NTNU – Trondheim
Norwegian University of
Science and Technology

Nonlinear Isogeometric Analysis vs NFEA of Tubular Joints

Rikke Ellingsen

Civil and Environmental Engineering
Submission date: June 2013
Supervisor: Kjell Magne Mathisen, KT

Norwegian University of Science and Technology
Department of Structural Engineering

Department of Structural Engineering
Faculty of Engineering Science and Technology
NTNU- Norwegian University of Science and Technology

ACCESSIBILITY: OPEN

MASTER THESIS 2013

SUBJECT AREA:	DATE:	NO. OF PAGES:
Computational Mechanics	6.6.2013	119

TITLE:

Nonlinear Isogeometric Analysis vs NFEA of Tubular Joints

Ikkelineær isogeometrisk analyse versus NFEA av rørknutepunkt

BY:

Rikke Ellingsen

SUMMARY:

In 2005, Hughes *et al.* introduced the isogeometric analysis. One purpose was to eliminate the conversion between geometry model and analysis model in finite element analyses. NURBS (Non-Uniform Rational B-Splines) were adopted as shape functions and the isoparametric concept was utilized, resulting in the above mentioned analysis method.

In this thesis, the differences between traditional finite element analysis and isogeometric analysis have been examined through nonlinear analyses of a gap K-joint subjected to prescribed displacements. The K-joint has been modelled both with solid and thin shell elements in Abaqus/Standard and with solid elements in IFEM. It has been focused on obtaining a mesh of similar refinement for both of the methods to easier be able to compare the results.

The results show, as expected, that the thin shell element representation is unsuitable for a three dimensional stress state as around the intersection of the braces and the chord. The analyses with solid elements show a dependence on the continuity of the shape functions. The continuity in an isogeometric analysis is C^{p-1} over element borders for basis functions of degree p. In traditional finite element analysis the continuity for solid elements is only C^0. This results in differences in the differentiated variables like stresses and strain. Also the calculation of derived values, such as the von Mises stress and principal stresses, has shown to generate differences in the results because this is done differently in Abaqus/Standard and IFEM. The computational time for the isogeometric analyses is higher than for the traditional finite element analyses, and increasing with increasing degree of basis functions because the continuity affects the average bandwidth of the global matrices, and must also be included in the discussions.

The conclusion is that more analysis results are needed to be able to make a substantiated conclusion as to which analysis method that is preferable for analyzing a gap K-joint.
Regarding which method that is more accurate or conservative, the findings are not consistent.

RESPONSIBLE TEACHER: Kjell Magne Mathisen

SUPERVISOR(S): Kjell Magne Mathisen

CARRIED OUT AT: Department of Structural Engineering, NTNU

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY NTNU
FACULTY OF ENGINEERING SCIENCES AND TECHNOLOGY
DEPARTMENT OF STRUCTURAL ENGINEERING

TKT4915 Beregningsmekanikk, masteroppgave

Masteroppgave 2013

for

Rikke Ellingsen

Ikkelineær isogeometrisk analyse versus NFEA av rørknutepunkt

Nonlinear Isogeometric Analysis vs NFEA of Tubular Joints

The engineering process starts with designers encapsulating their perception using Computer Aided Design (CAD) tools. Thus, the geometry described in CAD systems is to be considered *exact* in the sense that it represents the projection from the designers perception of the desired object onto an electronic description. Today most CAD systems use spline basis function and often Non-Uniform Rational B-Splines (NURBS) of different polynomial order.

During and after the design stages different levels of numerical simulations are performed on the object. Very often this involves using the Finite Element Method (FEM) where the geometry is represented by piecewise low order polynomials. This practise introduce either significant approximation errors or fine FEM models with a large number of finite elements that make the numerical simulation computational costly. Furthermore, a huge amount of man-hours have to be spent in order to *remodel* the object into a suitable finite element mesh. This information transfer between models suitable for design (CAD) and analysis (FEM) is considered being a severe bottleneck in industry today.

To address this issue Hughes *et al*. (2005) introduced an analysis framework in which the exact CAD geometry is represented by NURBS surfaces. Throughout, the isoparametric philosophy is invoked, that is, the solution space for dependent variables is represented in terms of the same functions which represent the geometry. For this reason, they denote it isogeometric analysis.

The purpose of this master thesis is to provide a review of how isogeometric analysis compares to finite element analysis of tubular joints when it comes to formulation, implementation and usage. The report should provide a review of the isogeometric concept in general and emphasize on the formulation when applied to solving nonlinear problems involving both material and geometrical nonlinearities. In particular aspects of nonlinear behavior of tubular joints should be numerically investigated and results obtained with traditional nonlinear finite element analysis should be compared to results obtained with isogeometric analysis. The study should emphasize theory and formulation of finite deformation problems in solid and structural mechanics when applying isogeometric analysis.

The master thesis should be organized as a research report. It is emphasized that clarity and structure together with precise references are central requirements in writing a scientific report.

Responsible teacher and supervisor: Kjell Magne Mathisen

The master thesis should be handed in at the Department of Structural Engineering within June 10, 2013.

NTNU, January 25, 2013
Kjell Magne Mathisen
Responsible teacher and supervisor

Abstract

In 2005, Hughes *et al.*[1] introduced the isogeometric analysis. One purpose was to eliminate the conversion between geometry model and analysis model in finite element analyses. NURBS (Non-Uniform Rational B-Splines) were adopted as shape functions and the isoparametric concept was utilized, resulting in the above mentioned analysis method.

In this thesis, the differences between traditional finite element analysis and isogeometric analysis have been examined through nonlinear analyses of a gap K-joint subjected to prescribed displacements. The K-joint has been modelled both with solid and thin shell elements in Abaqus/Standard and with solid elements in IFEM. It has been focused on obtaining a mesh of similar refinement for both of the methods to easier be able to compare the results.

The results show, as expected, that the thin shell element representation is unsuitable for a three dimensional stress state as around the intersection of the braces and the chord. The analyses with solid elements show a dependence on the continuity of the shape functions. The continuity in an isogeometric analysis is C^{p-1} over element borders for basis functions of degree p. In traditional finite element analysis the continuity for solid elements is only C^0. This results in differences in the differentiated variables like stresses and strain. Also the calculation of derived values, such as the von Mises stress and principal stresses, has shown to generate differences in the results because this is done differently in Abaqus/Standard and IFEM. The computational time for the isogeometric analyses is higher than for the traditional finite element analyses, and increasing with increasing degree of basis functions because the continuity affects the average bandwidth of the global matrices, and must also be included in the discussions.

The conclusion is that more analysis results are needed to be able to make a substantiated conclusion as to which analysis method that is preferable for analyzing a gap K-joint. Regarding which method that is more accurate or conservative, the findings are not consistent.

