{\partial \xi} & \frac{\partial R_2}{\partial \eta} & \frac{\partial R_2}{\partial \zeta} \\ \vdots & \vdots & \vdots \\ \frac{\partial R_{n_{en}}}{\partial \xi} & \frac{\partial R_{n_{en}}}{\partial \eta} & \frac{\partial R_{n_{en}}}{\partial \zeta} \end{bmatrix}$$

such that the Jacobian determinant of this transformation is given by

$$J_1 = \det(\mathbf{J})$$

where the derivatives of the NURBS basis functions are computed by (2.4). The matrix \mathbf{B}_a contains derivatives of the NURBS functions w.r.t. physical coordinates. So we need to find expressions for $\frac{\partial R}{\partial x_i}$. By the chain rule we have

$$\begin{aligned} \frac{\partial R}{\partial \xi} &= \frac{\partial R}{\partial x_1} \frac{\partial x_1}{\partial \xi} + \frac{\partial R}{\partial x_2} \frac{\partial x_2}{\partial \xi} + \frac{\partial R}{\partial x_3} \frac{\partial x_3}{\partial \xi} \\ \frac{\partial R}{\partial \eta} &= \frac{\partial R}{\partial x_1} \frac{\partial x_1}{\partial \eta} + \frac{\partial R}{\partial x_2} \frac{\partial x_2}{\partial \eta} + \frac{\partial R}{\partial x_3} \frac{\partial x_3}{\partial \eta} \\ \frac{\partial R}{\partial \zeta} &= \frac{\partial R}{\partial x_1} \frac{\partial x_1}{\partial \zeta} + \frac{\partial R}{\partial x_2} \frac{\partial x_2}{\partial \zeta} + \frac{\partial R}{\partial x_3} \frac{\partial x_3}{\partial \zeta}. \end{aligned}$$

And thus, we may write

$$\begin{bmatrix} \frac{\partial R}{\partial x_1} & \frac{\partial R}{\partial x_2} & \frac{\partial R}{\partial x_3} \end{bmatrix} \mathbf{J} = \begin{bmatrix} \frac{\partial R}{\partial \xi} & \frac{\partial R}{\partial \eta} & \frac{\partial R}{\partial \zeta} \end{bmatrix}. \quad (2.16)$$

Multiplying with the inverse of the Jacobian, \mathbf{J}^{-1} , from the right and taking the transpose on each side of the equation finally yields

$$\begin{bmatrix} \frac{\partial R}{\partial x_1} \\ \frac{\partial R}{\partial x_2} \\ \frac{\partial R}{\partial x_3} \end{bmatrix} = \mathbf{J}^{-\top} \begin{bmatrix} \frac{\partial R}{\partial \xi} \\ \frac{\partial R}{\partial \eta} \\ \frac{\partial R}{\partial \zeta} \end{bmatrix}.$$

By successfully placing these expressions in the matrix \mathbf{B} , we may finally write

$$k_{pq}^e = \mathbf{e}_i^\top \int_{\tilde{\Omega}^e} \mathbf{B}_a^\top \mathbf{C} \mathbf{B}_b |J_1| |J_2| d\tilde{\Omega} \mathbf{e}_j.$$

By carefully sequentially placing all values for $a, b = 1, \dots, n_{en}$ into the matrix \mathbf{B} we can compute the whole local stiffness matrix in one go by

$$\mathbf{k}^e = \int_{\tilde{\Omega}^e} \mathbf{B}^\top \mathbf{C} \mathbf{B} |J_1| |J_2| d\tilde{\Omega}.$$

The integrals is approximated with quadrature rules. If we want to approximate the integral

$$\int_{\tilde{\Omega}} g(\tilde{\xi}, \tilde{\eta}, \tilde{\zeta}) d\tilde{\Omega}$$

the approximation by Gaussian quadrature is given by

$$\int_{\tilde{\Omega}} g(\tilde{\xi}, \tilde{\eta}, \tilde{\zeta}) d\tilde{\Omega} \approx \sum_{q=1}^{n_q} \rho_q g(\tilde{\xi}_q, \tilde{\eta}_q, \tilde{\zeta}_q),$$

where n_q are the number of integration points, and $(\tilde{\xi}_q, \tilde{\eta}_q, \tilde{\zeta}_q)$ and ρ_q are given quadrature points and weights, respectively.

A somewhat similar procedure as shown above is needed to approximate the surface integral in (2.15). The full expression will depend on which part of the surface we integrate over. As an example, consider integrating over a surface where $\eta = 0$. As for the local stiffness matrix, we can compute the corresponding contribution to the load vector in one go as follows

$$\mathbf{f}^e = \int_{\tilde{\Gamma}_N^e} \left[\mathbf{R}^\top \Big|_{\eta=0} h_1 \quad \mathbf{R}^\top \Big|_{\eta=0} h_2 \quad \mathbf{R}^\top \Big|_{\eta=0} h_3 \right]^\top |J_1| |J_2| d\tilde{\Gamma}$$

where

$$\begin{aligned} \mathbf{R}^\top &= [R_1 \quad R_2 \quad \dots \quad R_{n_{en}}], \\ J_2 &= \frac{1}{4} (\xi_{i+1} - \xi_i)(\zeta_{i+1} - \zeta_i), \\ J_1 &= \det(\mathbf{J}), \end{aligned}$$

and

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi} & \frac{\partial x_1}{\partial \zeta} \\ \frac{\partial x_3}{\partial \xi} & \frac{\partial x_3}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \mathbf{P}'_1 & \mathbf{P}'_2 & \dots & \mathbf{P}'_{n_{en}} \end{bmatrix} \begin{bmatrix} \frac{\partial R_1}{\partial \xi} & \frac{\partial R_1}{\partial \zeta} \\ \frac{\partial R_2}{\partial \xi} & \frac{\partial R_2}{\partial \zeta} \\ \vdots & \vdots \\ \frac{\partial R_{n_{en}}}{\partial \xi} & \frac{\partial R_{n_{en}}}{\partial \zeta} \end{bmatrix}.$$

Here \mathbf{P}'_i contains the x_1 and x_3 component of the control point \mathbf{P}_i . The integral may now be approximated by quadrature rules in two dimensions.

It is very important to note that the global stiffness matrix is very sparse. Not only is it a huge advantage in MATLAB to have the matrix in sparse format when solving the linear system, but also when assembling the matrix. If the matrix is made sparse only after assembly, then the initialization would require MATLAB to allocated place for a full stiffness matrix. This is very memory consuming and should be avoided if one wants to run the program with many degrees of freedom. A matrix in sparse format contains 3 columns; the first two represent the indices in the matrix and the third column represent the corresponding value. A way to do this is to first construct these three columns in three arrays. By noting that each element stiffness matrix has $[3(p+1)(q+1)(r+1)]^2$ number of components, we initialize the arrays by the following.

```
sizeOfk_e = (3*(p+1)*(q+1)*(r+1))^2;
spIdxRow = zeros(1,sizeOfk_e*noElems);
spIdxCol = zeros(1,sizeOfk_e*noElems);
values = zeros(sizeOfk_e*noElems,1);
```

For each element, the element stiffness matrix is then added in the following way.

```
for i = 1:3*nen
    idx = (1:3*nen)+sizeOfk_e*(e-1)+(i-1)*3*nen;
    spIdxRow(idx) = sctrB;
    spIdxCol(idx) = ones(1,3*nen)*sctrB(i);
end
values(sizeOfk_e*(e-1)+1:sizeOfk_e*e) = reshape(k_e, sizeOfk_e, 1);
```

Note that we do not here sum overlapping element matrices, but this is automatically done when the `sparse` function is called in MATLAB.

```
K = sparse(spIdxRow,spIdxCol,values);
```

That is, there will be index combinations which will repeat and therefore this method would require a lot more memory. The other obvious method is simply using

```
K(sctrB,sctrB) = K(sctrB,sctrB) + k_e;
```

which typically results in the MATLAB warning “This sparse indexing expression is likely to be slow”. Experimentally we observe that the other method is 5 times as fast. In [6] it is described how one could optimize the assembly, and in [7] we find theory around efficiently use of quadrature for NURBS. This is however outside the scope of this project, and is therefore not included.

2.5 POST-PROCESSING

The visualizations is done in Paraview. Typically we create so called .vts files directly from MATLAB, and let Paraview do the illustrations from here.

We print the nodes, the displacement and every component of the stress for each node into such files. In addition we calculate the von Mises stress given by

$$\sigma_v = \sqrt{\frac{(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{11} - \sigma_{33})^2 + 6(\sigma_{23}^2 + \sigma_{13}^2 + \sigma_{12}^2)}{2}}$$

which we shall use throughout the report in the visualizations.

One must make a grid in the classical FEM style in order to visualize the result. The mesh is simply created by finding the physical coordinates for each physical element. Moreover, for each corner of an element in the parametric space, the displacement and the components of stress can be calculated. Using the guide in [8], the file writing then follows by looping through each element and write data from each of the 8 corners each time.

2.6 ERROR ANALYSIS

It is always important to present some numerical evidence that the implementation is correct. This is typically done by finding an analytic solution and analyze the convergence of the numerical solution towards this analytic solution. Let the minimal order of the NURBS basis be given by

$$k = \min\{p, q, r\}$$

and let the maximal diameter of the elements (in the parametric space) be noted by $h = h_{max}$. In the error analysis we shall start by the simplest open knot vectors. That is, for $p = 2$ we have $\Xi = \{0, 0, 0, 1, 1, 1\}$, such that $|\Xi| = n + p + 1$. In three dimensions we then have $h = \sqrt{3}$. Generally we have

$$h = \frac{\sqrt{3}}{n - p}$$

where we assume that the polynomial order is equal in all directions and we refine uniformly. The energy norm is defined by

$$\|\mathbf{u} - \mathbf{u}^h\|_E = \sqrt{a(\mathbf{u} - \mathbf{u}^h, \mathbf{u} - \mathbf{u}^h)}$$

where we compute the bilinear form by

$$\begin{aligned} a(\mathbf{u} - \mathbf{u}^h, \mathbf{u} - \mathbf{u}^h) &= \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u} - \mathbf{u}^h)^{\top} \mathbf{C} \boldsymbol{\varepsilon}(\mathbf{u} - \mathbf{u}^h) \, d\Omega \\ &= \int_{\Omega} [\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}(\mathbf{u}^h)]^{\top} \mathbf{C} [\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}(\mathbf{u}^h)] \, d\Omega \\ &= \sum_{e=1}^{n_{el}} \int_{\Omega_e} [\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}(\mathbf{u}^h)]^{\top} \mathbf{C} [\boldsymbol{\varepsilon}(\mathbf{u}) - \boldsymbol{\varepsilon}(\mathbf{u}^h)] \, d\Omega, \end{aligned}$$

using the same technique with transformation to the parent element for integration with quadratures. Thus, we need to compute $\boldsymbol{\varepsilon}(\mathbf{u})$ in order to do the error analysis.

In [9] we find the following estimate

$$\|\mathbf{u} - \mathbf{u}^h\|_E \leq Ch^k$$

where a quasi-uniform mesh refinement has been assumed. The constant C is among other things dependent of the transformation from the parametric space to the physical space. It is not however depending on h . So as h become smaller, we expect the convergence to be of order k . Of course, the analytic solution (and then also the norm) should be independent of h , so we shall divide by $\|\mathbf{u}\|_E$ to normalize the error.

We want to do an error analysis with an analytic solution and a geometry which is to some extent similar to what we shall use when analyzing the flex of skis. Analytic solution (with corresponding data) on complex domains is in general hard to find. We shall restrict our self to a domain given by

$$\Omega = (0, w_x) \times (0, w_y) \times (0, w_z)$$

where w_x , w_y and w_z is the width in x , y and z direction of the beam, respectively. In the following we shall present several analytic solutions with corresponding illustrative plots and convergence plots. In the first two cases we choose these three parameters, such that the domain looks like a plank in a way that $w_x > w_y > w_z$. In all test cases we want to construct solution which have homogeneous Dirichlet boundary conditions at $x = 0$. The boundary conditions corresponding to each solution and the analytic expression for $\boldsymbol{\varepsilon}(\mathbf{u})$ can be found in Maple files listed at

<http://org.ntnu.no/skimodeling/maplefiles>.

One of the simplest example that comes to mind is the solution

$$\mathbf{u} = \begin{bmatrix} 0 \\ x \\ 0 \end{bmatrix} \tag{2.17}$$

which is plotted in Figure 2.1. Of course, such linear solution is already in the search space \mathcal{S}^h , so we should expect the error to be near machine epsilon. And as we can see from the convergence plot given in Figure 2.3a this is indeed that case. Here the error is more likely to originate from the approximation in quadrature rules.

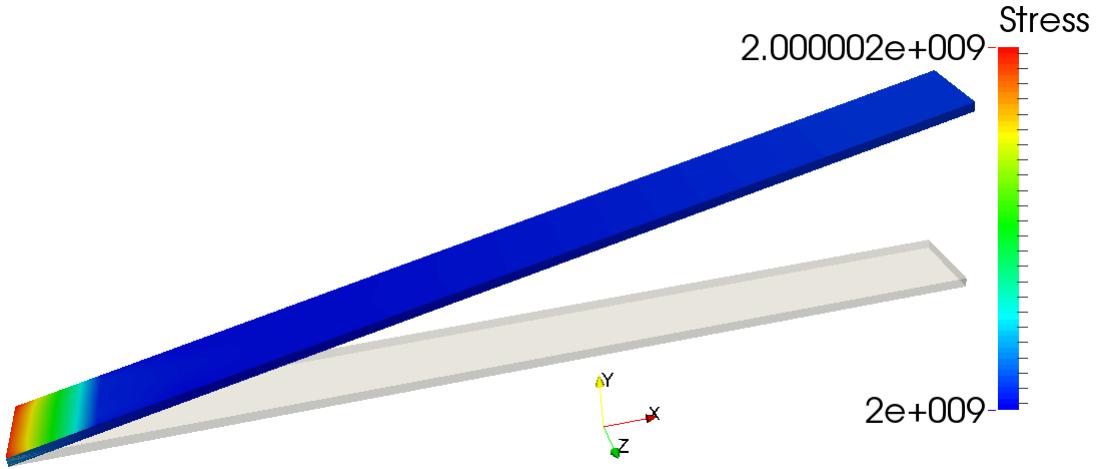


Figure 2.1: Plot of the analytical solution given by (2.17) (a scaling factor 0.2 is used on the displacement). The transparent plank is shown as a reference to the displaced plank which is colored with von Mises stress. Do not be fooled by the different coloring of the von Mises stresses. There is only one nonzero stress component, namely $\sigma_{12} = \frac{1}{2} \frac{E}{1+\nu}$, which is constant throughout the domain. The different coloring comes from machine precision.

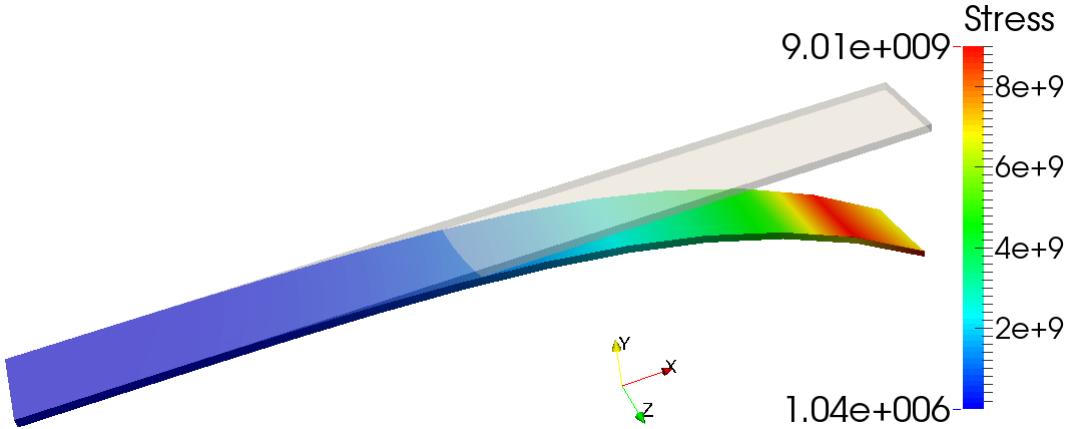


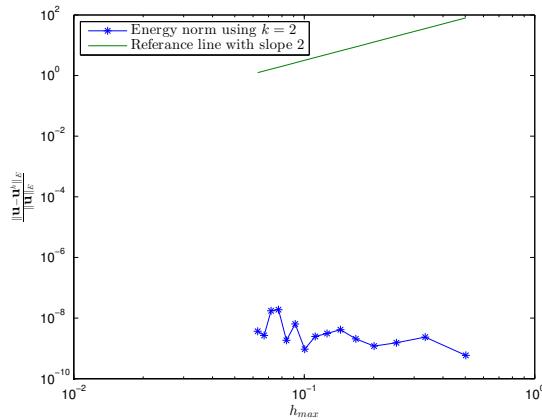
Figure 2.2: Plot of the analytical solution given by (2.18). The transparent beam is shown as a reference to the displaced box which is colored with von Mises stress.

A more complex solution is given by

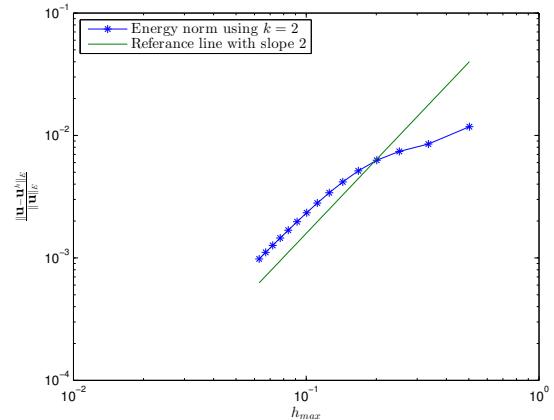
$$\mathbf{u} = \begin{bmatrix} \frac{1}{5}xy(2\nu^2x^2 - 6\nu^2z^2 - 4\nu x^2 + 15\nu z^2 - 6z^2) \\ \frac{1}{10}x^2(\nu x^2 - 3\nu z^2 - 2x^2 + 6z^2) \\ -\frac{3}{5}x^2yz(\nu - 2) \end{bmatrix}. \quad (2.18)$$

The plot of the solution is shown in Figure 2.2. The analysis is done with second order NURBS, so since the analytic solution has a third order polynomial in the x direction, the solution is not in the search space \mathcal{S}^h . The convergence plot is presented in Figure 2.3b. The order of convergence is not impressive in the beginning, but it gets closer to the expected second order of convergence as the element size gets smaller. This is probably due to the constant C , which we shall present evidence for in the next case.

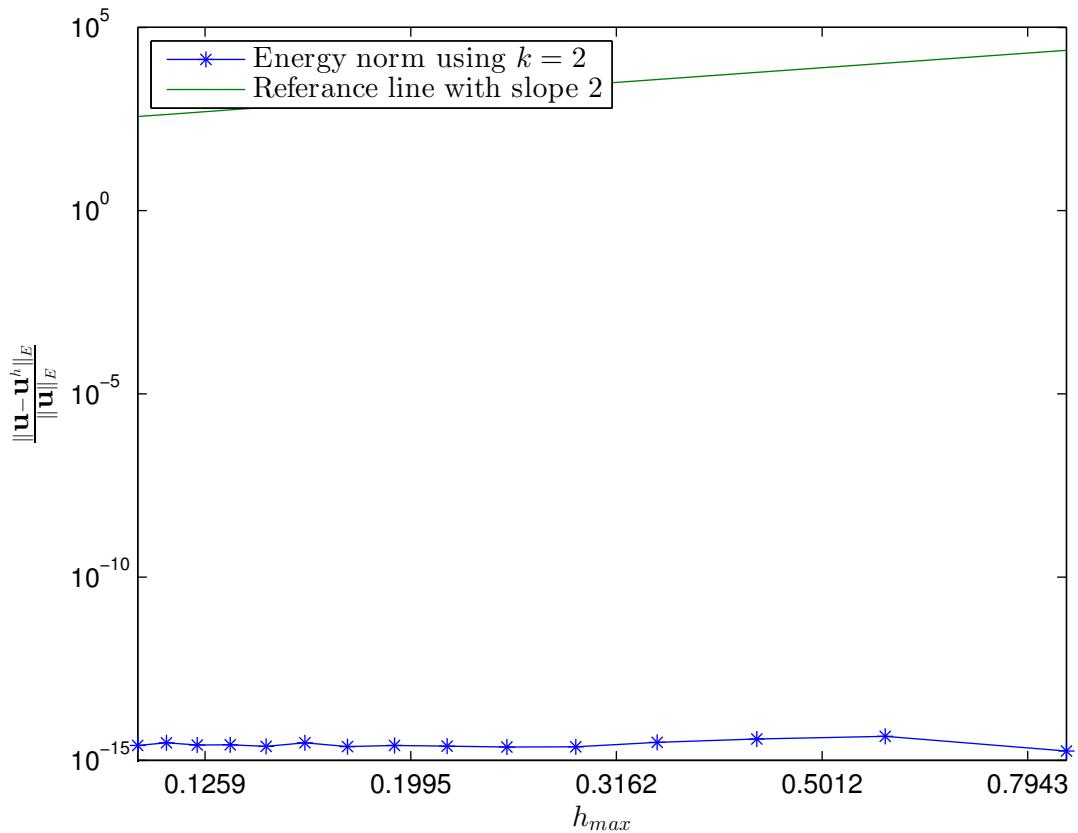
A more regular domain would be to choose $w_x = w_y = w_z = 1$, i.e. a cube. All though we have not presented so much theory on the case $\mathbf{f} \neq \mathbf{0}$ we shall present this case on the remaining

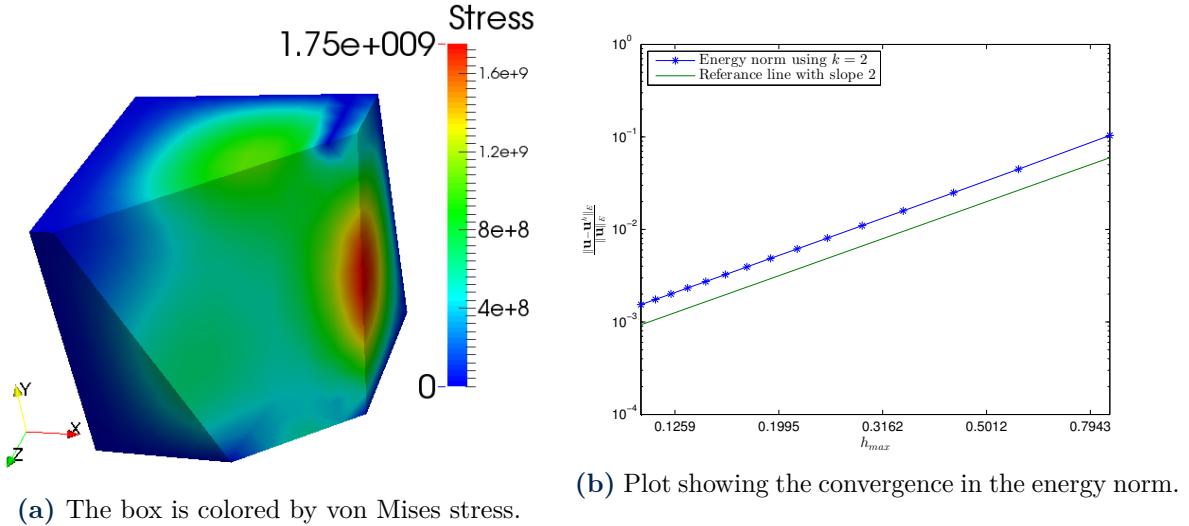


(a) Convergence of the solution in 2.17.



(b) Convergence of the solution in 2.18.

Figure 2.3: Plot illustrating that linear solutions are already in the solution space to the left and a plot showing second order convergence to the right.**Figure 2.4:** Convergence of the solution in 2.19.



(a) The box is colored by von Mises stress.

(b) Plot showing the convergence in the energy norm.

Figure 2.5: Plot of the analytical solution given by (2.20) with corresponding convergence plot of with different NURBS order.

three cases. For simplicity we construct the solution such that they are zero on the boundary of the cube. That is, we only have homogeneous Dirichlet boundary. Consider the solution

$$\mathbf{u} = \begin{bmatrix} x(x - w_x)y(y - w_y)z(z - w_z) \\ x(x - w_x)y(y - w_y)z(z - w_z) \\ x(x - w_x)y(y - w_y)z(z - w_z) \end{bmatrix}. \quad (2.19)$$

It does indeed satisfy the conditions with a rather ugly resulting function \mathbf{f} (the expression for this function is in fact so complex that we do not even bother to include it in the appendix). Once again, since this solution only has degree two for each polynomial in each of the spatial direction, the solution is thus an element in the search space \mathcal{S}^h , so we should expect the error to be close to machine epsilon again. As we can see from Figure 2.4 this is indeed the case.

By multiplying the solution in (2.19) by another factor x ,

$$\mathbf{u} = \begin{bmatrix} x^2(x - w_x)y(y - w_y)z(z - w_z) \\ x^2(x - w_x)y(y - w_y)z(z - w_z) \\ x^2(x - w_x)y(y - w_y)z(z - w_z) \end{bmatrix}, \quad (2.20)$$

we get a solution which should not be in the search space \mathcal{S}^h . The condition on the boundary does of course still hold, but now we have an even more complex function \mathbf{f} . The displacement of the cube is not so illustrative, rather, we cut off a part of the cube such that we can see the von Mises stress at a plane inside the cube as well. This plot is given in Figure 2.5a. The convergence plot is given in Figure 2.5b, and we here observe instantaneously convergence of second order. The finale example is presented just to illustrate that we have the same type of convergence when Neumann boundary condition is implemented. We plot the same analytic function in (2.18), but here we choose $w_x = w_y = w_z = 1$. The resulting plots are given in Figure 2.6 and Figure 2.7.

All of the discussed solutions have been polynomial solution, and would thus potentially be

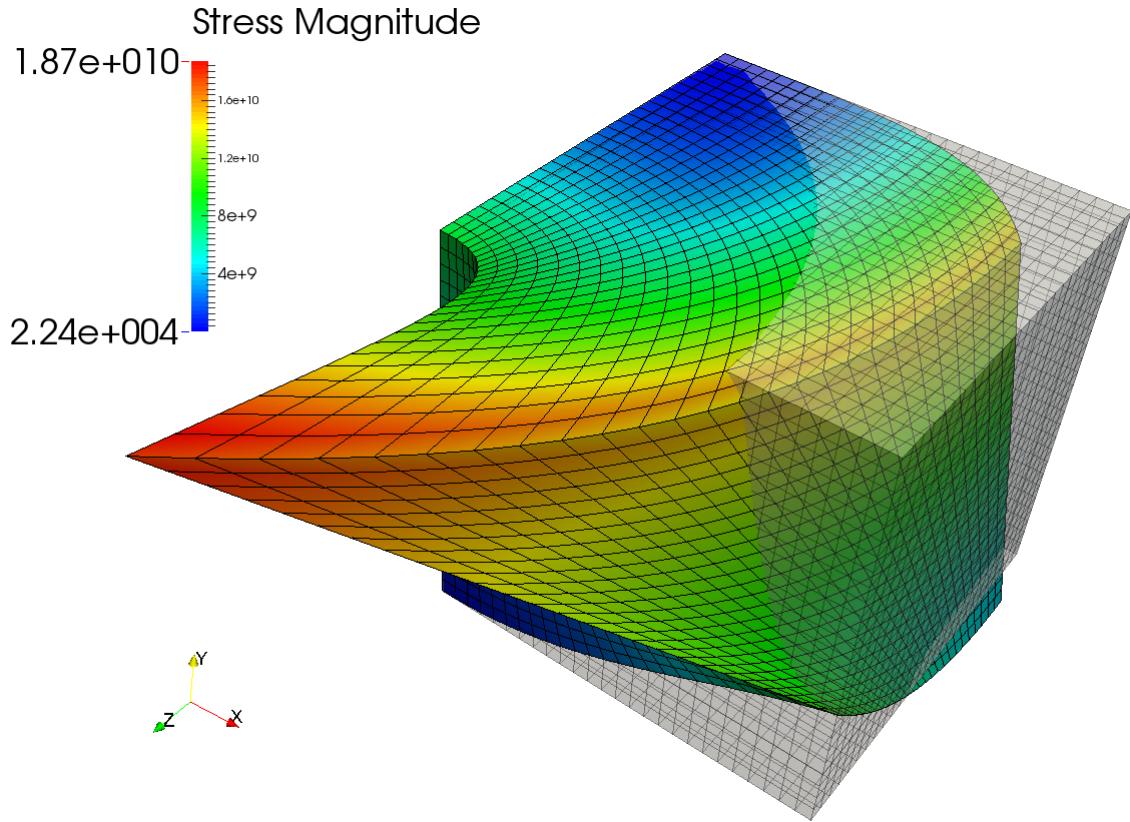


Figure 2.6: Plot of the analytical solution given by (2.20). The transparent box is shown as a reference to the displaced box which is colored with von Mises stress.

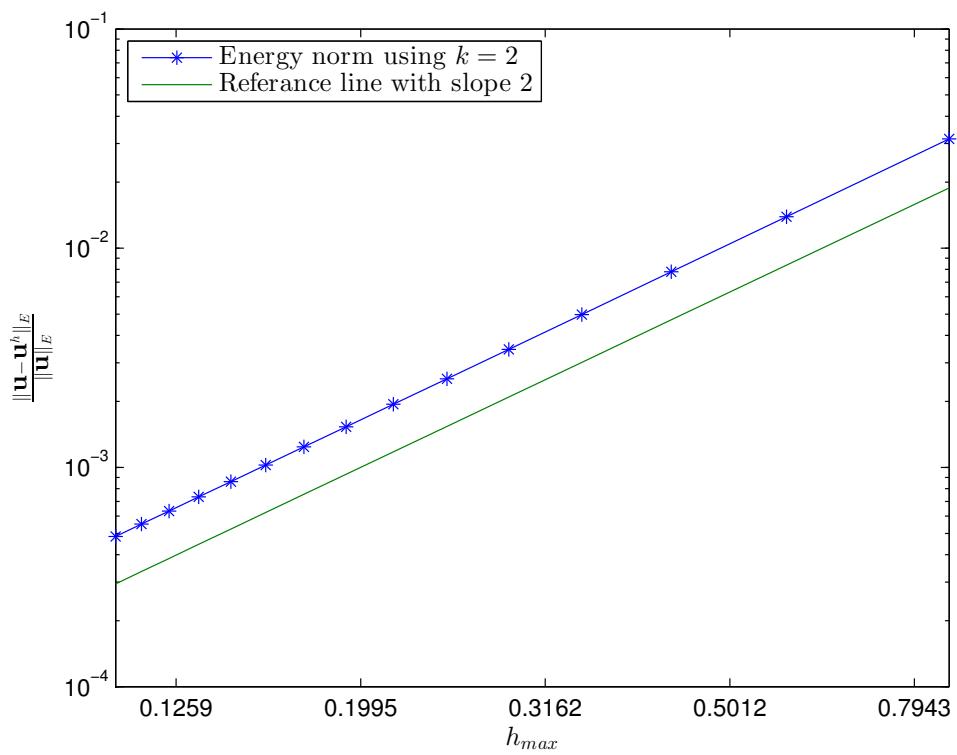


Figure 2.7: Plot showing the convergence in the energy norm.

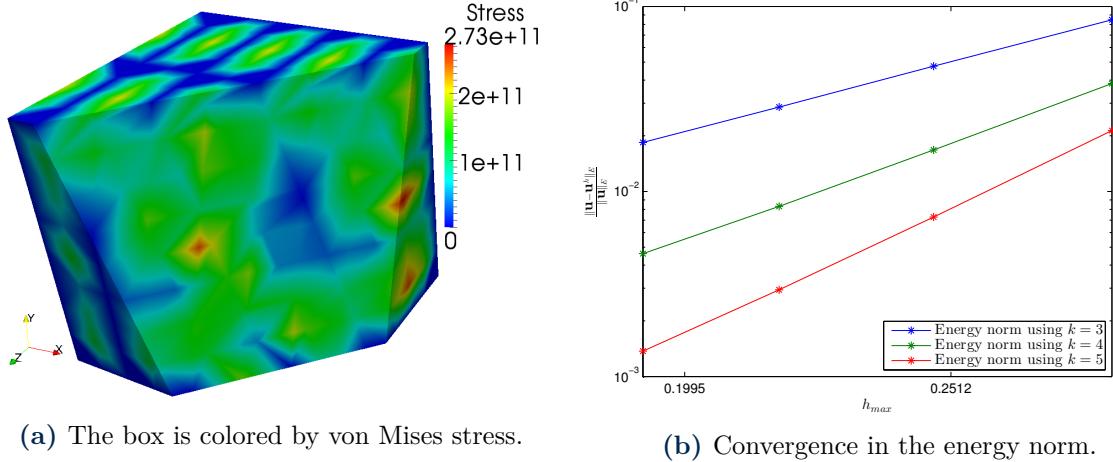


Figure 2.8: Plot of the analytical solution given by (2.21) with corresponding convergence plot of with different NURBS order.

in the solution space \mathcal{S}^h if we elevate the order of the NURBS. Thus, the solutions presented so far does not fit so much for analysis on the order elevation of NURBS. A trigonometric solution given by

$$\mathbf{u} = \begin{bmatrix} \sin\left(\frac{4\pi x}{w_x}\right) \sin\left(\frac{4\pi y}{w_y}\right) \sin\left(\frac{4\pi z}{w_z}\right) \\ \sin\left(\frac{4\pi x}{w_x}\right) \sin\left(\frac{4\pi y}{w_y}\right) \sin\left(\frac{4\pi z}{w_z}\right) \\ \sin\left(\frac{4\pi x}{w_x}\right) \sin\left(\frac{4\pi y}{w_y}\right) \sin\left(\frac{4\pi z}{w_z}\right) \end{bmatrix}, \quad (2.21)$$

will also satisfy homogeneous Dirichlet boundary conditions at the boundary. The resulting plots are given in Figure 2.8a and Figure 2.8b. In the convergence plot, we illustrate the different convergence rates by elevating the NURBS order. It should be noted that the convergence is slightly better than h^k .