Sammendrag

Isogeometrisk analyse ble for første gang introdusert i 2005 av Hughes *et al.*[1]. Én av grunnene var å eliminere prosessen med å konvertere den geometriske modellen til analysemodell ved diskretiserte elementanalyser. I arbeidet som fulgte, ble NURBS (ikke-uniforme, rasjonelle B-spline-funksjoner) tatt i bruk som formfunksjoner, og det isoparame-triske konseptet ble utnyttet. Dette førte til analysemetoden som nå er kjent som isogeometrisk analyse.

I denne oppgaven har jeg sett på forskjellen mellom tradisjonell elementanalyse og isogeometrisk analyse gjennom ikke-lineær analyse av et K-rørknutepunkt. Rørknutepunktet er modellert med både volumelementer og tynnskallelementer i Abaqus/Standard samt volumelementer i IFEM. Jeg har fokusert på å oppnå en mest mulig lik forfining av det diskrete elementmeshet for best mulig å kunne sammenligne resultatene.

Resultatene viser som forventet at tynne skallelementer ikke egner seg til å modellere den tredimensjonale spenningstilstanden i skjæringspunktet mellom rørene. Resultatene fra modellene med volumelementer i Abaqus/Standard og IFEM viser at forskjellig kontinuitet i løsningen har mye å si. Isogeometrisk analyse har C^{p-1}-kontinuitet mellom elementer i analyser med basisfunksjoner av grad p , mens volumelementer i standard elementanalyse kun er C^0-kontinuerlige. Dette resulterer i store forskjeller i de deriverte variablene som spenningene og tøyningene mellom de to metodene. Også beregningene av utlede verdier som von Mises-spenninger og hovedspenninger, har skapt forskjeller mellom de to analysemetodene fordi dette gjøres forskjellig i Abaqus/Standard og IFEM. Beregningstiden for den isogeometriske analysen er generelt høyere enn for tradisjonell elementmetode fordi høyere kontinuitet øker den gjennomsnittlige båndbredden til de globale matrisene, og må også tas med i vurderingen av de to metodene.

Diskusjonene med at det er behov for flere analyseresultater for å ha nok materiale til å ta en beslutning om hvilken metode som er å foretrekke ved analysering av et K-rørknutepunkt. Funnene er ikke konsekvente når det kommer til hvilken metode som er mest nøyaktig eller konservativ.

Preface

This master thesis has been written as the final evaluation of the master's degree in Civil and Environmental Engineering at Norwegian University of Science and Technology (NTNU), spring 2013. The area of specialization has been within computational mechanics at Department of Structural Engineering (KT).

The thesis has been conducted partly as a literature study of nonlinear finite element analyses and a continuation of the literature study carried out in the specialization project completed fall 2012 on isogeometric analysis, and partly as discussion and interpretation of analysis results. The analyses have been run with Abaqus/Standard 6.11-1 and IFEM, and it has been aspired to analyse as similar models as possible for both the analysis methods.

My experience with IFEM is that a substantial amount of development work still remains to be done before it is suitable for isogeometric analyses of industrial problems involving more complex geometry. The initially proposed geometries for this thesis could not be modelled, and the modelling of the gap K-joint was too complex making it impossible for me to learn within the limited time frame that was spent working on this thesis, and the definition of the model and the analyses must be run by scientists already familiar with the software.

Acknowledgements should be made to my principal advisor Kjell Magne Mathisen, professor at KT, for excellent guidance and motivation, and with thorough help with the analyses run in IFEM. Kjetil Andre Johannessen, PhD Candidate at Department of Mathematical Sciences, and Knut Morten Okstad, Research Scientist at SINTEF, Department of Applied Mathematics, should also be acknowledged for providing the files necessary to run the IFEM analyses.

In the end, a big thank you goes to my uncle Inge Fjulsrud for providing his excellent knowledge of the English language to proofread my thesis.

Contents

Notation

In this master thesis the notation given below will be used for matrices of varying size.

Name	Notation	Matrix form
Column matrix	$\{\boldsymbol{A}\}$	$\begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{Bmatrix}$
Row matrix	$\lfloor \boldsymbol{A} \rfloor$	$\lfloor a_1 \quad a_2 \quad \cdots \quad a_n \rfloor$
Symmetric or asymmetric matrix	$[\boldsymbol{A}]$	$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$
Diagonal matrix	$\lceil \boldsymbol{A} \rfloor$	$\begin{bmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,n} \end{bmatrix} =$ $\left\lceil a_{1,1} \quad a_{2,2} \quad \cdots \quad a_{n,n} \right\rfloor$

Chapter 1

Introduction

The conversion between the geometry model and the analysis model in finite element analyses (FEA) has been studied for many years. The traditional way of performing an analysis is to define a geometric model in a computer aided design (CAD) tool, and thereafter analysing the model in a computer aided engineering (CAE) analysis program. The difficulties in this conversion often lies within maintaining the exact geometric properties given in the geometric model. This is a result of the CAD and CAE tools using different shape functions to define the geometry and the discretized model respectively. As of now, the most common basis functions in CAD tools are *NURBS (Non-Uniform Rational B-Splines)*, whereas analysis tools generally use lower order *Lagrange polynomials*. As a result, 80 % of the overall analysis time[1] is usually spent on first creating the geometric model and thereafter meshing the analysis model to get a good approximation of the geometry [2].

In 2005, Hughes *et al.* [1] presented the concept of *IsoGeometric Analysis (IGA)*. The main object of this representation is to use the same model to define the geometry and to perform analyses. To achieve this, the analysis must utilize the same basis functions as the CAD tools used to define the geometry. The geometry technology that is the most common, and which will be described in this thesis is NURBS, and consequently also *B-splines*. The basic theory for lower order Lagrange polynomials and NURBS will be discussed, with focus on solid and shell representation for the traditional analysis and solid representation for the IGA. The area of focus will be the representation of shells by solid and shell elements respectively, and the difference between a solid model in traditional FEA and IGA. A gap K-joint has been modelled and analyzed to exemplify occurring differences.

The software used to perform the analyses is the well-known and well-established FEA program Abaqus/Standard 6.11-1, and the newer and non-established isogeometrically based analysis program IFEM. Abaqus/Standard is a product of Dassault Systèmes Simulia Corp., Providence, RI, USA. It is based on the traditional Finite Element Method (FEM), employing shape functions constructed from amongst other lower order Lagrange polynomials and serendipity functions[2] in addition to traditional meshing and refinement

[1] The anatomy of the process of creating the appropriate geometry for analysis models is studied by Ted Blacker, Manager of Simulation Sciences, Sandia National Laboratories, [2].