Chapter 3

MATERIALS AND THEIR PROPERTIES

The skies we are using are designed and made by Endre Hals, the creator of the flex-test for alpine skis in FRI FLYT. The cross section showed in Figure 3.1 illustrates the different materials from which a ski is made. The following list includes more detailed information:

- Core: The core is made of laminated wood with aspen in the center to save weight and ash at the edges for added stiffness. Endre Hals has cut and glued the core in such a way that the properties and faults in the wood will be symmetrical about centerline along ski.
- Composites: The composites that encase the wooden core are either fiberglass or carbon fiber. This layer enhances the stiffness in the skis, with very little added weight.
- Base: The base is the contact point between the ski and the snow and protects the core from water and damage at the same time as it provides less friction.
- Steel edge: At the edges there will be installed steel bands to increase the carving capabilities and make the skis more durable.
- Protective plastic: This layer is on top of the ski to protect the core and composite from damage. It is also usually decorated in elaborate graphics.

All these materials are glued together with epoxy. The dimensions and composition of these materials defines the properties of a complete ski, and is hence an essential aspect of our project.

The different materials have different properties, both regarding their stiffness and temperature dependencies, but also all the way down to material direction variations. In the following, we will define our approach to each material's properties.

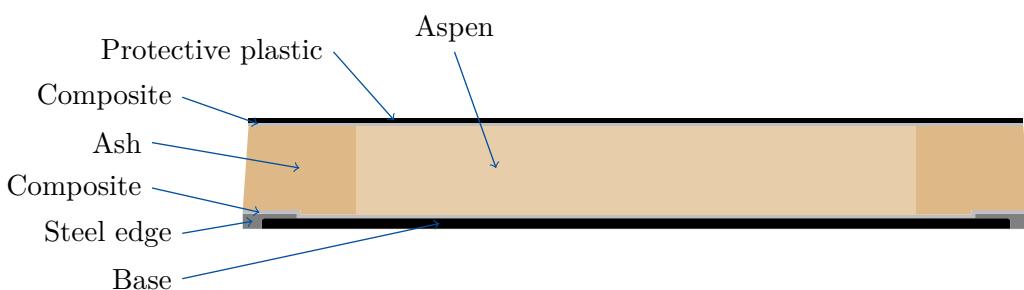


Figure 3.1: Cross section of modeled ski.

3.1 WOOD

Wood is a living material and the properties differ from piece to piece. Its properties are also dependent on the relative humidity. Especially aspen is a seldom used wood type in structures, and data regarding the properties are not easily available or does not exist at all. Another complicating element is the fact that wood is not isotropic, but cylindrically orthotropic. This means that the material has different properties in axial, radial and tangential direction. To get the mechanical properties right, we have conducted a set of tests of our own design. The test procedures needed to establish the full elasticity matrix for any wooden material is both far too complex and too expensive to conduct within the limits of this project, but we solved this problem in another manner. Since our mathematical code only need to be able to represent two specific bending cases, we assumed that if we tested the material under the same conditions as the data extracted would be used to calculate, we could use the measured material stiffness from these test, and thence assume isotropy. The idea was discussed with professor Odd Sture Hopperstad, and together we concluded that this approach would provide good accuracy in the properties.

Hals contributed with two examples of wooden cores, on which we could perform our tests. Hals tests his skis in a cantilever bending test and a cantilever torsion test. Our replication of these tests is described below. Test 4 extends beyond what we need to establish the elasticity matrix, but will give necessary information about the stiffness of the complete ski, so that we can compare the final results from our model with a proper set of laboratory data.

TEST 1

The wooden cores were fixed using two clamps about 15 cm apart. The load was applied using an aluminum plate to evenly distribute the load across the width of the cross section. In the plate we had drilled two holes so that we could hang a platform from, on which we placed weights, see Figure 3.2. To even out and get rid of possible mistakes we applied five different loads three times at three different places on both the front end and back end of the two cores. In every one of the different load cases the deflection was measured with a linear variable differential transformer (LVDT) 100 mm from the clamped end.

TEST 2

In the second test we wanted to find the shear modulus of the cores. To do that we used the same platform as in Test 1, but now we applied the load to the edge of the core to create a torsion moment, and then measured the deflection on each side of the core using two LVDTs, see Figure 3.3 and Figure 3.4. The difference in deflection was used to calculate the angle. The angle can then be used to calculate the shear-modulus.



Figure 3.2: Test setup for the deflection of the wooden core

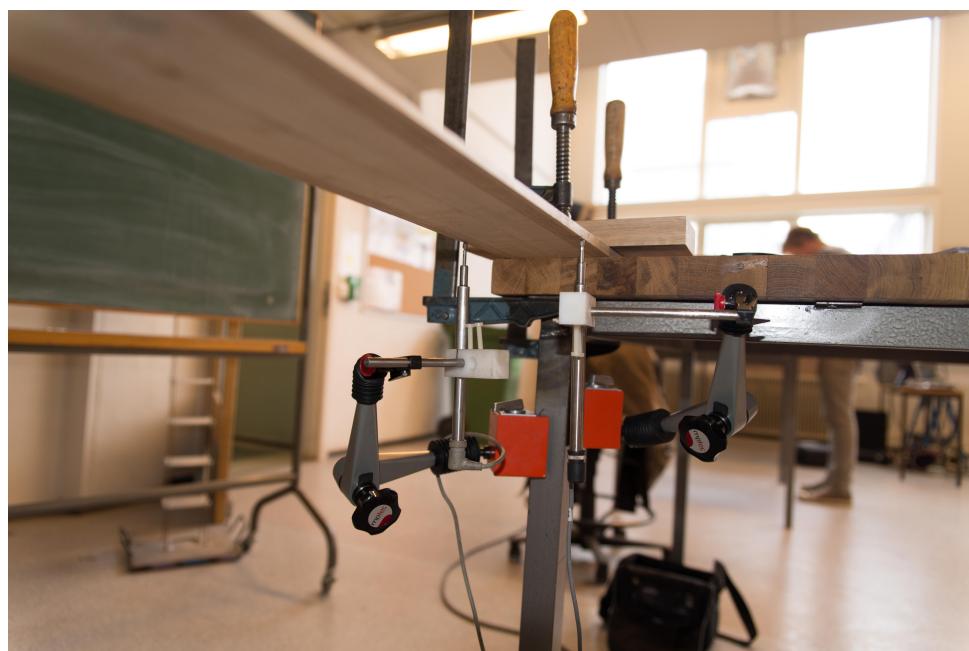


Figure 3.3: Test setup for the rotation of the wooden core

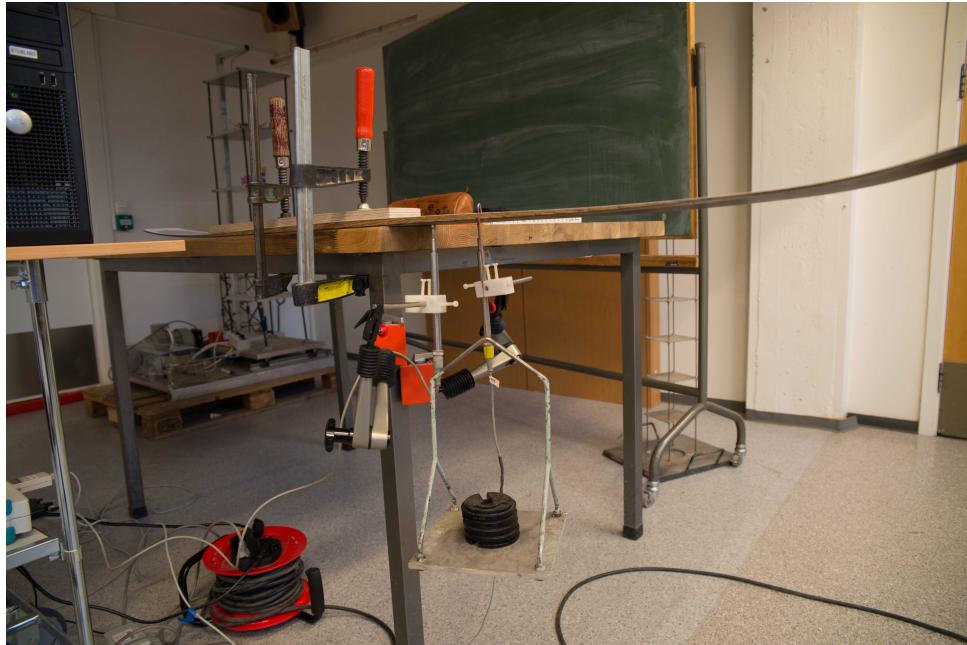


Figure 3.4: Rotation test of completed ski

TEST 3

After both cores were tested, the cores were cut pieces about 40x2 cm so that we could use the same procedures as in test 1 and 2, with a small exception; the pieces were too narrow to use two LVDTs to measure the angle. Instead we used an electronic level, which is not nearly as accurate as the LVDTs. These tests will then give us accurate E-modulus, but will make for an uncertainty around the shear modulus.

TEST 4

Finally we performed both bending and torsion tests on a complete ski (Figure 3.4). The set up for these two tests is described in subsection Test 1 and Test 2.

3.1.1 CALCULATIONS

Given our simple test set up, we are able to find an analytical expression relating the material stiffness and the measured deflections and rotations. The method used to find these expressions, *the unit load method*, are based on beam theory and the principle of virtual work. Since the tests involve small loadings, and consequently small displacements, we can assume that all outer work is conserved as elastic strain energy. By this, we will assume that the displacement produced by two loadings are independent of the loading sequence. We will illustrate the derivation of our expressions with basis in the cantilever bending case. Superposition, combined with the assumption of sequence independence yields that we can combine two load cases as shown in Figure 3.5. This represents a virtual unit load combined with the known actual loading, producing an unknown virtual displacement combined with the known actual displacement.

Note that all variables noted with a tilde are virtual quantities.

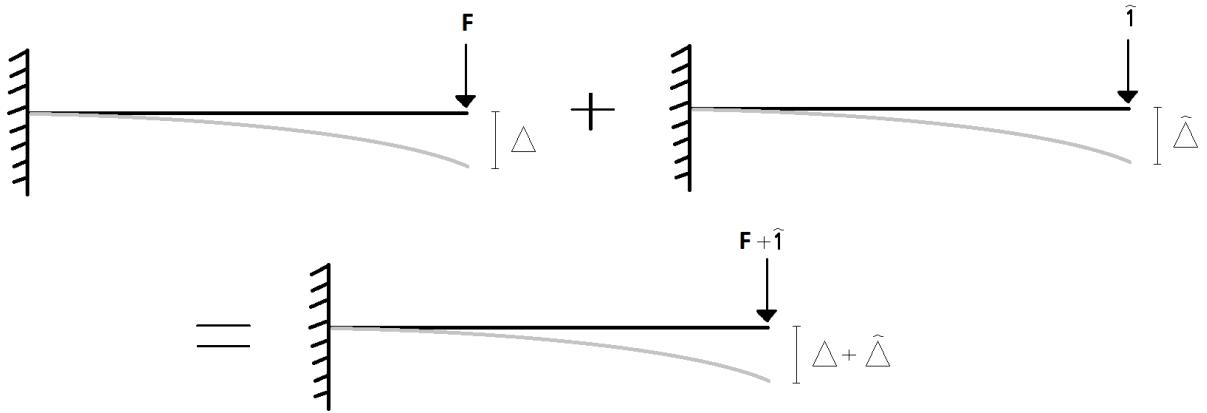


Figure 3.5: Superposition of load cases

By the conservation of work we obtain that the internal and external work must be of equal size; $W_e = W_i$. This can further be expressed as

$$\int_0^{\Delta+\tilde{\Delta}} (F + \tilde{I}) dy = \int_V \int_0^{\varepsilon+\tilde{\varepsilon}} (\sigma + \tilde{\sigma}) d\varepsilon dV,$$

where we evaluate the integrals, and get

$$F\Delta + F\tilde{\Delta} + \tilde{I}\Delta + \tilde{I}\tilde{\Delta} = \int_V \sigma\varepsilon + \sigma\tilde{\varepsilon} + \tilde{\sigma}\varepsilon + \tilde{\sigma}\tilde{\varepsilon} dV.$$

Since $W_e = W_i$ for all subcases, we can split the equation into four independent equations, and keep the one containing the wanted information, namely the one combining real displacement with our virtual unit load

$$\tilde{I}\Delta = \Delta = \int_V \tilde{\sigma}\varepsilon dV.$$

From Hook's law we find the expression for the strain ε

$$\varepsilon = \frac{\sigma}{E},$$

where the normal stress σ is given by

$$\sigma = \frac{M}{I_z}y.$$

By this, we reach the expression for displacement

$$\Delta = \int_V \left(\frac{\tilde{M}}{I_z}y \right) \left(\frac{M}{EI_z}y \right) dV,$$

where the moments, the modulus of elasticity and the second moment of inertia all are dependent of the length coordinate only. This can be rewritten as

$$\int_V \frac{\tilde{M}M}{EI_z^2} y^2 dV = \int_L \frac{\tilde{M}M}{EI_z^2} \int_A y^2 dA dx,$$

where

$$I_z = \int_A y^2 dA.$$

Finally, we reach our explicit expression for the displacement

$$\Delta = \int_0^L \frac{\tilde{M}M}{EI_z} dx.$$

By further manipulations, we have our expression for the elasticity modulus

$$E = \int_0^L \frac{\tilde{M}M}{\Delta I_z} dx.$$

After going through the same procedure using a virtual unit *rotation*, we reach the corresponding expression for the shear modulus

$$G = \int_0^L \frac{\tilde{T}T}{\theta J} dx,$$

where J denotes the polar second moment of area.

Since both the height and the width of the cross section varies along the length, these variations will have to be expressed. The variations over our relatively short distance of interest are small, and is approximated as linear

$$h(x) = h_0 \cdot \left(1 + \frac{x \cdot (h_1 - h_0)}{L}\right) \quad (3.1)$$

$$b(x) = b_0 \cdot \left(1 + \frac{x \cdot (b_1 - b_0)}{L}\right). \quad (3.2)$$

To calculate the shear modulus we need the polar moment of inertia, J . For a rectangle, like our cross section, we use $J = k \cdot (b)^3 \cdot (h)$, where we calculate k by using the following formula [11]

$$k = \frac{1}{3} \left[1 - \frac{192}{\pi^5} \frac{b}{h} \cdot \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n^5} \tanh\left(\frac{n\pi h}{2b}\right) \right]. \quad (3.3)$$

With an expression for both elasticity and shear modulus we wrote a script in MATLAB to solve the integrals for the 947 measurements that were done (Appendix B, Appendix C).

3.1.2 TEST RESULTS

Based on discussions with the laboratory personnel, we have found our final values for the wanted quantities by the mean value of the mean values from each separate section tested. The results are listed in Table 3.1. We also did some tests of the finished skis and got a stiffness of $102.16 N/mm$, $30cm$ behind setback as it would give us most reliable results.

As we can see from our test results, most of the shear-stiffness comes from the ash. As both the E-modulus and the shear-modulus resemble the tabled versions of the properties [12], our test results seems credible. As we see from Table 3.1 each wood type have lower stiffness than the core as a whole. The reason for this could be that the glue that was used to fuse the core

Table 3.1: E- and shear-modulus of wood

	E-modulus [MPa]	Shear-modulus [MPa]
Wood core	11 603	950
Ash	11 046	857
Aspen	11 030	402

together was cut out in the process of making the pieces. Another source of error could be the measuring process, as the tests are extremely sensitive.

An effect of the orthotropic properties of wood which was not foreseen, was that the isotropic relation between Poisson's-ratio, E-modulus and shear-modulus no longer holds. After a short consultation with professor Malo we decided to use the E- and shear-modulus as found, and use $\nu = 0.05$ as the Poisson's-ratio in the elasticity matrix shown in Equation 3.4.

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \quad (3.4)$$

3.2 COMPOSITES

The two types of composites we will be using are fiberglass and carbonfiber. Neither materials are isotropic, but anisotropic. Hals uses both materials in the production of his skies. Carbonfiber is a newer and more expensive material, but as shown in Table 3.2 it is both lighter and stronger [13, 15].

In the skis tested in Test 4, Hals used carbon fiber. After som research we found this elasticity matrix [14] (with every value in GPa), when the fibers are laid with 0° angle to the lenght axis

$$\mathbf{C} = \begin{bmatrix} 246 & 180 & 180 & 0 & 0 & 0 \\ 11.69 & 22.30 & 16.66 & 0 & 0 & 0 \\ 11.69 & 16.66 & 22.30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.82 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.1 \end{bmatrix},$$

and after we had used the transformation matrices we have this for sheets laid at 45° angle to the lenght axis

$$\mathbf{C} = \begin{bmatrix} 122.10 & 107.90 & 98.33 & 0 & 0 & 27.70 \\ 107.89 & 122.10 & 98.33 & 0 & 0 & 27.70 \\ 14.17 & 14.17 & 22.30 & 0 & 0 & -4.97 \\ 0 & 0 & 0 & 4.96 & 2.14 & 0 \\ 0 & 0 & 0 & 2.14 & 4.96 & 0 \\ 98 & 98 & 81.67 & 0 & 0 & 38.30 \end{bmatrix}.$$

At a time these matrices were found, it was no time to include them in the IGA. The values that were used is found in Table 3.2, and is based on a much simpler approach, namely that the composites are isotropic.