[2] Other options are i.e. Hermite polynomials and mixed formulations.

techniques. IFEM is a framework using NURBS to perform linear and nonlinear IGAs of solid problems. IFEM is developed as part of the KMB (Knowledge-building projects with user involvement) research project ICADA (Integrated Computer Aided Design and Analysis) which is a collaboration project between Department of Applied Mathematics at SINTEF, Department of Mathematical Sciences and Department of Structural Engineering (KT) at Norwegian University of Science and Technology (NTNU), Ceetron AS, unit for Research & Innovation at Det Norske Veritas (DNV) and Statoil, with the purpose of developing a suitable alternative to the traditional finite element programs by utilizing spline finite elements. The main objective of the ICADA project is to *build competence in Norway for taking advantages of Integrated Computer Aided Design and Analysis based on a coherent representation for geometry and analysis* [3]. Throughout the thesis, IFEM is not as thoroughly discussed as Abaqus/Standard. This is a result of the current limited documentation of IFEM, which has made details of its manner of operation partly unavailable for me. It has also increased my dependence of professionals with knowledge about IFEM to be able to run the analyses in IFEM. To visualize the results from IFEM, GLview Inova 9.0 has been used. *GLview Inova is a modern, fully featured post-processor and 3D visualization tool* [4], which makes post-processing from IFEM easier.

Chapter 2

Traditional Finite Element Analysis

A FEA consists of discretizing the structure so that element matrices can be assembled in a local manner. The deformation of each element is calculated before being globalized, and thereby representing the solution field of the structure.

2.1 Discretization

2.1.1 Mesh

In a FEA, the structure is divided into finite elements. The collection of elements is called a mesh, and one or several meshes span the entire structure. In each node, which is located in each elements, there are a given number of degrees of freedom. These are the values that are calculated in a FEA. The interpolation function used to describe the geometry and the solution are local to only one element at a time.

To make the solution space more accurate the mesh can be refined. In a FEA this can be done in three ways [5]. *h-refinement* consists of adding elements of the same type to the mesh. This can be done either uniformly through the entire structure, or non-uniformly in the areas where the accuracy is unsatisfactory. If the degree of the highest complete polynomial is increased, the refinement method is called *p-refinement*. With p-refinement, the number of elements is not changed, but the number of degrees of freedom can be increased with or without adding new nodes. The last refinement method is called *r-refinement*. This refinement method consists of rearranging nodes so that the element geometry changes. Examples of the different methods are shown in Figure 2.1.

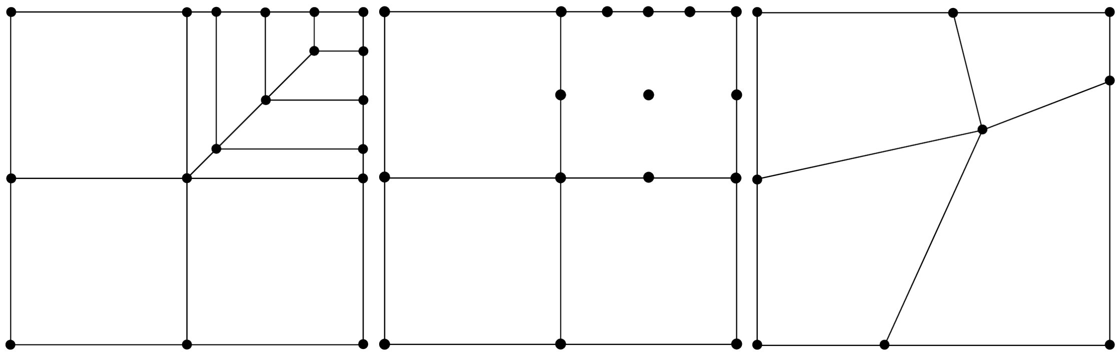
2.1.2 Degrees of Freedom

The discrete displacement vector is defined as

$$\text{locally:} \quad \{\boldsymbol{d}\} = \{d_j^a\}, \tag{2.1}$$

$$\text{globally:} \quad \{\boldsymbol{D}\} = \{D_j^A\}, \tag{2.2}$$

where j is the number of degrees of freedom in each node, a and A are the numbering of the nodes in one element and the entire structure respectively, and $a = 1, 2, \cdots, n_e$



(a) h-refinement **(b)** p-refinement **(c)** r-refinement

Figure 2.1: Refinement methods in FEA

and $A = 1, 2, \cdots, n_G$ for each $j = 1, 2, \cdots, n_{dof}$. The degrees of freedom are located in the nodes and are interpolated according to the shape functions derived in Section 2.2 to express the total solution of the structure.

Solid

The definition of the degrees of freedom for a structure modelled with solid elements is

$$u_j = \sum_a N_a d_j^a. \tag{2.3}$$

The physical coordinates are also interpolated between the nodes according to the shape functions derived in Section 2.2. This is called the *isoparametric concept*, and applies to both the initial and current coordinates, X_J and x_j.

$$X_J = \sum_a N_a X_J^a \tag{2.4}$$

$$x_j = \sum_a N_a x_j^a \tag{2.5}$$

The element described in this thesis will be a 20-node quadratic brick element, with degrees of freedom being translations, u, v and w, in the three orthogonal directions, ξ, η and ζ. It is up to the analyst to decide the order of the degrees of freedom. The options can be

$$\{\boldsymbol{d}\} = \lfloor u^1 \quad \cdots \quad u^{n_e} \quad v^1 \quad \cdots \quad v^{n_e} \quad w^1 \quad \cdots \quad w^{n_e} \rfloor^T, \tag{2.6}$$

$$\{\bar{\boldsymbol{d}}\} = \lfloor u^1 \quad v^1 \quad w^1 \quad u^2 \quad v^2 \quad w^2 \cdots \quad u^{n_e} \quad v^{n_e} \quad w^{n_e} \rfloor^T, \tag{2.7}$$

depending on the purpose of the calculations. There are 8 corner nodes and 12 midside nodes, as shown in Figure 2.2. The faces do not need to be plane and the corners may have arbitrary location relative to each other. The element resists deformation due to strain energy from both translational and transverse deformation, but have no rotational degrees of freedom, and the edges of the element are thus free to rotate. The element does not exhibit shear locking, which may be caused if excessive shear strain evolve when pure bending is modelled [5].

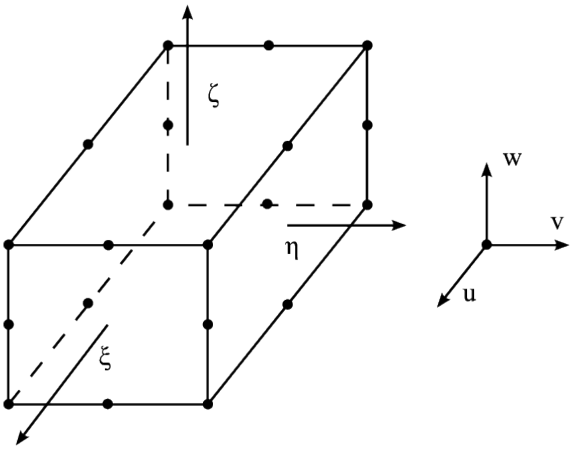


Figure 2.2: 20-node solid element with three degrees of freedom in each node

Shell

The shell element formulation chosen in this thesis is the *Kirchhoff thin shell* representation. A thin shell representation can be used if the ratio between the thickness and other properties of the shell is below a given limit. The limit differs from the thickness/smallest radii ratio being below 1/20 [5] to the ratio between the thickness and a characteristic length of the element surface to be below 1/15 [6], and indicates that each case must be assessed individually. Another option would have been the *Mindlin thick shell theory*, which takes into account the transverse shear deformations. Thin shell theory is based on the idea that transverse shear deformation is prohibited. This implies that the angle between the mid surface and any section of the shell remains constant during deformation. The element described in this thesis has translation in ξ-, η- and ζ-direction and rotation about the ξ- and η-axis respectively as degrees of freedom, and is an 8-node serendipity element, shown in Figure 2.3.