Table 3.2: Material properties of carbon fiber and fiberglass

	Carbon fiber	Fiberglass
E-modulus	242 GPa	72 GPa
ν	0.3	0.22
Specific gravity	1.81 g/(cm) ³	2.52 g/(cm) ³

3.3 EDGES

The placement of the steel edges leads to large stresses when the skies are exposed to bending. The stiffness of the edges will lead to less deformation of the skies and therefore less stress in the other components. If the edges had been a softer material it would have led to greater deformations and larger stresses in the other materials.

Table 3.3: Material properties of steel edges

	Steel
E-modulus	210 GPa
ν	0.3

3.4 BASE

The material used in the base is Ultra High Molecular Weight Polyethylene (UHMWPE) that has very good friction properties, but mostly add weight, and will have very little impact on the stiffness of the ski [16]. UHMWPE is not an isotropic material, but we decided to use the isotropic material property as an approximation since the degree at which this affect the analysis is so small (due to low material stiffness). We then use the following properties.

Table 3.4: Material properties of UHMWPE

UHMWPE	
E-modulus	0.72 GPa
ν	0.46
Specific gravity	0.94 g/(cm) ³

3.5 PROTECTIV PLASTIC

Hals uses a polyester-based material as the top layer. The producer of this material do not want to share the mechanical properties, so we assumed the data in Table 3.5 which we found in [17]. As the base, this approximation will have very little impact to the result due to the low material stiffness. Also here isotropic material property is assumed.

Table 3.5: Material properties of polyester

Polyester	
E-modulus	3.65 GPa
ν	0.48

Chapter 4

CAD AND GEOMETRY

Through the course of this project several methods for generating the digital representation of the geometry have been used. The main reason for this is that the link between CAD and IGA has been far harder to establish than what was initially assumed. Our first plan was to use the output from a common CAD tool, Siemens NX8,5. This was put to a halt because it simply proved to be too difficult. However, it was put back on track when we reached out to Kjetil Andre Johannessen, who decided to help us. The main thing he found out, was that he could use the output from NX to generate the model volume as a whole only, not including its internal subsections. From this point, the role of our 3D model generated in NX sank from being generic input material for the analysis, to being nothing but a visual effect. Toward the ending of the project, Kvamsdal engaged SINTEF employee Arne Morten Kvarving to help us. Kvarving set out to make what would essentially be a specialized CAD program, which were to produce an output format tailor-made for our analysis. The idea of the program was that it should be generic, which means that it should be able to represent any ski of the topology presented below, by running a set of defining dimensions. This program is a work in progress, and will not be complete within the time limits of this project. This leads us to our last alternative, to model a much simpler representation of the ski in MATLAB. This ski will be a poor representation of the real thing, but will introduce all the different difficulties we can expect from an exact model. This way, we can still test our mathematical programs on relevant geometry, even if it may not produce useful results.

Even if the geometry used in our tests is not that close to reality, the computer programs written have been designed to handle all variants of skis within the topology of Hals' skis. Further, the program Kvarving is working on will produce skis of the same topology. Hence, for the sake of continued work, the geometry of the skis will be defined.

4.1 GEOMETRY DEFINITION

The geometry will be defined in four steps, found in the four following sections. First, we define the cross section. Then, we define the guides to sweep the cross section along. One guide is the 2D side view, the other is the 2D top view of the ski. Finally, the cross section is swept along the guides to span the full volume of the model. Note that the parameters given in the figures below represent all the needed data to fully define a unique ski, when combined with the constraints listed in each section. Thus, these parameters will form the input for Kvarvings program.

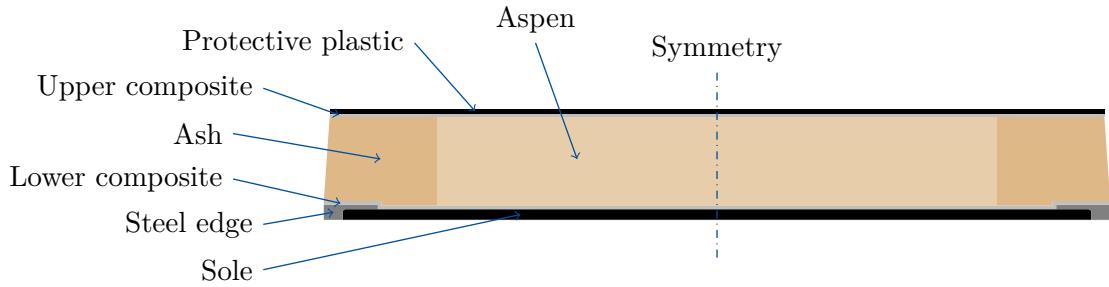


Figure 4.1: Cross section of simplified geometry of ski.

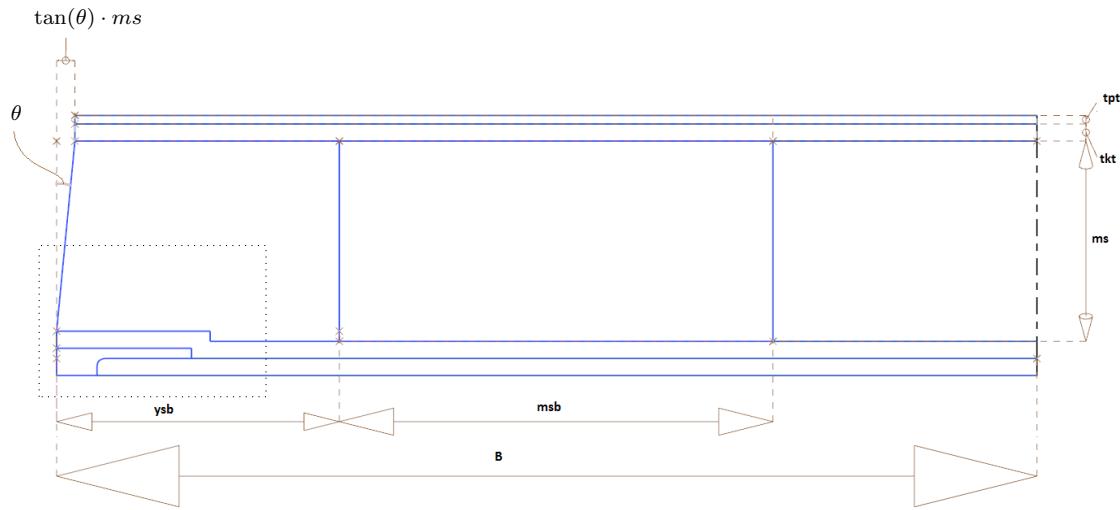


Figure 4.2: Cross section geometry

4.1.1 CROSS SECTION

As the previous chapter suggests, an arbitrary cross section of a ski consists of several different materials, as can be seen in Figure 4.1. These materials divide the cross section into many subsections, and will have to be defined geometrically. Figure 4.2 show the constraining dimensions. Note that the highly detailed corner indicated with a dotted square is dimensioned in Figure 4.3.

4.1.2 SIDE VIEW DEFINITION

The side view of the ski is defined in two steps. First, the base line, see Figure 4.4. All figures displays the ski with its front tip to the right. All transitions are constrained to preserve a continuous tangent. The next step is to define the thickness profile. The thickness is measured normal to the base line at the points indicated in Figure 4.5. Finally, the top line is interpolated through all the thickness lines, this interpolation also has to preserve continuous tangents all along the ski length, and has to be normal to the thickness at both ends.

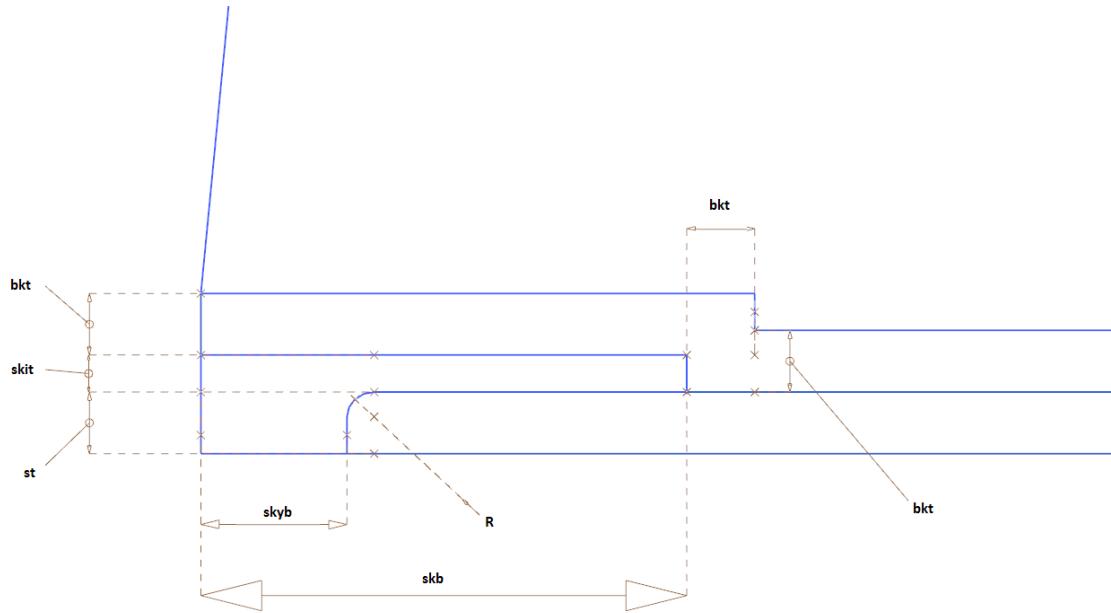


Figure 4.3: Closer view around the steel edge

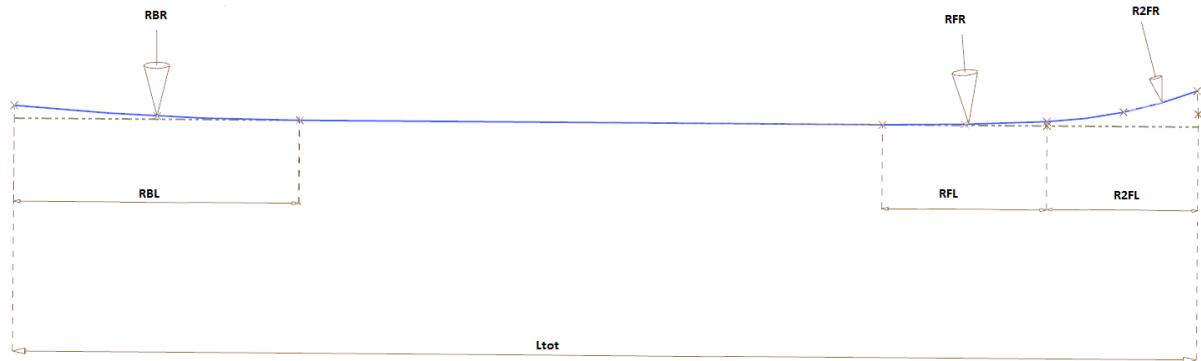


Figure 4.4: Base line

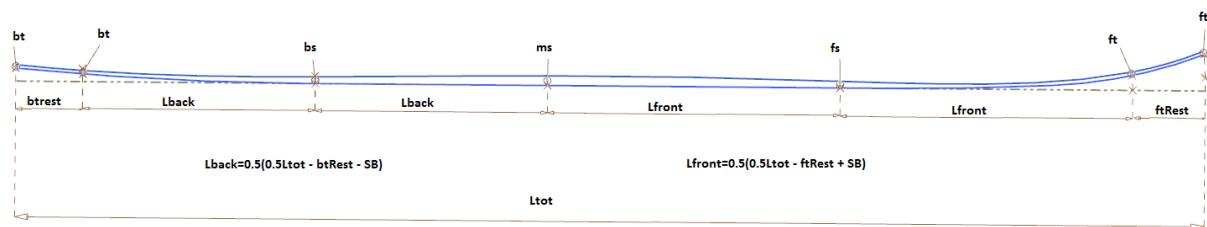


Figure 4.5: Thickness profile

4.1.3 TOP VIEW DEFINITION

Being symmetric about its length axis, we only need to define one half of the ski to fully constrict the top view geometry. As before, all changes in curvature happens with continuous tangents. The dimensions in Figure 4.6 gives most constraints, except one thing. The sharp finishing curves in both ends are given by a tangent transition from the curve with given length and radius, which ends normal to the symmetry line. Note also that the width defined is the narrowest point on the ski, not counting the endings.

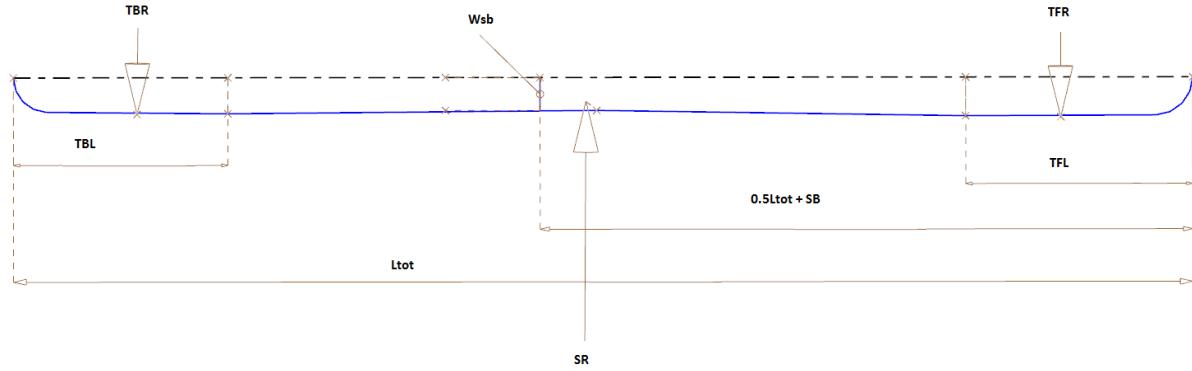


Figure 4.6: Top view geometry

4.1.4 SWEEPING OF CROSS SECTION ALONG GUIDES

Now that the 2D geometry is defined in two different views, these can be combined to produce a 3D shape constriction, or guides, along which the cross section is swept. The two 2D views are combined by extruding each one in the direction normal to their drawing planes. The intersection line between these will then be our 3D guide line. Figure 4.7 shows the guides with the cross section in some different positions. Note that the figure only show half of the width, symmetry defines the other half.

Note also, that only the wooden core changes height, and that both the steel edges, the ash section and the outer aspen section remains of constant width as the total width changes. The resulting geometry can be seen in Figure 4.8, the parameter values defining this ski can be found in Appendix A.

4.1.5 SIMPLIFICATIONS IN NX MODEL

A couple of details have been left out from the cross section. First of all, a layer of thin plastic foil separating the lower composite from the steel edges has been neglected. This is assumed to have an imperceptible effect on the cross sectional bending and torsion stiffness. Likewise, a slight curvature of the top surface is neglected, as it is only introduced in the physical ski to make sure that no air is trapped when compressing the composites onto the wooden core. Further, in the longitudinal direction, all length dimensions are measured straight horizontally,

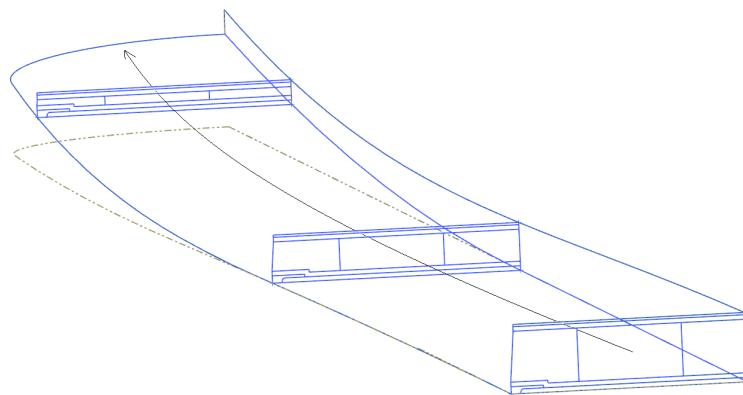


Figure 4.7: Sweeping cross section



Figure 4.8: Resulting side view



Figure 4.9: Resulting top view

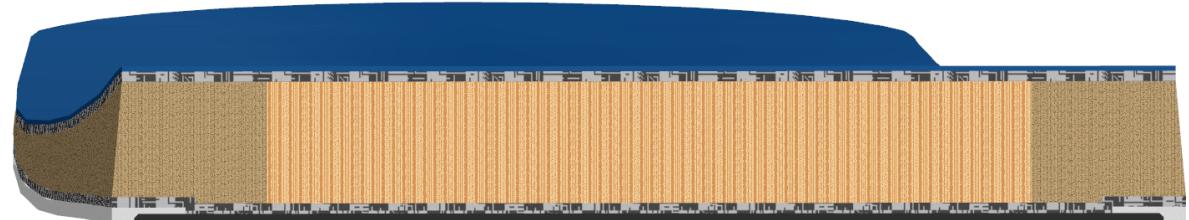


Figure 4.10: Resulting cross section with different materials.

rather than along the curvature of the ski. As an example, the ski defined by the values in Appendix A, is 1.8766 meters long measured along its curvature, compared to 1.8700 measured straight horizontally.