$$\{\boldsymbol{d}\} = \lfloor u^1 \quad \cdots \quad u^{n_e} \quad v^1 \quad \cdots \quad v^{n_e} \quad w^1 \quad \cdots \quad w^{n_e} \quad \psi_\xi^1 \quad \cdots \quad \psi_\xi^{n_e} \quad \psi_\eta^1 \quad \cdots \quad \psi_\eta^{n_e} \rfloor^T, \tag{2.8}$$

$$\{\bar{\boldsymbol{d}}\} = \lfloor u^1 \quad \cdots \quad \psi_\eta^1 \quad u^2 \quad \cdots \quad \psi_\eta^2 \quad \cdots \quad u^{n_e} \quad \cdots \quad \psi_\eta^{n_e} \rfloor^T. \tag{2.9}$$

In contrast to the solid representation, the expression for deformation in an arbitrary point in the shell in general also includes rotational degrees of freedom [5]. Figure 2.3 shows the definition of the parametric axes of a shell element, and point P with midsurface coordinates corresponding to a node, but with arbitrary location in the thickness. As a result, dependence of the rotational degrees of freedom for the deformations of point P arises, and the total deformations may be expressed as

$$u_{\bar{j}} = \sum_a N_a \left(u_{\bar{j}}^a + \zeta \frac{t_a}{2} [\boldsymbol{\mu}_{\bar{j}}^a] \begin{Bmatrix} \psi_\xi^a \\ \psi_\eta^a \end{Bmatrix} \right), \tag{2.10}$$

for $\bar{j} = 1, 2, 3$. t_a is the thickness of the shell at node a, and $[\boldsymbol{\mu}_{\bar{j}}^a]$ is a direction cosine

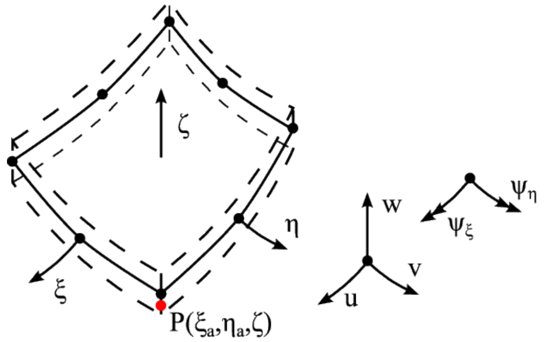


Figure 2.3: 8-node general shell with five degrees of freedom in each node

matrix derived from the two first covariant base vectors described in Section 4.2:

$$[\boldsymbol{\mu}^a_{j}] = \left[-\frac{\{\boldsymbol{g}^a_\eta\}_{\bar{j}}}{|\{\boldsymbol{g}^a_\eta\}|} \quad \frac{\{\boldsymbol{g}^a_\xi\}_{\bar{j}}}{|\{\boldsymbol{g}^a_\xi\}|} \right] = \begin{bmatrix} -l_{\eta a} & l_{\xi a} \\ -m_{\eta a} & m_{\xi a} \\ -n_{\eta a} & n_{\xi a} \end{bmatrix}, \tag{2.11}$$

$\{\boldsymbol{g}^a_i\}_{\bar{j}}$ being the $\bar{j}$'th component of the covariant base vectors in node a respectively. The stiffness of the shell is a combination of bending and membrane stiffness. The bending stiffness is in general much smaller than the membrane stiffness and (especially with straight edges) drastically decreasing with increasing element length. Shell elements with curved edges may exhibit membrane locking. Membrane locking causes deformations initially only resisted by bending stiffness also to be resisted by membrane stiffness [5].

The initial and current coordinates of the point P are calculated in the same manner as the displacements:

$$X_J = \sum_a N_a \left(X^a_J + \zeta \frac{t_a}{2} \{\boldsymbol{\Lambda}^a\}_J \right), \tag{2.12}$$

$$x_j = \sum_a N_a \left(x^a_j + \zeta \frac{t_a}{2} \{\boldsymbol{\lambda}^a\}_j \right). \tag{2.13}$$

$\{\boldsymbol{\Lambda}^a\}_J$ and $\{\boldsymbol{\lambda}^a\}_j$ are the J'th and j'th components of the direction cosine matrices of the third covariant base vector in initial and current configuration in node a [1]

$$\{\boldsymbol{\Lambda}^a\}_J = \frac{\{\boldsymbol{G}^a_Z\}_j}{|\{\boldsymbol{G}^a_Z\}|}, \quad \{\boldsymbol{\lambda}^a\}_j = \frac{\left\{\boldsymbol{g}^a_\zeta\right\}_j}{|\left\{\boldsymbol{g}^a_\zeta\right\}|}. \tag{2.14}$$

2.2 Interpolation Functions

As seen in the previous section, the interpolation functions are used to interpolate both the displacement and the geometry between the nodal points.

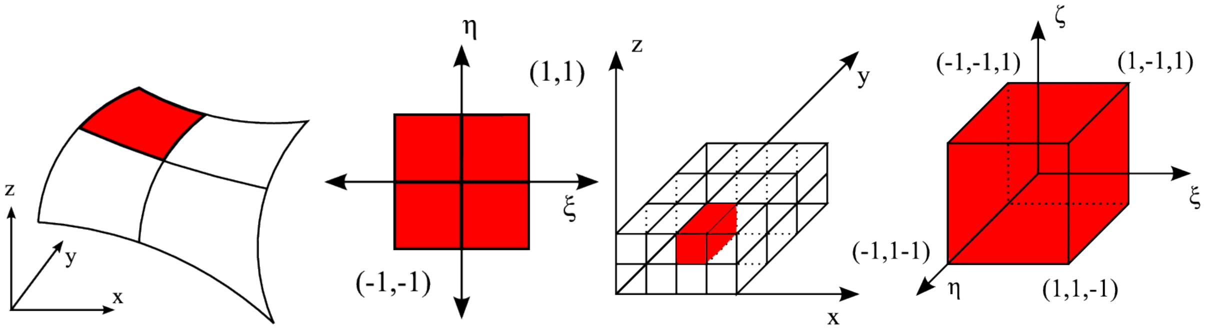
[1] Z represents here the capital ζ

The interpolation functions in one direction are usually constructed by Lagrange polynomials by the formula

$$N_a = \prod_{\substack{0<k\leq n_e \\ k\neq a}} \frac{x_k - x}{x_k - x_a}, \tag{2.15}$$

where $k = 1, 2, \cdots, n_e$. Hence we see that an n_e-noded element with only translational degrees of freedom requires a shape function of degree $n_e - 1$. If also the derivatives of the displacement are degrees of freedom, the degree of the interpolation functions can be decreased with the order of the derivatives. For a shell element, also including the rotational degrees of freedom the Hermite interpolation functions are used [7].

The shape functions are often expressed by parametric coordinates in the sense that $x = x(\xi)$, $y = y(\eta)$ and $z = z(\zeta)$.Each physical element is then mapped from the physical to the parameter space as in Figure 2.4. The relation between the derivatives of the physical and the parametric coordinates is function of the *Jacobian matrix*, $[\boldsymbol{J}]$:



(a) Shell **(b)** Solid

Figure 2.4: Physical and parent element in FEA

$$\begin{Bmatrix} \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\zeta} \end{Bmatrix} = [\boldsymbol{J}] \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} = \begin{bmatrix} x_{,\xi} & y_{,\xi} & z_{,\xi} \\ x_{,\eta} & y_{,\eta} & z_{,\eta} \\ x_{,\zeta} & y_{,\zeta} & z_{,\zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix}. \tag{2.16}$$

The expression $x_{,\xi}$ denotes the derivative of x w.r.t. ξ and so on. For the solid representation the Jacobian matrix yields

$$[\boldsymbol{J}] = \begin{bmatrix} \sum_a N_{a,\xi}x_a & \sum_a N_{a,\xi}y_a & \sum_a N_{a,\xi}z_a \\ \sum_a N_{a,\eta}x_a & \sum_a N_{a,\eta}y_a & \sum_a N_{a,\eta}z_a \\ \sum_a N_{a,\zeta}x_a & \sum_a N_{a,\zeta}y_a & \sum_a N_{a,\zeta}z_a \end{bmatrix}, \tag{2.17}$$

and for shell:

$$[\boldsymbol{J}] = \begin{bmatrix} \sum_a N_{a,\xi}(x_a + \zeta t_a l_{\zeta a}/2) & \sum_a N_{a,\xi}(y_a + \zeta t_a m_{\zeta a}) & \sum_a N_{a,\xi}(z_a + \zeta t_a n_{\zeta a}) \\ \sum_a N_{a,\eta}(x_a + \zeta t_a l_{\zeta a}/2) & \sum_a N_{a,\eta}(y_a + \zeta t_a m_{\zeta a}) & \sum_a N_{a,\eta}(z_a + \zeta t_a n_{\zeta a}) \\ \sum_a N_a t_a l_{\zeta a}/2 & \sum_a N_a t_a m_{\zeta a}/2 & \sum_a N_a t_a n_{\zeta a}/2 \end{bmatrix}. \tag{2.18}$$

This gives the expression for the derivative of the displacements

$$\begin{Bmatrix} u_{,x} \\ u_{,y} \\ u_{,z} \\ v_{,x} \\ v_{,y} \\ v_{,z} \\ w_{,x} \\ w_{,y} \\ w_{,z} \end{Bmatrix} = \begin{bmatrix} \mathbf{\Gamma} & 0 & 0 \\ 0 & \mathbf{\Gamma} & 0 \\ 0 & 0 & \mathbf{\Gamma} \end{bmatrix} \begin{Bmatrix} u_{,\xi} \\ u_{,\eta} \\ u_{,\zeta} \\ v_{,\xi} \\ v_{,\eta} \\ v_{,\zeta} \\ w_{,\xi} \\ w_{,\eta} \\ w_{,\zeta} \end{Bmatrix}, \tag{2.19}$$

where

$$[\mathbf{\Gamma}] = [\boldsymbol{J}]^{-1}. \tag{2.20}$$

The discretization can also be done by generalized coordinates, and for triangles; area coordinates, but this will not be discussed in this thesis. Description of these methods can be found in [5].

2.2.1 Requirements to the Shape Functions

The interpolation functions chosen have to fulfill two requirements [5]:

– completeness
– compatibility

The *completeness* requirement states that every shape function must be able to describe a constant value of the field variable itself and its derivatives. This implies that when choosing the interpolation function, each polynomial degree has to be complete so that the shape functions display partition of unity:

$$\sum_{a=1}^{n_e} N_a = 1. \tag{2.21}$$

Introducing ξ^p, η^q and ζ^r, it is necessary to include all elements with $p+q+r=1$ before including elements with $p+q+r=2$, and all elements with $p+q+r=2$ before elements with $p+q+r=3$ and so on. This requirement ensures the ability to express rigid body motion.

Compatibility controls continuity of the elements. The total amount of energy utilised by the virtual work are taken as the sum of the energy within each element. This is done by integration over the element boundaries:

$$(\delta W)_{el} = \int_{el} \{\delta \boldsymbol{\epsilon}\}^T \{\boldsymbol{\sigma}\}\, dv. \tag{2.22}$$

The continuity requirement states that there has to be $C^{\mu-1}$ continuity between elements. Here μ represents the highest degree of the differential operator in the strain-displacement relation. For solid elements it will been shown that the strains are functions of the first derivative of the displacements, as in Section 4.8. This implies that the actual value of the displacements has to be continuous, thus C^0 continuity is required. As the structure is defined with thin shell elements, the rotational degrees of freedom may be expressed as derivatives of the translational degrees of freedom, and the differential operator thus includes derivatives of 2nd degree. The continuity requirement of thin shells is therefore C^1 which means that both the translational and rotational degrees of freedom must be continous over element borders.

2.2.2 Example of Interpolation Functions

An example of interpolation functions can be shown by taking the element in Figure 2.5 into account. The element has four nodes, assumed with only translational degrees of freedom, which indicates that four third order Lagrange polynomials have to be constructed. Stating the spacing of the nodes to be uniform and the use of parametric coordinates gives the shape functions:

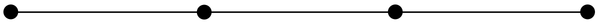


Figure 2.5: 1D finite element with four nodes

$$N_1 = \frac{1}{16}\left(-1 + \xi + 9\xi^2 - 9\xi^3\right), \tag{2.23}$$

$$N_2 = \frac{9}{16}\left(1 - 3\xi - \xi^2 + 3\xi^3\right), \tag{2.24}$$

$$N_3 = \frac{9}{16}\left(1 + 3\xi - \xi^2 - 3\xi^3\right), \tag{2.25}$$

$$N_4 = \frac{1}{16}\left(-1 - \xi + 9\xi^2 + 9\xi^3\right). \tag{2.26}$$

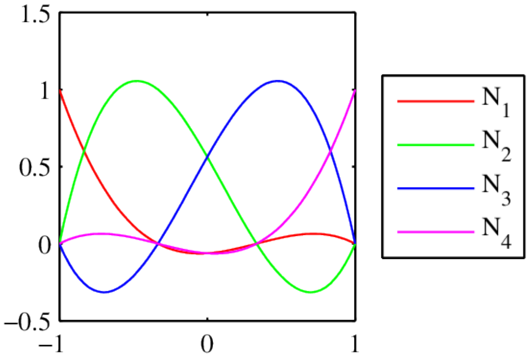


Figure 2.6: Example of interpolation functions

Chapter 3

Isogeometric Analysis

3.1 B-Splines

The shape functions of an IGA can amongst others be NURBS, and to get an understanding of what a NURBS is, we have to look at B-splines, considering that NURBS are a generalization of B-splines. Therefore, all theory in this section also applies to Section 3.2. B-splines are non-rational piecewise functions, divided by knot vectors and constructed by basis functions.