Our boldest assumptions are related to the composites, since these goes through some heating and compression processes after applied to the wooden core. Hence, the final measures will vary from one ski to the next, depending on how much of the mixture is impregnated into the core, and how precisely the processes is conducted. After discussions with Hals, we agreed to assume a final thickness corresponding to 75% of unprocessed thickness. Last, we will stress the fact that the skis of interest are handmade, and will vary as such.

4.2 MATLAB SKI GEOMETRY

The model created in MATLAB is created essentially by trial and error. One could first start with a simple plank and then move around the control points, in order to have a ski which (to some extent) resembles the ski made in CAD. The part of the control points defining the thickness of the ski is made symmetric about the origin (setback) and varies as $e^{-2(x \cdot \frac{2}{L})^2}$. All other control points are chosen by trial and error in order to have a ski that to some extent represents the the reality.

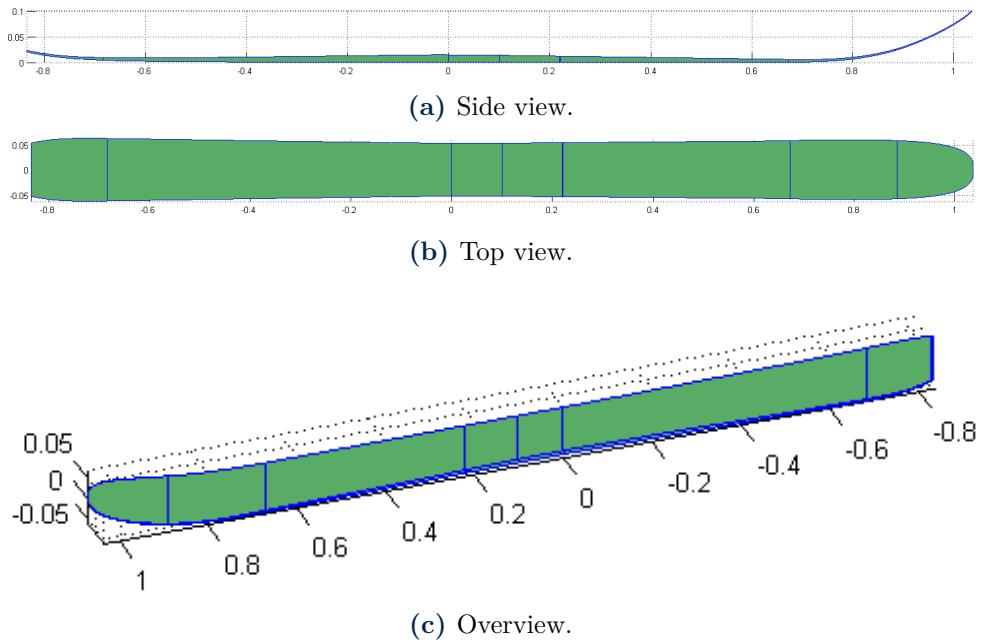


Figure 4.11: The modeled ski in MATLAB. All units are in meters.

When the surface of the volume has been made, knots may be inserted to create different materials in the ski. This will in practice be the extrusion of a cross section which we create in the setback. The problem with such cross section extrusion is that all materials will vary corresponding to the topology of the surface. That is, the height of the steel edge will decrease from the origin and the width will vary as the carving of the ski. The cross section in the origin is the most correct topology, compared with the original ski. But even here, simplifications has been made. This cross section is illustrated in Figure 4.12, and we can see that the curved part of the steel edge has been removed. Moreover, the sidewalls is composed by straight lines at a 4° angle. The parameters should otherwise be the same.



Figure 4.12: The simplified mesh of the cross section used in MATLAB.

Chapter 5

THE FLEX COMPUTER

With the IGA library in our arsenal, we are now ready to tackle more complex geometry and even compute flex curves on skis. We find the flex by measuring the displacement given by an applied force. The stiffness is found by the applied force divided by this displacement. Ideally one should take the measurement where the force is applied, but as we can see from Figure 5.1, the measurement is made between the clamped area and the area of applied force. In the end we would like to compare the data from experiments with our model, so we shall also take the measurement between the clamped area and the force area. To obtain a flex curve several measurement along the ski is taken, typically with 100 mm separating each measurement. The measurement is taken 100 mm from the clamped area and the applied force is taken 120mm from the clamped area. We shall use 2 mm as the force width, and will as Endre Hals take 16 measurements (8 on each half of the Ski) to produce the flex curve which is found by interpolation. In Figure 5.2 the flowchart of the program is shown. Only an overview of the program will be presented here. The code in it's entirety may be found at

[http://org.ntnu.no/skimodeling/isogeometric-analysis.](http://org.ntnu.no/skimodeling/isogeometric-analysis)

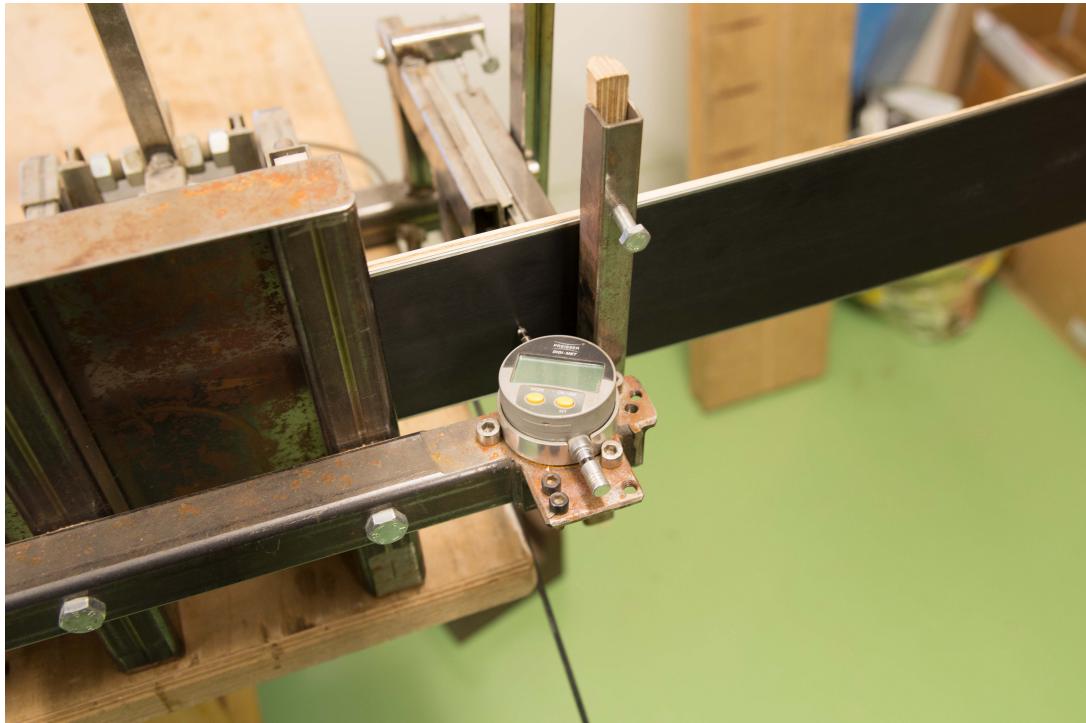


Figure 5.1: This image is taken at the workshop of Endre Hals, and shows the device used to get data for computing the stiffness.

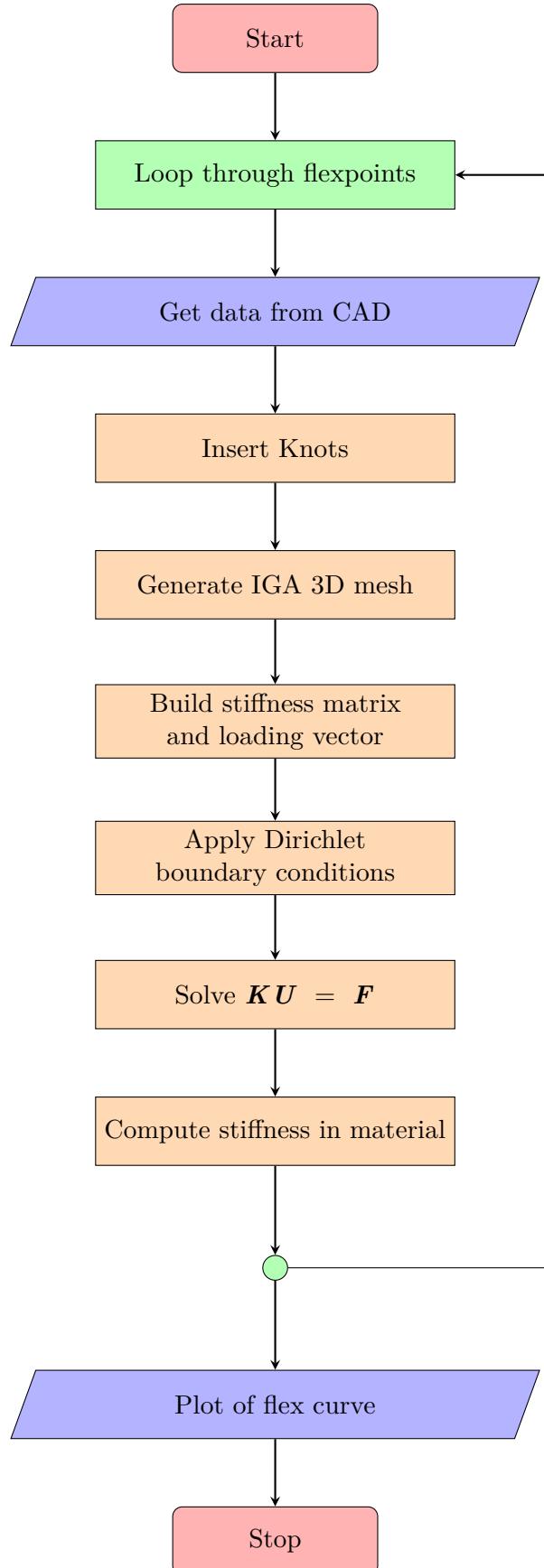


Figure 5.2: Flowchart of the flex computing program. The loop in green actually consists of two loops; one outer loop which loops over the front half of the ski and the back half of the ski and the inner loop loops through the number of flexpoints on the given half (typically 8, such that we get 16 flex points in total).

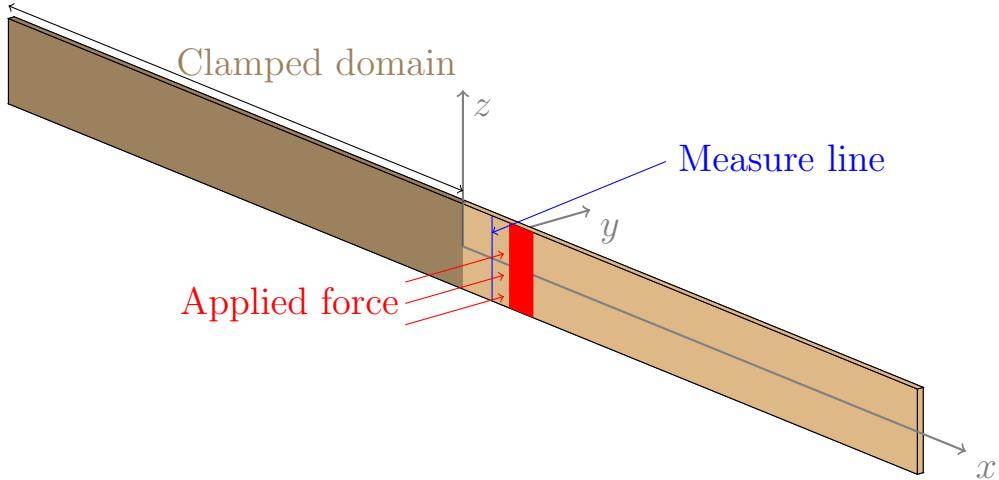


Figure 5.3: Sketch illustrating how the flex is computed on a simple plank. Note that the force width (in x -direction) will be much smaller, it is here chosen larger only for visualization purposes.

The program consist of a for loop which loops through the 16 points of measurements, call them *flexpoints*. When this is done, the program may plot the flex curve based on this data set using spline interpolation. The first step in the for loop would be to get data from CAD. This part is included in the for loop just for convenience. If it was not, we would have to store the control points in a separate array since this will change when inserting knots. When we insert knots we first create repeated knots at the end of the clamped domain such that we have interpolation here (that is, we insert knots into Ξ such that we have p knots in total at the clamped line). Then we insert two further knots into Ξ for the force area; one corresponding to the line at which the force starts, and the second where it ends. Again, this is only done if such a knot does not already exist in the knot vector. We do not need repeated knot in this case, since this knot insertion is only to get exact boundary integration when applying the loading. From here, we may refine the mesh and elevate the order of the NURBS as we like. We do a mesh refinement by inserting knots which is not already in the knot vector. Then, the standard procedure of IGA follows, namely create mesh, build stiffness matrix and loading vector, applying Dirichlet boundary conditions and solving the linear system. After this, the stiffness in the material may be computed. Since we not necessarily have knots at a given flexpoint, we need to create a functions which first finds the coordinates in the parametric space which corresponds to the physical coordinates of the flexpoint, and then another function which evaluates

$$\mathbf{u}(x, y, z) = \sum_{A=1}^{nml} R_A(\xi, \eta, \zeta) \mathbf{U}_A$$

at this point. Obviously we will only need the y -component of the resulting vector for the calculation of the stiffness.

We should provide some numerical evidence that the program is correct. Consider the sketch in Figure 5.3. It shows a simple plank with homogeneous material (wood say, with $E = 13 \cdot 10^{13}$ and $\nu = 0.3$). Since we do not know the analytic solution to this problem, we can not plot the energy norm. What is typically done is a plot of the displacement at a given point as a function

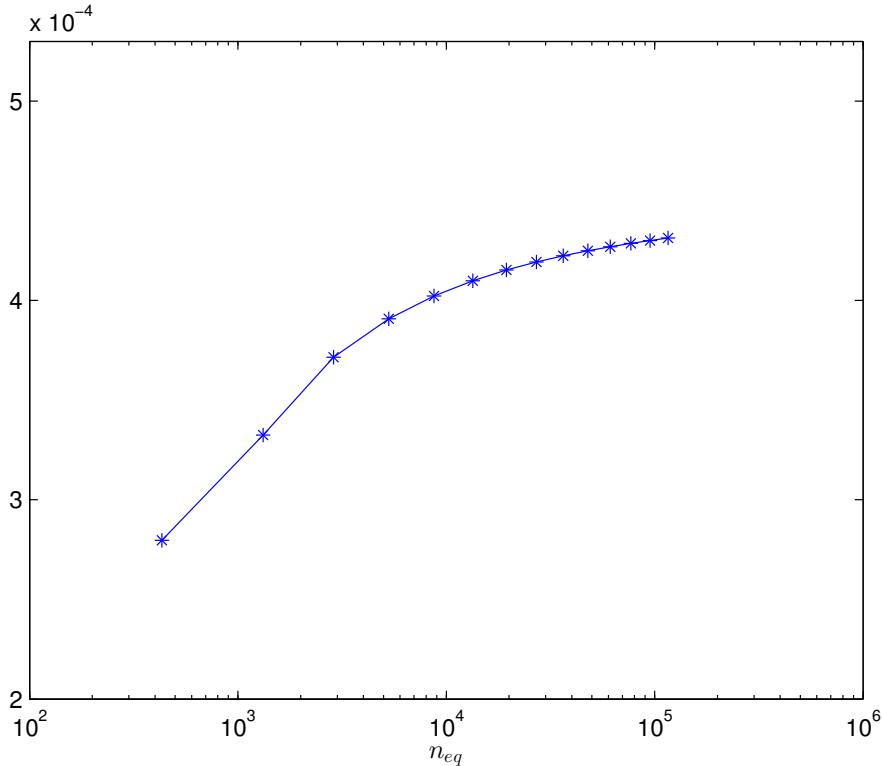


Figure 5.4: Plot of the convergence in displacement.

of the degrees of freedom. The resulting plot should then converge towards the analytic value for the displacement at this location. In this experiment we let $p = q = r = 3$. And we refine homogeneous in all directions. For the ξ direction we only refine between the clamped area to the end of the force area. Outside this area, there is not so much information to be gained by refining. In Figure 5.4 we have the resulting plot. The convergence towards a specific value seems to be rather slow (the relative difference between the last two point is at 0.29%), so it does not yield conclusive evidence that the implementation is correct.

We shall finally present evidence that the model converges to the right answer also with different materials. Consider the same test set up as in the previous example only with different materials inserted. This is done by inserting repeated knots where separation of materials is wanted and then create different elasticity matrices (different values for ν and E) depending on which element is evaluated. In Figure 5.5, y_i and z_i represent the physical coordinate at which we need to insert such repeated knots. Since the inverse of the function, which maps parametric coordinates to physical coordinates is unknown, we have to make a search function which given a physical coordinate, the approximation of a parametric coordinate is returned. We may do this by a binary search algorithm. Since we want to define the cross section in setback, the value for ξ can first be found before knot insertion. When we want to insert knots corresponding to the coordinate y_i , we only need to search for η_i since we could set $\zeta = 0.5$. Correspondingly, if we want to insert knots corresponding to the coordinate z_i , we only need to search for ζ_i since we should set $\eta = 0$ (the coordinates y_i is given at the bottom of the ski).