3.1.1 Knot Vectors

The B-spline is divided into polynomials by parametric knot vectors, which in three parametric orthogonal directions are defined as

$$\boldsymbol{\Xi} = \lfloor \xi_1, \xi_2, .., \xi_i, .., \xi_{N-1}, \xi_N \rfloor, \tag{3.1a}$$

$$\boldsymbol{H} = \lfloor \eta_1, \eta_2, .., \eta_j, .., \eta_{M-1}, \eta_M \rfloor, \tag{3.1b}$$

$$\boldsymbol{Z} = \lfloor \zeta_1, \zeta_2, .., \zeta_k, .., \zeta_{L-1}, \zeta_L \rfloor. \tag{3.1c}$$

This thesis will only derive the concept in one direction, but all of the below derivations are also valid for the two other directions.

The number of elements in the knot vector is always

$$N = n + p + 1, \tag{3.2}$$

where n is the number of basis functions used to construct the B-spline, and p is the degree of the basis functions considered. Either of these values can be determined first to get the best result of the analysis.

If the multiplicity of the first and last knot is $p + 1$, the knot vector is said to be *open* and the number of multiplicity determines the degree of the basis functions. We will only look at open knot vectors in this thesis. At the end of an open knot vector the spline is interpolated, which means that it is only the first and last basis functions that are non-zero at the first and last knot respectively. The result of this is that the spline always is tangential to the control polygon, see Section 3.1.3, at the beginning and at the end of an open knot vector, which ensures that the spline is correct at the beginning and at the end.

The mapping of the knot vectors onto the physical geometry is what creates the element boundaries in the physical space.

3.1.2 Basis Functions

The basis functions are functions of the parametric coordinates, ξ, η and ζ, and are constructed by the the same method in all the directions.

$$N_{a,0}(\xi) = \begin{cases} 1 & \text{if } \xi_a \leq \xi < \xi_{a+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

where a is the knot span number, $a = 1, 2, ..., N-1$.

For $p = 1, 2, 3, ...$ they are defined as

$$N_{a,p}(\xi) = \frac{\xi - \xi_a}{\xi_{a+p} - \xi_a} N_{a,p-1}(\xi) + \frac{\xi_{a+p+1} - \xi}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1,p-1}(\xi). \tag{3.4}$$

This is called the Cox-deBoor recursion formula [2].

The first derivative of a basis function is

$$N'_{a,p}(\xi) = \frac{p}{\xi_{a+p} - \xi_a} N_{a,p-1}(\xi) - \frac{p}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1,p-1}(\xi). \tag{3.5}$$

Only the first two derivative basis functions are non-zero in the first knot. This is utilized when deriving the criteria for continuity.

Important Features of the Basis Functions

There are several significant features of the basis functions that are important to notice.

One is that the basis functions constitute a partition of unity within each degree,

$$\sum_{a=1}^{n} N_{a,p}(\xi) = 1. \tag{3.6}$$

This is necessary to be able to express rigid body displacement.

Another feature is that the basis functions are point-wise non-negative,

$$N_{a,p}(\xi) > 0 \quad \forall \xi, \tag{3.7}$$

which is utilized when deriving the mass matrix.

A third feature is that each basis function of degree p is C^{p-1} continuous across element boundaries. This is very important in element modelling. The ability to reduce continuity is also a useful feature when it is necessary to deal with singular points on a B-spline. A basis function is C^{p-s} continuous at knots that have multiplicity s.

It is also worth noticing that the basis functions have support on $p+1$ knot spans, regardless of the extent of the knot span in the parametric or physical space. A disadvantage related to this property is that the system of equations obtain a large bandwidth which increases the computation time. An advantage is that the functions get better continuity given that the basis functions are C^∞ continuous inside knot spans and C^{p-s} continuous at knots.

Basis functions are not interpolative and the value in each knot depends on several basis functions. To reduce the number of knots that a basis function is influencing, knots can be multiplied. This gives, however, the possible disadvantage of reduced continuity.

Requirements to the Basis Functions

As seen for the interpolation functions in Section 2.2, the basis functions also have to satisfy certain criteria to ensure convergence. The criteria are:

- C^1 within knot spans and across knots
- C^0 between patches (see 3.3.2)
- completeness

The continuity requirement for each element is not restrictive. This is implied by the feature of C^∞ continuity inside knot spans and C^{p-s} continuity over the element boundaries, which can fulfil C^1 for all splines with $p \geq 2$. The completeness requirement is explained in Section 2.2.1, and applies to the basis functions as stated in Equation (3.4).

Example of Basis Functions Constructed from an Open Knot Vector

To give a better understanding of how the basis functions are constructed, the open knot vector $\Xi = [\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \xi_7] = [0, 0, 0, 2, 4, 4, 4]$ will be considered.

The multiplicity of the first and last knot is three, hence the highest degree of the basis functions is two. By applying Equation (3.2), the number of basis functions that need to be used is six, five and four for $p = 0, 1, 2$ respectively.

Starting with $a = 1, p = 0$ we see from Equation (3.3) that $N_{1,0}$ is defined as zero because the two first objects in the knot vector are both zero, and ξ can therefore never be in between these two values. The same applies to $N_{2,0}$, $N_{5,0}$ and $N_{6,0}$, because the first three and last three objects in the knot vector are multiplicities of one knot.

Putting $a = 3, p = 0$ gives $\xi_a = 0$ and $\xi_{a+1} = 2$ and results in the basis function shown in Equation (3.8).

$$N_{3,0}(\xi) = \begin{cases} 1 & \text{if } 0 \leq \xi < 2, \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

$a = 4$ and $p = 0$ gives $\xi_a = 2$ and $\xi_{a+1} = 4$ and

$$N_{4,0}(\xi) = \begin{cases} 1 & \text{if } 2 \leq \xi < 4, \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

Equation (3.10) shows derivation of $N_{1,1}$:

$$N_{1,1}(\xi) = \frac{\xi - 0}{0 - 0} N_{1,0}(\xi) + \frac{0 - \xi}{0 - \xi_{a+1}} N_{2,0}(\xi) = 0 \tag{3.10}$$

$N_{1,0}$ is as shown above equal to zero and the expression $\frac{0}{0}$ is by definition equal to zero which gives $N_{1,1}$ the value zero.

$N_{2,1}$, $N_{3,1}$, $N_{4,1}$ and $N_{5,1}$ are derived in Equations (3.11) through (3.14) and show the linearity of the basis functions of degree one.

$$\begin{aligned} N_{2,1}(\xi) &= \frac{\xi - 0}{0 - 0} N_{2,0}(\xi) + \frac{2 - \xi}{2 - 0} N_{3,0}(\xi) \\ &= 0 + \left(1 - \frac{\xi}{2}\right) N_{3,0}(\xi) = \begin{cases} 1 - \frac{1}{2}\xi & \text{if } 0 \le \xi < 2, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.11}$$

$$\begin{aligned} N_{3,1}(\xi) &= \frac{\xi - 0}{2 - 0} N_{3,0}(\xi) + \frac{4 - \xi}{4 - 2} N_{4,0}(\xi) \\ &= \frac{\xi}{2} N_{3,0}(\xi) + \left(2 - \frac{\xi}{2}\right) N_{4,0}(\xi) = \begin{cases} \frac{1}{2}\xi & \text{if } 0 \le \xi < 2, \\ 2 - \frac{1}{2}\xi & \text{if } 2 \le \xi < 4, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.12}$$

$$\begin{aligned} N_{4,1}(\xi) &= \frac{\xi - 2}{4 - 2} N_{4,0}(\xi) + \frac{4 - \xi}{4 - 4} N_{5,0}(\xi) \\ &= \left(\frac{\xi}{2} - 1\right) N_{4,0}(\xi) = \begin{cases} -1 + \frac{1}{2}\xi & \text{if } 2 \le \xi < 4, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.13}$$

$$N_{5,1}(\xi) = \frac{\xi - 4}{4 - 4} N_{5,0}(\xi) + \frac{4 - \xi}{4 - 4} N_{6,0}(\xi) = 0. \tag{3.14}$$

The basis functions of degree two are derived in Equations (3.15) through (3.18).

$$\begin{aligned} N_{1,2}(\xi) &= \frac{\xi - 0}{0 - 0} N_{1,1}(\xi) + \frac{2 - \xi}{2 - 0} N_{2,1}(\xi) \\ &= 0 + (1 - \frac{\xi}{2}) N_{2,1}(\xi) = \begin{cases} 1 - \xi + \frac{1}{4}\xi^2 & \text{if } 0 \le \xi < 2, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.15}$$

$$\begin{aligned} N_{2,2}(\xi) &= \frac{\xi - 0}{2 - 0} N_{2,1}(\xi) + \frac{4 - \xi}{4 - 0} N_{3,1}(\xi) \\ &= \frac{\xi}{2} N_{2,1}(\xi) + (1 - \frac{\xi}{4}) N_{3,1}(\xi) = \begin{cases} \xi - \frac{3}{8}\xi^2 & \text{if } 0 \le \xi < 2, \\ 2 - \xi + \frac{1}{8}\xi^2 & \text{if } 2 \le \xi < 4, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.16}$$

$$N_{3,2}(\xi) = \frac{\xi - 0}{4 - 0} N_{3,1}(\xi) + \frac{4 - \xi}{4 - 2} N_{4,1}(\xi)$$
$$= \frac{\xi}{4} N_{3,1}(\xi) + (2 - \frac{\xi}{2}) N_{4,1}(\xi) = \begin{cases} \frac{1}{8}\xi^2 & \text{if } 0 \leq \xi < 2, \\ -2 + 2\xi - \frac{3}{8}\xi^2 & \text{if } 2 \leq \xi < 4, \\ 0 & \text{otherwise,} \end{cases} \tag{3.17}$$

$$N_{4,2}(\xi) = \frac{\xi - 2}{4 - 2} N_{4,1}(\xi) + \frac{4 - \xi}{4 - 4} N_{5,1}(\xi)$$
$$= \left(\frac{\xi}{2} - 1\right) N_{4,1}(\xi) = \begin{cases} 1 - \xi + \frac{1}{4}\xi^2 & \text{if } 2 \leq \xi < 4, \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

The result is shown in Figure 3.1.

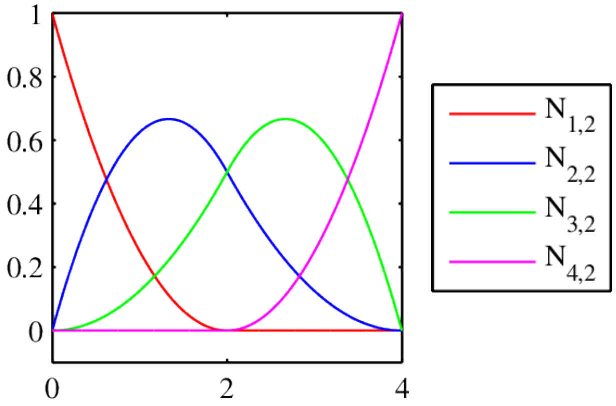


Figure 3.1: Example of basis functions

3.1.3 Control Points

To be able to use the basis functions in construction of B-splines there is need for control points. The control points are expressed as

$$\lfloor \boldsymbol{P}_a \rfloor \in \mathbb{R}^3 \tag{3.19}$$

for $a = 1, 2, ..., n_{cp}$. The control points are represented in the physical space and transform the B-splines from the parametric space into the physical space.

As seen in Equation (3.21), the number of control points must be equal to the number of basis functions used to construct the B-spline, i.e., $n_{cp} = n$ with n as in Equation (3.2). By linearly interpolating between the control points we get the control polygon. An example with

$$[\boldsymbol{P}] = \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \\ \boldsymbol{P}_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \tag{3.20}$$

is shown in Figure 3.2a. Here all z-coordinates are held constant, so the control points can be represented in the 2D space. By keeping two coordinates constant for all control points, they can be represented in the 1D space.

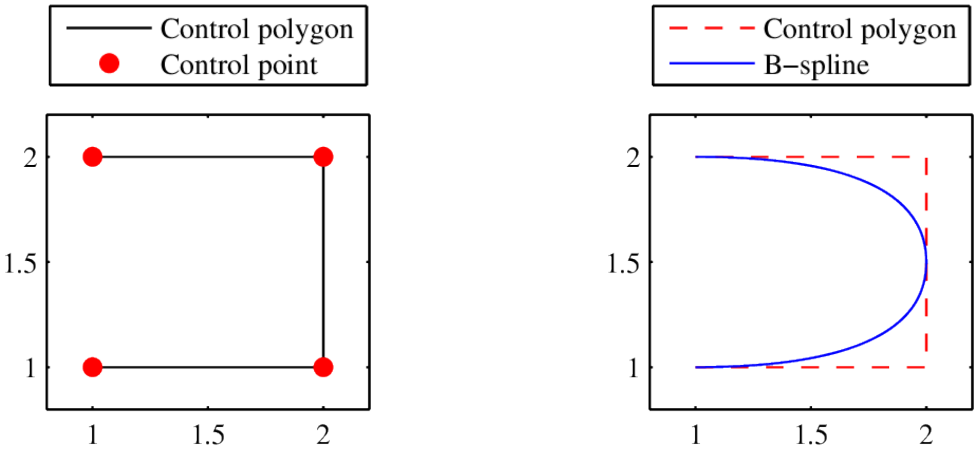
3.1.4 Construction of B-splines

B-spline curves, $\lfloor \boldsymbol{C}(\xi) \rfloor$, are generated by taking linear combinations of the basis functions and the influence of the control points

$$\lfloor \boldsymbol{C}(\xi) \rfloor = \sum_{a=1}^{n} N_{a,p}(\xi) \lfloor \boldsymbol{P}_a \rfloor = \lfloor \boldsymbol{N} \rfloor [\boldsymbol{P}]. \tag{3.21}$$

The three components of $\lfloor \boldsymbol{C}(\xi) \rfloor$ are the physical coordinates as function of the parametric coordinates. The components will be denoted $\lfloor \boldsymbol{C} \rfloor_1$, $\lfloor \boldsymbol{C} \rfloor_2$ and $\lfloor \boldsymbol{C} \rfloor_3$.