Now when the topology is defined, we may start refining. We shall here not choose homogeneous refinement due to the different sizes each material occupy. Intuitively one should refine

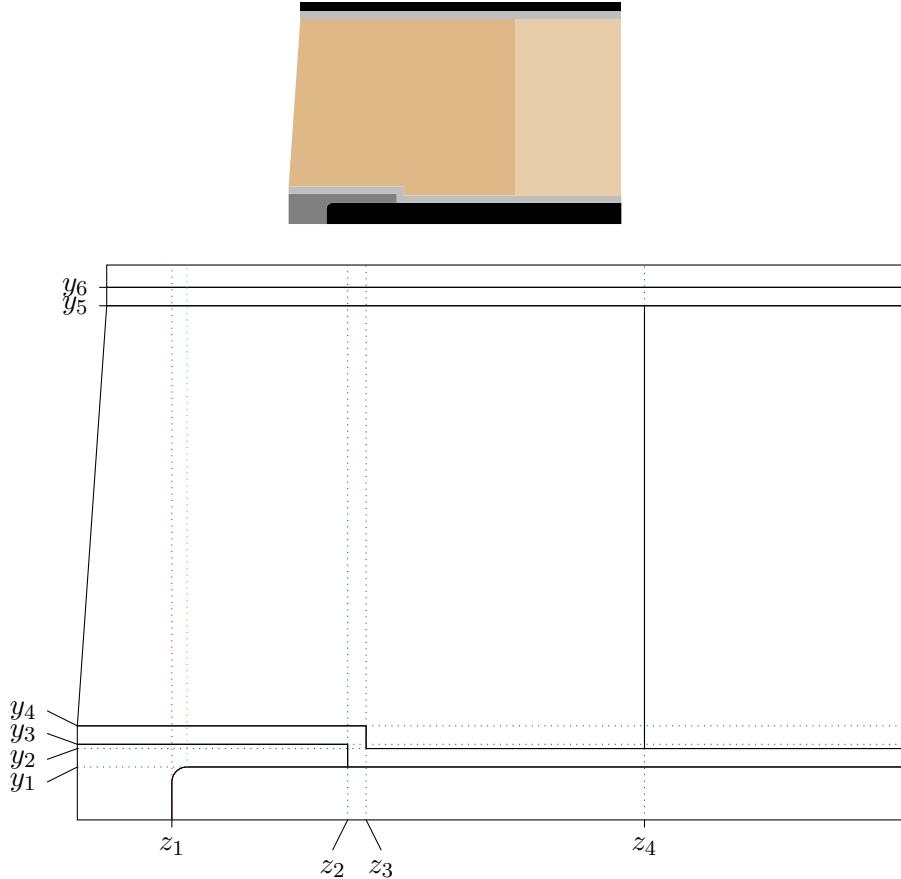
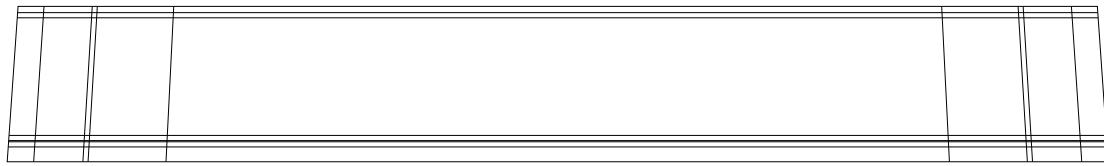


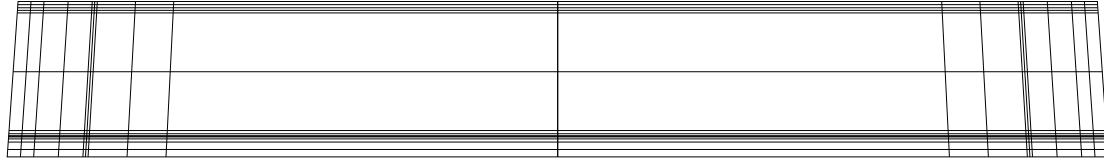
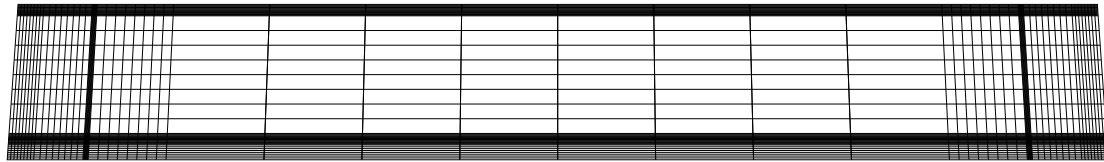
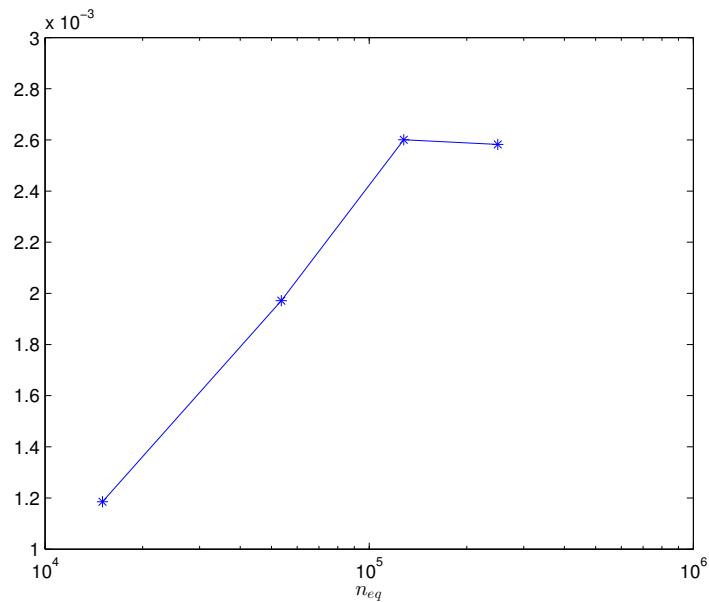
Figure 5.5: Magnified left part of the cross section in Figure 4.1 in the top figure, and the corresponding (even larger magnified) block insertions. One of the refinement is to skip the circular part of the steel edge. This is done by using the red dotted line instead of the green dotted line. If the green line had been chosen, far to much steel would have been added.

equally much in the interval $[\eta_i, \eta_{i+1}]$ as $[\eta_{i+1}, \eta_{i+2}]$ and correspondingly for the ζ direction. Note that the vertical lines given in Figure 5.5 is somewhat misleading. Since the sidewalls of the ski is not vertical (they have 4° angle with the vertical plane), the elements in the physical mesh will also not be vertical (except a possible center line) as we can see from the different physical meshes in Figure 5.6.

In Figure 5.7 we plot the resulting convergence plot of the displacement. The relative difference between the last two point is at 0.7%. This simulation took several hours, so any further data of a more refined mesh was not obtained.



(a) Mesh 1: The original mesh without refinement.

(b) Mesh 2: One knot inserted between each knot interval in both η and ζ direction.(c) Mesh 3: Three knots inserted between each knot interval in both η and ζ direction.(d) Mesh 4: Seven knots inserted between each knot interval in both η and ζ direction.**Figure 5.6:** Different mesh of the cross section.**Figure 5.7:** Different materials: Plot of the convergence in displacement.

Chapter 6

RESULTS AND DISCUSSION

Introductory we want to mention that since the laboratory testing of the complete ski given to us by producer Hals has been completely under our control, we choose these as our main basis for comparison, not the data from Hals' own tests.

Our measured stiffness of the ski at 30 cm behind setback is 102.16 N/mm. The same point in our model gives a stiffness of 70.72 N/mm. Considering the approximated geometry, and all the rest of our approximations, we are very pleased with the result. Still, it should be noted that the high amount of assumptions makes it possible that different assumptions will cause errors which cancels each other out.

Further, if we consider the stress over the cross section in the same point (30 cm behind setback), we can extract some interesting details. Figure 6.1 shows the von Mises stress over parts of the cross section at the point mentioned. The first thing to notice is the clear discontinuity over material borders. The steel edges and composites have the by far highest material stiffness, which is captured well by the colour plot. The wooden core represents a big part of the area, but with the moderate material stiffness it is clear that the contributions to the total stiffness is rather limited. If we look at Figure 6.2, where the stress span captured by the colours is reduced, we can clearly see the variation over the height. This variation is from beam theory expected to be linear, and although it might look right in the figure, we can not firmly conclude whether this is the case here. We also see that the top plastic and the base represent very little stiffness contributions. This also is nothing but expected, since the material stiffness of these components

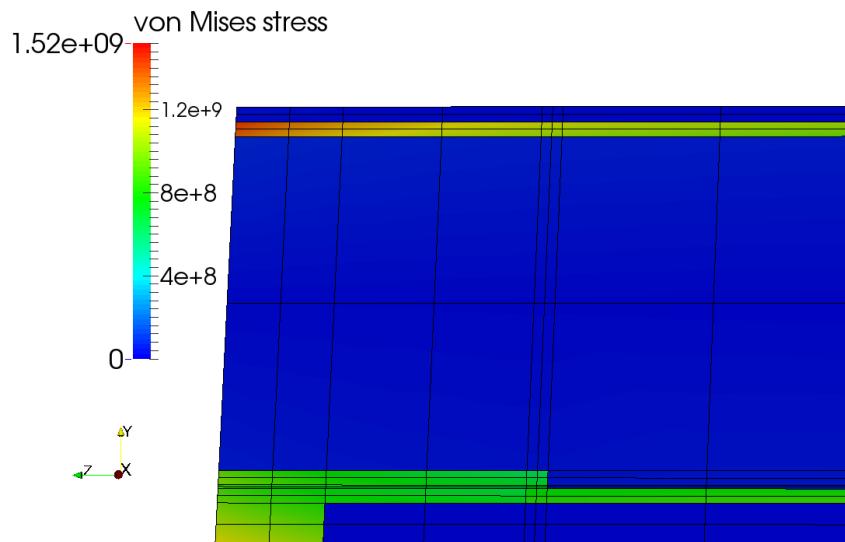


Figure 6.1: Plot of von Mises stress. The black lines illustrates the grid.

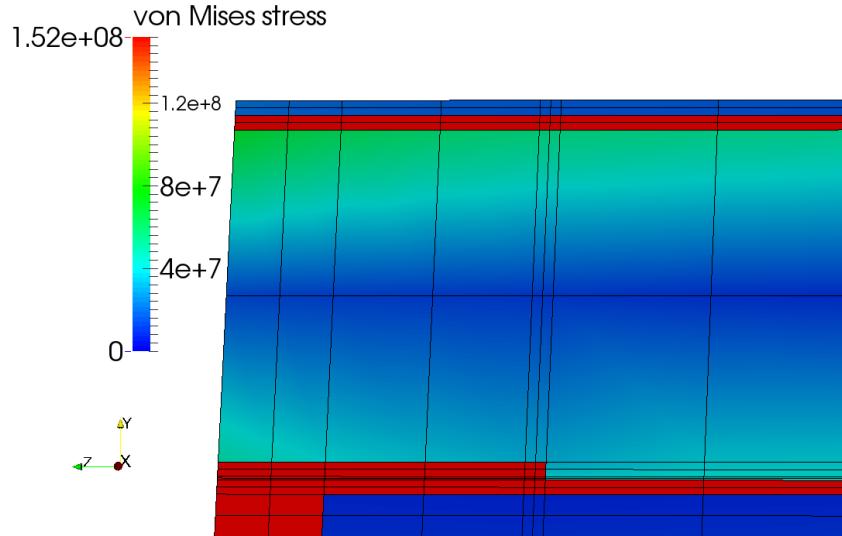


Figure 6.2: Another plot of the von Mises stress. Here, the colouring is set sensitive to a much narrower interval, to emphasise the variation over the wooden core.

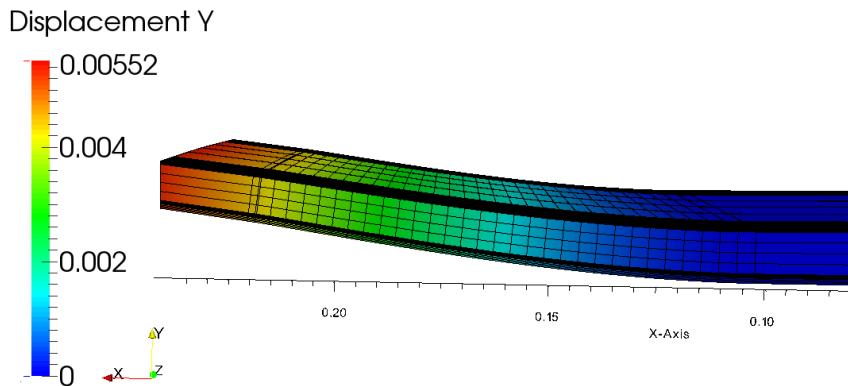


Figure 6.3: Plot of displacement in the area around the clamped end point and the force area. The displacement plot is scaled by a factor 3.

are next to nothing compared to the stiffness of the composite sections and the steel edges.

Investigation of the deformed shape, seen in figure Figure 6.3, supplies further credibility to the model. Again we base our evaluation of the results upon beam theory, which predicts a cubic variation of the displacement under a point loaded cantilever beam. As seen, we have no deformation over the fixed area, whilst the free domain takes a shape which could be well represented by a cubic variation. Beyond the loading area, the curvature of the ski should go to zero, and as expected we see a linear continuation of the angle at the end of loaded area.

The one thing we see which may indicate some sort of error, is the high magnitude of the stress we see in the two cross section figures. This is well beyond the yielding point of most steels, and is clearly not a correct value. This may indicate that some of the softer materials in the cross section, for instance the wooden core, have been assigned a too low material stiffness. A higher stiffness in the softer parts would distribute the stress more evenly over the entire cross section, and hence reduce the maximum value. This will also make the ski as a whole stiffer, and since our representation seems to be about 30% below the real ski, this is not an unlikely reason for the unexpected stress magnitudes.

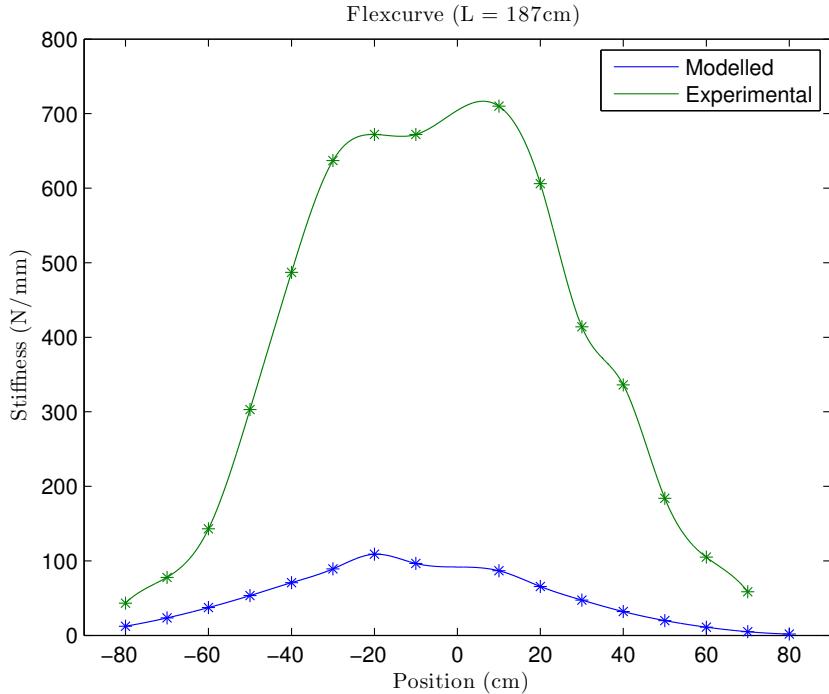


Figure 6.4: The resulting flex curve compared with the flex curve based on the experimental data from Hals. Don't be fooled by the apparent asymmetric appearance of the resulting blue curve. The topology of the used ski is symmetric about the setback point (-10 cm), and the curve should therefore also be symmetric. However, the point at 0 is not measured due to the loading procedure explained to us by Hals, and if it were, it should have the same value as the value at -20 cm.

The resulting flex curve of the ski is given in Figure 6.4. Obviously it is not at all equal to the one based on data from producer Hals, but at least the magnitude of the stiffness seems to be of the same order. Here we would like to point out the fact that the test results from our own data differ significantly from the data given to us by Hals. We do not possess all the details about the loading mechanism used, but it is likely to be some misinterpretations behind the big deviations.

It is important to note that throughout this report, several assumptions has been made. Each will yield an error. Hopefully, these errors are all negligible. We list the following assumptions below.

- We assume linear elasticity which only make sense for small displacements.
- We neglect the glue between materials. That is, we assume all materials are perfectly attached to each other.
- Isotropic material property has been assumed for all non wood material. The wood material is assumed to be simplified version of orthotropic.

Other sources for error can be found in

- The geometry approximations discussed in chapter 4.
- Error in measurements from Hals.

- The results from our laboratory tests, discussed in chapter 3.
- The various sources for the parameter values for Young's modulus (E) and the Poisson's ratio (ν) for the other materials.
- The Galerkin approximation done in the IGA analysis.

Where the geometry approximation arguably represents the biggest source of error.

Chapter 7

WEB PRESENTATION

One of the goals of the project was to make some details of the process, methods and results available to the broader public and upcoming students at NTNU. To obtain easy readability, we chose an online web presentation which can continue to exist for a period. The website can be found at

<http://org.ntnu.no/skimodeling/>.

A general interest was to use up to date web tools, related software and design in order to help create attention and enthusiasm for the project topic. Below we outline some details of the configuration and its special components.

7.1 SETTING UP A CONTENT MANAGEMENT SYSTEM

Developing professional websites from scratch in 2014 requires a vast amount of time and experience. With content requiring mainly only *static* pages, our backend structure of the website could have been simplified greatly. However, using a *dynamic* content management system (CMS) allows for rapid development and more fluid editorial aspects. Moreover, this permits possible future online projects to build on the work relatively efficiently.

With a large volume of CMS providers, the choice of necessary and sufficient software can be a puzzling decision. We settled on WORDPRESS [18], which is a popular open source CMS with a large user base. In addition, its modularity when it comes to plugins and extensive development community were desirable properties.

7.2 DESIGN AND COMPATIBILITY ACROSS DEVICES

Given the time constraints and competence, it was not possible to create the frontend design from bottom up. To compensate, we embraced and installed a free and complete WORDPRESS-theme from THEMESMARTS [19]. Some minor changes were made to suit the desired output.

A key property with regards to modern web designs is the notion of *responsiveness* [20]. With an increasing number of devices and screen formats, websites should be able to scale properly, or be *responsive*, and display customized versions on each device, whether it is a widescreen

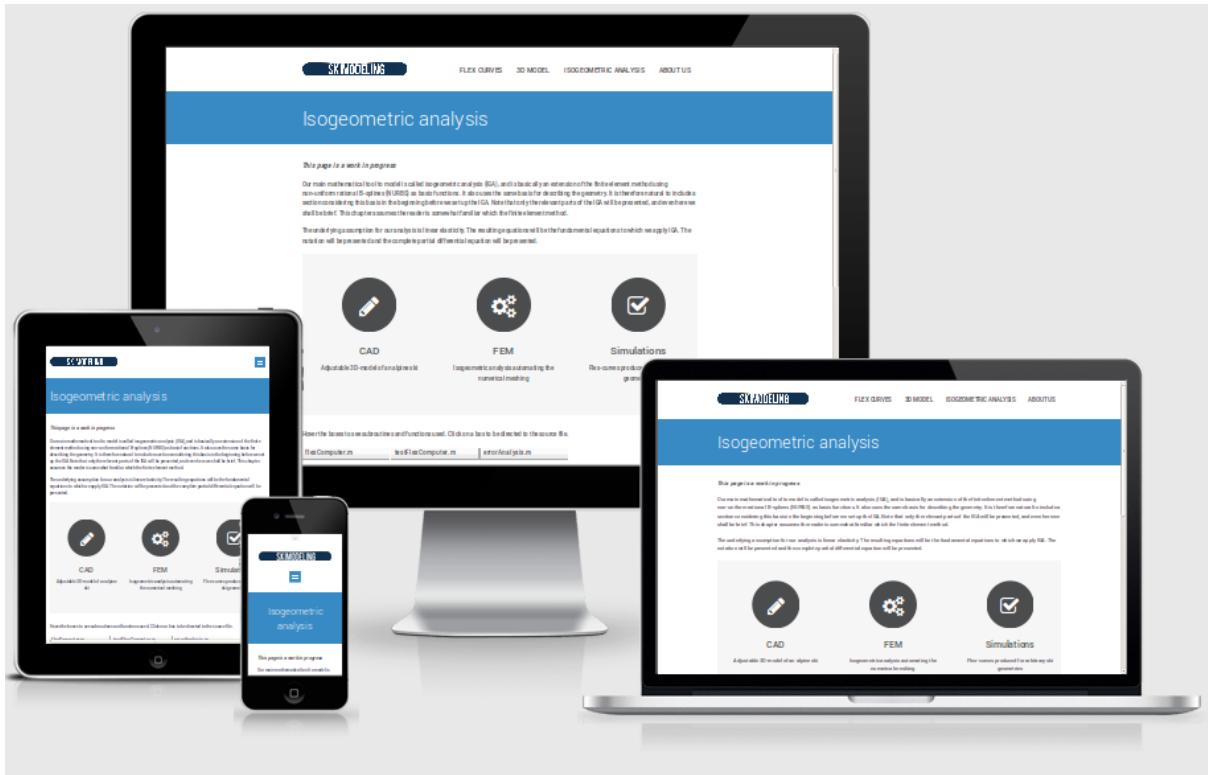


Figure 7.1: Display of the responsiveness of the website on a selection of devices. The image is generated via a tool from <http://ami.responsivedesign.is/>. **REMEMBER: UPDATE FIGURE WITH FINAL WEBSITE**

computer, a small mobile phone or a midsize tablet. In Figure 7.1 is seen how the website neatly adapts to different screen sizes. Both the menu, content columns and images change in order to give the reader a proper view.

DISPLAY OF LATEX-CONTENT

In discussing chapter 2 online, displaying LATEX-content in an elegant way was of great importance. Via a WORDPRESS-plugin [21] utilizing a web service called QUICKLATEX [22], it became possible to publish *raw* LATEX-content without compromising quality. The plugin generates cached images for rendering speed and even enables functionality such as equation referencing.

7.3 A 3D MODEL VIA THE x3dom FRAMEWORK

Since we were already making a 3D model for our analysis, it was useful to find a tool for presenting it on the website. By exporting the CAD model to a file type known as Virtual Reality Modeling Language (VRML), we found a way to display the 3D ski into a browser-understandable format using the so-called x3dom¹ framework [23].

¹x3dom is pronounced «x-freedom».

The open-source `x3dom`-framework uses X3D data embedded in (X)HTML pages. Moreover, X3D is a data format used for representing 3D computer graphics, being the successor of VRML. To this end, the VRML file had to be converted into an X3D file type, which then was exported again into a HTML readable format. There were no loss of quality between the steps, and thus the online 3D model is essentially the same as the one used in the CAD software.

Chapter 8

CONCLUSIONS AND SUGGESTED FURTHER WORK

Due to inconsistency with experimental data done by us and Endre Hals, the data from Endre Hals was not chosen to validate results made from the modeling. All the sources of errors taken into account, the modeled result came close to the experimental result. Still, some results deviates from what was to be expected and we can not exclude the possibility of an implementation error.

As mentioned, the exact geometry is a work in progress, and to use this in the Flex Computer would be the obvious next step. When everything seem to fit the experimental data, one should be able to consider, to what extent each material contributes to the stiffness of the ski, and also actually calculate the SFI. This way, one could really make use of the developed program. A further study could then be to investigate what properties of a ski constitutes an ideal ski for a given costumer.

The flex computing program is a rather heavy program to run. Many optimizations can be done to decrease the computational time. This include implementing T-splines (such that local refinement of the IGA code is possible), optimize the assembly process (as discussed) and implement the code in c. The latter optimization should especially be done for the NURBS implementation (which was done in [5]).

Appendix A

PARAMETER VALUES FOR DISPLAYED SKI

(a) Cross section

Parameter	Value	Unit
st	0.001	m
skit	0.0006	m
skb	0.00715	m
skyb	0.0025	m
R	0.0004	m
bkt	0.000486	m
tkt	0.000486	m
tpt	0.0005	m
ms	0.01171	m
B	0.052	m
ysb	0.015	m
θ	4	Deg
msb	0.023	m

(b) Top View

Parameter	Value	Unit
Ltot	1.87	m
TFL	0.36	m
TFR	5	m
TBL	0.34	m
TBR	5	m
Wsb	0.052	m
SB	0.1	m
SR	25	m

(c) Side view

Parameter	Value	Unit
RFR	5	m
RBR	5	m
RFL	0.5	m
RBL	0.45	m
R2FR	0.8	m
R2FL	0.24	m
btRest	0.105	m
ftRest	0.115	m
bt	0.00192	m
bs	0.00701	m
fs	0.00687	m
ft	0.00184	m

Note that the parameters are listed under the view where it is introduced, and may well be used in several views.

Appendix B

LABORATORY TESTS

Table B.1: Test of E-modulus for wood core.

(a) Cross-section 1 front ski 1.			(b) Cross-section 2 front ski 1.			(c) Cross-section 1 back ski 1.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	0.869	0.079	1	1.453	0.2192	1	1.453	0.096
2	1.184	0.094	2	1.768	0.2741	2	1.768	0.1408
3	1.499	0.117	3	2.083	0.3	3	2.083	0.151
4	0.869	0.06	4	2.398	0.3619	4	2.398	0.1737
5	1.184	0.081	5	2.713	0.4251	5	2.713	0.1939
6	1.499	0.102	6	1.453	0.1952	6	1.453	0.09537
7	1.453	0.1307	7	1.768	0.2747	7	1.768	0.1352
8	1.768	0.1705	8	2.083	0.2956	8	2.083	0.144
9	2.083	0.1768	9	2.398	0.3436	9	2.398	0.1693
10	2.398	0.2173	10	2.713	0.3985	10	2.713	0.187
11	2.713	0.2457	11	1.453	0.2147	11	1.453	0.09448
			12	1.768	0.2918	12	1.768	0.1339
			13	2.083	0.3139	13	2.083	0.1415
			14	2.398	0.3651	14	2.398	0.1667
			15	2.713	0.4213	15	2.713	0.1958

(d) Cross-section 2 back ski 1.			(e) Cross-section 3 back ski 1.			(f) Cross-section 1 front ski 2.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1592	1	1.453	0.2286	1	1.453	0.12
2	1.768	0.2154	2	1.768	0.2836	2	1.768	0.1478
3	2.083	0.2425	3	2.083	0.3322	3	2.083	0.1851
4	2.398	0.2825	4	2.398	0.3821	4	2.398	0.2166
5	2.713	0.324	5	2.713	0.4345	5	2.713	0.2444
6	1.453	0.1497	6	1.453	0.2261	6	1.453	0.1067
7	1.768	0.1983	7	1.768	0.2811	7	1.768	0.1377
8	2.083	0.2192	8	2.083	0.3278	8	2.083	0.1623
9	2.398	0.259	9	2.398	0.3777	9	2.398	0.1895
10	2.713	0.3057	10	2.713	0.4364	10	2.713	0.2255
11	1.453	0.1364	11	1.453	0.2103	11	1.453	0.1251
12	1.768	0.1832	12	1.768	0.2691	12	1.768	0.1573
13	2.083	0.2141	13	2.083	0.312	13	2.083	0.1731
14	2.398	0.2539	14	2.398	0.36	14	2.398	0.2008
15	2.713	0.2975	15	2.713	0.4232	15	2.713	0.2305

(g) Cross-section 2 front ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.2097
2	1.768	0.2545
3	2.083	0.3019
4	2.398	0.3493
5	2.713	0.3954
6	1.453	0.1926
7	1.768	0.2432
8	2.083	0.2943
9	2.398	0.343
10	2.713	0.3997
11	1.453	0.2059
12	1.768	0.2425
13	2.083	0.2956
14	2.398	0.3392
15	2.713	0.3884

(h) Cross-section 3 front ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.2716
2	1.768	0.3398
3	2.083	0.408
4	2.398	0.4699
5	2.713	0.54
6	1.453	0.271
7	1.768	0.3436
8	2.083	0.4074
9	2.398	0.4718
10	2.713	0.54
11	1.453	0.2785
12	1.768	0.3385
13	2.083	0.3998
14	2.398	0.463
15	2.713	0.5305

(i) Cross-section 1 back ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.09285
2	1.768	0.144
3	2.083	0.1446
4	2.398	0.1535
5	2.713	0.186
6	1.453	0.1086
7	1.768	0.1093
8	2.083	0.1421
9	2.398	0.1421
10	2.713	0.1516
11	1.453	0.03347
12	1.768	0.05432
13	2.083	0.1017
14	2.398	0.1086
15	2.713	0.1301

(j) Cross-section 2 back ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.0739
2	1.768	0.103
3	2.083	0.1402
4	2.398	0.192
5	2.713	0.2116
6	1.453	0.1383
7	1.768	0.1787
8	2.083	0.1983
9	2.398	0.2248
10	2.713	0.2653
11	1.453	0.1118
12	1.768	0.1541
13	2.083	0.1863
14	2.398	0.24
15	2.713	0.2564

(k) Cross-section 3 back ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1863
2	1.768	0.2501
3	2.083	0.2855
4	2.398	0.3234
5	2.713	0.3859
6	1.453	0.1055
7	1.768	0.1547
8	2.083	0.2091
9	2.398	0.2741
10	2.713	0.3297
11	1.453	0.1585
12	1.768	0.2128
13	2.083	0.2943
14	2.398	0.3335
15	2.713	0.3771

Table B.3: Test of shear-modulus for wood core.

(a) Cross-section 1 front ski 1.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.2815	0.025 26
2	1.8602	0.4322	0.052 42
3	2.4902	0.58	0.069 48
4	3.2102	0.732	0.096
5	1.2302	0.2718	0.003 79
6	1.8602	0.418	0.0278
7	2.4902	0.5616	0.039 16
8	3.2102	0.704	0.058 74
9	1.2302	0.2734	0.015 16
10	1.8602	0.418	0.041 69
11	2.4902	0.5668	0.0517
12	3.2102	0.7149	0.072

(b) Cross-section 2 front ski 1.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.36	0.052 42
2	1.8602	0.545	0.1074
3	2.4902	0.7323	0.1333
4	3.2102	0.925	0.189
5	1.2302	0.3555	0.058 74
6	1.8602	0.5417	0.1086
7	2.4902	0.73	0.1421
8	3.2102	0.9191	0.1832
9	1.2302	0.36	0.049 26
10	1.8602	0.5455	0.1074
11	2.4902	0.7355	0.1352
12	3.2102	0.921	0.175

(c) Cross-section 1 back ski 1.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.1797	0.022 74
2	1.8602	0.2686	0.048
3	2.4902	0.3626	0.060 63
4	3.2102	0.4612	0.069 48
5	1.2302	0.1771	0.032 21
6	1.8602	0.2705	0.042 42
7	2.4902	0.3645	0.056 84
8	3.2102	0.4592	0.060 63
9	1.2302	0.1752	0.018 32
10	1.8602	0.266	0.024 63
11	2.4902	0.36	0.041 05
12	3.2102	0.4592	0.067 58

(d) Cross-section 2 back ski 1.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.2332	0.075 79
2	1.8602	0.3523	0.1105
3	2.4902	0.4766	0.1288
4	3.2102	0.5996	0.1623
5	1.2302	0.2261	0.050 53
6	1.8602	0.3452	0.108
7	2.4902	0.4663	0.1187
8	3.2102	0.5887	0.139
9	1.2302	0.2274	0.050 53
10	1.8602	0.3491	0.1143
11	2.4902	0.4702	0.1288
12	3.2102	0.5926	0.1566

(e) Cross-section 3 back ski 1.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.2795	0.1131
2	1.8602	0.4232	0.1743
3	2.4902	0.5694	0.2089
4	3.2102	0.7201	0.2774
5	1.2302	0.277	0.1023
6	1.8602	0.4232	0.1699
7	2.4902	0.575	0.2149
8	3.2102	0.722	0.2779
9	1.2302	0.275	0.095 37
10	1.8602	0.4219	0.168
11	2.4902	0.5694	0.2128
12	3.2102	0.724	0.2817

(f) Cross-section 1 front ski 2.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.2473	0.041 05
2	1.8602	0.3716	0.058 74
3	2.4902	0.4959	0.084 63
4	3.2102	0.619	0.097 27
5	1.2302	0.2396	0.041 05
6	1.8602	0.3613	0.058 74
7	2.4902	0.4837	0.081 45
8	3.2102	0.6054	0.095 37
9	1.2302	0.2396	0.034 11
10	1.8602	0.3639	0.060 63
11	2.4902	0.4844	0.082 74
12	3.2102	0.6067	0.099 79

(g) Cross-section 2 front ski 2.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.3208	0.08463
2	1.8602	0.4792	0.127
3	2.4902	0.6383	0.1686
4	3.2102	0.798	0.2127
5	1.2302	0.3162	0.084
6	1.8602	0.4811	0.1295
7	2.4902	0.6389	0.1724
8	3.2102	0.7955	0.204
9	1.2302	0.3117	0.07579
10	1.8602	0.4721	0.1181
11	2.4902	0.6319	0.1636
12	3.2102	0.7929	0.2027

(h) Cross-section 3 front ski 2.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.3819	0.1143
2	1.8602	0.5739	0.1945
3	2.4902	0.7685	0.2735
4	3.2102	0.9584	0.3215
5	1.2302	0.3832	0.1213
6	1.8602	0.5752	0.1926
7	2.4902	0.771	0.2691
8	3.2102	0.9616	0.3316
9	1.2302	0.3819	0.1326
10	1.8602	0.5765	0.199
11	2.4902	0.7697	0.2621
12	3.2102	0.9636	0.3354

(i) Cross-section 1 back ski 2.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.1894	0.04232
2	1.8602	0.2976	0.06253
3	2.4902	0.4	0.07579
4	3.2102	0.505	0.1042
5	1.2302	0.1881	0.03284
6	1.8602	0.286	0.05242
7	2.4902	0.3923	0.08463
8	3.2102	0.489	0.1004
9	1.2302	0.1874	0.03537
10	1.8602	0.285	0.05874
11	2.4902	0.385	0.0767
12	3.2102	0.4844	0.09916

(j) Cross-section 2 back ski 2.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.248	0.07895
2	1.8602	0.3781	0.1206
3	2.4902	0.5095	0.168
4	3.2102	0.6409	0.2185
5	1.2302	0.248	0.08789
6	1.8602	0.3768	0.1213
7	2.4902	0.5089	0.1667
8	3.2102	0.6383	0.2116
9	1.2302	0.2525	0.0859
10	1.8602	0.38	0.1219
11	2.4902	0.5101	0.1642
12	3.2102	0.6422	0.2167

(k) Cross-section 3 back ski 2.

Test no.	Weight [kg]	Deflection [mm]	Deflection [mm]
1	1.2302	0.3111	0.1288
2	1.8602	0.4728	0.2002
3	2.4902	0.6338	0.2716
4	3.2102	0.8	0.35
5	1.2302	0.3085	0.1244
6	1.8602	0.4695	0.1964
7	2.4902	0.6312	0.2667
8	3.2102	0.7922	0.3392
9	1.2302	0.3085	0.1408
10	1.8602	0.4692	0.1965
11	2.4902	0.6299	0.2667
12	3.2102	0.7929	0.3493

Table B.5: Test of E-modulus for ash sections.

(a) Cross-section 1.			(b) Cross-section 2.			(c) Cross-section 3.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	0.334	0.2128	1	0.334	0.2154	1	0.334	0.168
2	0.649	0.439	2	0.649	0.403	2	0.649	0.3227
3	0.964	0.6663	3	0.964	0.6133	3	0.964	0.4945
4	0.334	0.2065	4	0.334	0.18	4	0.334	0.1648
5	0.649	0.4194	5	0.649	0.3746	5	0.649	0.3082
6	0.964	0.6512	6	0.964	0.5893	6	0.964	0.475
7	0.334	0.2116	7	0.334	0.2261	7	0.334	0.1516
8	0.649	0.4251	8	0.649	0.4465	8	0.649	0.3063
9	0.964	0.6487	9	0.964	0.6589	9	0.964	0.4724

(d) Cross-section 4.			(e) Cross-section 5.			(f) Cross-section 6.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	0.334	0.187	1	0.334	0.1617	1	0.334	0.1901
2	0.649	0.3606	2	0.649	0.3139	2	0.649	0.3474
3	0.964	0.5514	3	0.964	0.487	3	0.964	0.5255
4	0.334	0.1686	4	0.334	0.1352	4	0.334	0.1699
5	0.649	0.3373	5	0.649	0.3019	5	0.649	0.3101
6	0.964	0.5166	6	0.964	0.4636	6	0.964	0.4838
7	0.334	0.1699	7	0.334	0.1402	7	0.334	0.1648
8	0.649	0.3335	8	0.649	0.295	8	0.649	0.3114
9	0.964	0.5236	9	0.964	0.4611	9	0.964	0.4914

(g) Cross-section 7.			(h) Cross-section 8.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	0.334	0.2217	1	0.334	0.2369
2	0.649	0.4345	2	0.649	0.4263
3	0.964	0.667	3	0.964	0.6588
4	0.334	0.2154	4	0.334	0.2046
5	0.649	0.4213	5	0.649	0.4213
6	0.964	0.6455	6	0.964	0.6348
7	0.334	0.2147	7	0.334	0.1983
8	0.649	0.4194	8	0.649	0.4068
9	0.964	0.6322	9	0.964	0.6329

Table B.6: Test of shear-modulus for ash sections.

(a) Cross-section 1.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.1
2	2.175	0.1
3	2.805	0.2
4	1.545	0.04
5	2.175	0.1
6	2.805	0.15

(b) Cross-section 2.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.1
2	2.175	0.2
3	2.805	0.2
4	1.545	0.1
5	2.175	0.2
6	2.805	0.25

(c) Cross-section 3.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.1
2	2.175	0.2
3	2.805	0.3
4	1.545	0.1
5	2.175	0.2
6	2.805	0.3

(d) Cross-section 4.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.2
2	2.175	0.2
3	2.805	0.3
4	1.545	0.2
5	2.175	0.2
6	2.805	0.3

(e) Cross-section 5.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.2
2	2.175	0.2
3	2.805	0.3
4	1.545	0.1
5	2.175	0.2
6	2.805	0.3

(f) Cross-section 6.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.2
2	2.175	0.2
3	2.805	0.3
4	1.545	0.15
5	2.175	0.2
6	2.805	0.3

(g) Cross-section 7.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.1
2	2.175	0.2
3	2.805	0.3

(h) Cross-section 8.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.2
2	2.175	0.3
3	2.805	0.3
4	1.545	0.2
5	2.175	0.3
6	2.805	0.4

Table B.7: Test of E-modulus for aspen sections.

(a) Cross-section 1.			(b) Cross-section 2.			(c) Cross-section 3.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	0.334	0.2962	1	0.334	0.7175	1	0.334	0.2116
2	0.649	0.6007	2	0.649	1.336	2	0.649	0.4232
3	0.964	0.912	3	0.964	1.88	3	0.964	0.6417
4	0.334	0.2779	4	0.334	0.732	4	0.334	0.2046
5	0.649	0.5564	5	0.649	1.347	5	0.649	0.4011
6	0.964	0.8539	6	0.964	1.908	6	0.964	0.612
7	0.334	0.271	7	0.334	0.7434	7	0.334	0.2059
8	0.649	0.5545	8	0.649	1.356	8	0.649	0.403
9	0.964	0.8457	9	0.964	1.916	9	0.964	0.6114

(d) Cross-section 4.			(e) Cross-section 5.			(f) Cross-section 6.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	0.334	0.5981	1	0.334	0.2053	1	0.334	0.5103
2	0.649	1.119	2	0.649	0.4188	2	0.649	0.9398
3	0.964	1.484	3	0.964	0.6164	3	0.964	1.304
4	0.334	0.5767	4	0.334	0.2147	4	0.334	0.5217
5	0.649	1.095	5	0.649	0.4244	5	0.649	0.9613
6	0.964	1.44	6	0.964	0.631	6	0.964	1.314
7	0.334	0.6291	7	0.334	0.2192	7	0.334	0.523
8	0.649	1.156	8	0.649	0.4276	8	0.649	0.9607
9	0.964	1.501	9	0.964	0.6379	9	0.964	1.311

(g) Cross-section 7.			(h) Cross-section 8.		
Test no.	Weight [kg]	Deflection [mm]	Test no.	Weight [kg]	Deflection [mm]
1	0.334	0.276	1	0.334	1.008
2	0.649	0.5476	2	0.649	1.84
3	0.964	0.8362	3	0.964	2.377
4	0.334	0.295	4	0.334	1.022
5	0.649	0.5665	5	0.649	1.851
6	0.964	0.8508	6	0.964	2.378
7	0.334	0.2918	7	0.334	1.02
8	0.649	0.564	8	0.649	1.859
9	0.964	0.8564	9	0.964	2.384

Table B.8: Test of shear-modulus for aspen sections.

(a) Cross-section 1.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.4
2	2.175	0.5
3	2.805	0.6
4	1.545	0.35
5	2.175	0.5
6	2.805	0.6

(b) Cross-section 2.

Test no.	Weight [kg]	Rotation [°]
1	1.545	1.2
2	2.175	1.3
3	2.805	1.4
4	1.545	1.1
5	2.175	0.12
6	2.805	1.3

(c) Cross-section 3.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.5
2	2.175	0.65
3	2.805	0.8
4	1.545	0.4
5	2.175	0.55
6	2.805	0.75

(d) Cross-section 4.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.75
2	2.175	1
3	2.805	1.2
4	1.545	0.75
5	2.175	1.5
6	2.805	1.35

(e) Cross-section 5.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.3
2	2.175	0.45
3	2.805	0.6
4	1.545	0.3
5	2.175	0.45
6	2.805	0.6

(f) Cross-section 6.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.5
2	2.175	0.6
3	2.805	0.8
4	1.545	0.45
5	2.175	0.65
6	2.805	0.8

(g) Cross-section 7.

Test no.	Weight [kg]	Rotation [°]
1	1.545	0.5
2	2.175	0.7
3	2.805	0.9
4	1.545	0.4
5	2.175	0.6
6	2.805	0.8

(h) Cross-section 8.

Test no.	Weight [kg]	Rotation [°]
1	1.545	1.25
2	2.175	1.75
3	2.805	2.15
4	1.545	1.3
5	2.175	1.8
6	2.805	2.3

Appendix C

LABORATORY TESTS OF FINISHED SKIS

Table C.1: Test of deflection for the first finished ski.

(a) Cross-section 1 front ski 1.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.076 65
2	1.768	0.098 85
3	2.083	0.1108
4	2.398	0.1275
5	1.453	0.068 92
6	1.768	0.9339
7	2.083	0.1063
8	2.398	0.1211
9	1.453	0.070 21
10	1.768	0.088 24
11	2.083	0.1037
12	2.398	0.1211

(b) Cross-section 2 front ski 1.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1043
2	1.768	0.134
3	2.083	0.1539
4	2.398	0.1758
5	1.453	0.1063
6	1.768	0.1327
7	2.083	0.1559
8	2.398	0.1771
9	1.453	0.1031
10	1.768	0.1356
11	2.083	0.1565
12	2.398	0.1797

(c) Cross-section 3 front ski 1.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1423
2	1.768	0.1797
3	2.083	0.2093
4	2.398	0.2402
5	1.453	0.1443
6	1.768	0.1842
7	2.083	0.2093
8	2.398	0.2435
9	1.453	0.1372
10	1.768	0.1752
11	2.083	0.2074
12	2.398	0.237

(d) Cross-section 1 back ski 1.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1417
2	1.768	0.1745
3	2.083	0.1952
4	2.398	0.2222
5	1.453	0.134
6	1.768	0.1713
7	2.083	0.1926
8	2.398	0.2196
9	1.453	0.1398
10	1.768	0.1662
11	2.083	0.1907
12	2.398	0.2151

(e) Cross-section 2 back ski 1.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1913
2	1.768	0.2377
3	2.083	0.2712
4	2.398	0.3066
5	1.453	0.1919
6	1.768	0.2344
7	2.083	0.2699
8	2.398	0.3072
9	1.453	0.1939
10	1.768	0.2338
11	2.083	0.2699
12	2.398	0.3066

(f) Cross-section 3 back ski 1.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.2473
2	1.768	0.2963
3	2.083	0.3381
4	2.398	0.3826
5	1.453	0.2467
6	1.768	0.2963
7	2.083	0.3401
8	2.398	0.3845
9	1.453	0.246
10	1.768	0.2979
11	2.083	0.3407
12	2.398	0.3832

Table C.2: Test of deflection for the second finished ski.

(g) Cross-section 1 front ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.06119
2	1.768	0.08309
3	2.083	0.09275
4	2.398	0.1089
5	1.453	0.05732
6	1.768	0.07793
7	2.083	0.09017
8	2.398	0.1063
9	1.453	0.05668
10	1.768	0.07793
11	2.083	0.0921
12	2.398	0.1056

(h) Cross-section 2 front ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.09533
2	1.768	0.1217
3	2.083	0.143
4	2.398	0.1662
5	1.453	0.08953
6	1.768	0.1153
7	2.083	0.1385
8	2.398	0.1597
9	1.453	0.08824
10	1.768	0.114
11	2.083	0.1353
12	2.398	0.1604

(i) Cross-section 3 front ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.123
2	1.768	0.1572
3	2.083	0.1874
4	2.398	0.219
5	1.453	0.1159
6	1.768	0.1533
7	2.083	0.1797
8	2.398	0.2106
9	1.453	0.125
10	1.768	0.1552
11	2.083	0.1855
12	2.398	0.2222

(j) Cross-section 1 back ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1514
2	1.768	0.1952
3	2.083	0.2216
4	2.398	0.2525
5	1.453	0.1418
6	1.768	0.1881
7	2.083	0.2125
8	2.398	0.2473
9	1.453	0.1282
10	1.768	0.1784
11	2.083	0.1997
12	2.398	0.2267

(k) Cross-section 2 back ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.1913
2	1.768	0.2415
3	2.083	0.2782
4	2.398	0.3182
5	1.453	0.1939
6	1.768	0.2525
7	2.083	0.2802
8	2.398	0.3208
9	1.453	0.1932
10	1.768	0.2454
11	2.083	0.2821
12	2.398	0.3201

(l) Cross-section 3 back ski 2.

Test no.	Weight [kg]	Deflection [mm]
1	1.453	0.2667
2	1.768	0.3227
3	2.083	0.3736
4	2.398	0.4167
5	1.453	0.2506
6	1.768	0.3137
7	2.083	0.3626
8	2.398	0.4122
9	1.453	0.2705
10	1.768	0.3304
11	2.083	0.3839
12	2.398	0.4257

REFERENCES

- [1] E. Hals, “Introducing the Ski Flex Index (SFI).” [Online]. Available: <http://www.endrehals.no/>
- [2] P. Mastrogiovanni, “The effects of core material and thickness on the performance and behaviour of a ski,” 2007. [Online]. Available: <http://www.skibuilders.com/articles/EffectsofCore.pdf>
- [3] T. Hughes, J. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39–41, pp. 4135 – 4195, 2005.
- [4] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, 1st ed. Wiley, 9 2009.
- [5] V. P. Nguyen, R. N. Simpson, S. P. A. Bordas, and T. Rabczuk, “Isogeometric analysis: an overview and computer implementation aspects,” *CoRR*, vol. abs/1205.2129v2, 2012.
- [6] A. Karatarakis, P. Karakitsios, and M. Papadrakakis, “Computation of the isogeometric analysis stiffness matrix on gpu,” 2013. [Online]. Available: <http://eccomasproceedings.org/cs2013/pdf/2143.pdf>
- [7] T. Hughes, A. Reali, and G. Sangalli, “Efficient quadrature for nurbs-based isogeometric analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 5–8, pp. 301 – 313, 2010, computational Geometry and Analysis.
- [8] Kitware, “File formats for vtk version 4.2,” 2010. [Online]. Available: <http://www.vtk.org/VTK/img/file-formats.pdf>
- [9] N. Nguyen-Thanh, H. Nguyen-Xuan, S. Bordas, and T. Rabczuk, “Isogeometric analysis using polynomial splines over hierarchical t-meshes for two-dimensional elastic solids,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 21–22, pp. 1892 – 1908, 2011.
- [10] K. K. Leang, “Skibuilders.com llc,” 2011. [Online]. Available: <http://www.skibuilders.com/>
- [11] R. T. Haftka, “Lecture: Applied elasticity and advanced mechanics of solids,” 2008. [Online]. Available: <http://www2.mae.ufl.edu/haftka/adv-elast/lectures/sections6.5-6.pdf>
- [12] B. Kucera and R. M. Næss, *Tre - naturens vakreste råstoff*. Landbruksforlaget, 1999. [Online]. Available: <http://www.lauvtrebruk.no/pages/26>

- [13] Unicarbon, “Technical data sheet,” 2014. [Online]. Available: <http://www.unicarbon.com/products/unidry/unidry-150.html>
- [14] Forelesningnotater, “Tmm4155 specialization composites 2013,” 2013.
- [15] H. S. K. John V. Milewski, *Handbook of Reinforcement for Plastics*. Van Nostrand Reinhold Co., Sep. 1987. [Online]. Available: <http://www.ceramicindustry.com/ext/resources/CI/Protected/Files/PDF/fiberglass-compositions.pdf>
- [16] Goodfellow, “Material information (uhmw pe),” 2014. [Online]. Available: <http://www.goodfellow.com/E/Polyethylene-UHMW.html>
- [17] MatWeb, “Overview of material properties for polyester film,” 2014. [Online]. Available: <http://www.matweb.com/search/DataSheet.aspx?MatGUID=40559706b4fd4aa0a43f5739799728f5>
- [18] WordPress, “Blog Tool, Publishing Platform, and CMS,” 2014. [Online]. Available: <http://wordpress.org/>
- [19] ThemeSmarts Free WordPress Themes, “Simple Dark WordPress Theme,” 2013. [Online]. Available: <http://themesmarts.com/themes/simple-dark-wordpress-theme/>
- [20] Wikipedia, “Responsive web design,” 2014. [Online]. Available: http://en.wikipedia.org/wiki/Responsive_web_design
- [21] WP QuickLaTeX, 2014. [Online]. Available: <https://wordpress.org/plugins/wp-quicklatex/>
- [22] QuickLaTeX, “Advanced L^AT_EX web rendering service,” 2011. [Online]. Available: <http://quicklatex.com/>
- [23] x3dom, “Instant 3D the HTML way!” 2014. [Online]. Available: <http://www.x3dom.org/>
- [24] Y. Bazilevs, “Isogeometric Analysis of Turbulence and Fluid-Structure Interaction,” 2006. [Online]. Available: http://www.lib.utexas.edu/etd/d/2006/bazilevsj57469/bazilevsj57469.pdf?origin=publication_detail
- [25] V. P. Nguyen and S. P. A. Bordas, *Extended Isogeometric Analysis for Strong and Weak Discontinuities*, Cardiff University, Wales, UK.
- [26] I. Altman, “Writing a Publishable Math Paper,” 2009. [Online]. Available: <http://math.mit.edu/ci-sandbox/wp-content/uploads/2009/12/Altman1.pdf>