An example with the basis functions from Figure 3.1 and control points from Figure 3.2a is shown in Figure 3.2b.



(a) Control points and polygon

(b) B-spline

Figure 3.2: Example of control points and B-spline

Derivative of B-spline Curves

The derivative of a B-spline curve is

$$\lfloor \boldsymbol{C}'(\xi) \rfloor = \sum_{a=1}^{n} N'_{a,p}(\xi) \lfloor \boldsymbol{P}_a \rfloor. \tag{3.22}$$

The points of interest regarding the derivatives are the end points of the knot vectors where two B-splines join. Considering a B-spline with the knot vector and basis functions as in the example in Section 3.1.2, we get the derivative of the end points of the B-spline. In this example, the highest $p = 2$.

First,

$$\lfloor \boldsymbol{C}(0) \rfloor = \sum_{a=1}^{n} N_{a,p}(0) \lfloor \boldsymbol{P}_a \rfloor, \tag{3.23}$$

and as seen in Figure 3.1, only $N_{1,2}(0) \neq 0$ giving

$$\lfloor \boldsymbol{C}(0) \rfloor = N_{1,2}(0) \lfloor \boldsymbol{P}_1 \rfloor = \lfloor \boldsymbol{P}_1 \rfloor. \tag{3.24}$$

The first derivative of the B-spline in the starting point is

$$\begin{aligned}
\lfloor \boldsymbol{C}'(0) \rfloor &= \sum_{a=1}^{n} N'_{a,p}(0) \lfloor \boldsymbol{P}_a \rfloor & (3.25)\\
&= \sum_{a=1}^{n} \left(\frac{p}{\xi_{a+p} - \xi_a} N_{a,p-1}(0) - \frac{p}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1,p-1}(0) \right) \lfloor \boldsymbol{P}_a \rfloor. & (3.26)
\end{aligned}$$

From the example we see that the only basis function of degree $p-1 = 1$ that is not equal to zero for $\xi = 0$ is $N_{2,1}$. This gives

$$\begin{aligned}
\lfloor \boldsymbol{C}'(0) \rfloor &= \frac{p}{\xi_{a+p} - \xi_a} N_{a=2,p-1=1}(0) \lfloor \boldsymbol{P}_a \rfloor - \frac{p}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1=2,p-1=1}(0) \lfloor \boldsymbol{P}_a \rfloor & (3.27)\\
&= \frac{p}{\xi_{2+p} - \xi_2} N_{2,1}(0) \lfloor \boldsymbol{P}_2 \rfloor - \frac{p}{\xi_{1+p+1} - \xi_{1+1}} N_{2,1}(0) \lfloor \boldsymbol{P}_1 \rfloor & (3.28)\\
&= \frac{p}{\xi_{2+p} - 0} \left(1 \cdot \lfloor \boldsymbol{P}_2 \rfloor - 1 \cdot \lfloor \boldsymbol{P}_1 \rfloor \right) & (3.29)\\
&= \frac{p}{\xi_{2+p}} \left(\lfloor \boldsymbol{P}_2 \rfloor - \lfloor \boldsymbol{P}_1 \rfloor \right). & (3.30)
\end{aligned}$$

The expression is general with the assumption of open knot vectors with a first knot with the value zero and $p \geq 1$.

The first derivative of the B-spline in the end point is

$$\begin{aligned}
\lfloor \boldsymbol{C}'(4) \rfloor &= \sum_{a=1}^{n} N'_{a,p}(4) \lfloor \boldsymbol{P}_a \rfloor & (3.31)\\
&= \sum_{a=1}^{n} \left(\frac{p}{\xi_{a+p} - \xi_a} N_{a,p-1}(4) - \frac{p}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1,p-1}(4) \right) \lfloor \boldsymbol{P}_a \rfloor. & (3.32)
\end{aligned}$$

From the example we see that the only basis function of degree $p-1 = 1$ that is not equal to zero for $\xi = 4$ is $N_{4,1}$. It is seen that the index $4 = n$ and it will henceforth be replaced with n. This gives

$$\begin{aligned}
\lfloor \boldsymbol{C}'(4) \rfloor &= \frac{p}{\xi_{a+p} - \xi_a} N_{a=4,p-1=1}(4) \lfloor \boldsymbol{P}_a \rfloor - \frac{p}{\xi_{a+p+1} - \xi_{a+1}} N_{a+1=4,p-1=1}(4) \lfloor \boldsymbol{P}_a \rfloor & (3.33)\\
&= \frac{p}{\xi_{n+p} - \xi_n} N_{4,1}(4) \lfloor \boldsymbol{P}_n \rfloor - \frac{p}{\xi_{n-1+p+1} - \xi_{n-1+1}} N_{4,1}(4) \lfloor \boldsymbol{P}_{n-1} \rfloor. & (3.34)
\end{aligned}$$

We ascertained in Equation (3.2) that $n + p + 1 = N$, and with the assumptions given above $\xi_N = \xi_{N-f}$ for $f = 1, 2, \ldots, p$. It gives

$$\lfloor \boldsymbol{C}'(\xi_N) \rfloor = \frac{p}{\xi_N - \xi_n} \left(\lfloor \boldsymbol{P}_n \rfloor - \lfloor \boldsymbol{P}_{n-1} \rfloor \right). \tag{3.35}$$

B-splines in multiple dimensions

Construction of B-spline surfaces and solids is shown below.

$$\lfloor \boldsymbol{S}(\xi,\eta) \rfloor = \sum_{a=1}^{n}\sum_{b=1}^{m} N_{a,p}(\xi) M_{b,q}(\eta) \lfloor \boldsymbol{P}_{a,b} \rfloor, \tag{3.36}$$

$$\lfloor \boldsymbol{S}(\xi,\eta,\zeta) \rfloor = \sum_{a=1}^{n}\sum_{b=1}^{m}\sum_{c=1}^{l} N_{a,p}(\xi) M_{b,q}(\eta) L_{c,r}(\zeta) \lfloor \boldsymbol{P}_{a,b,c} \rfloor, \tag{3.37}$$

where $N_{a,p}$, $M_{b,q}$ and $L_{c,r}$ is defined as in Section 3.1.2.

The first derivatives of the two dimensional B-spline surface on the four edges $\xi = 0$, $\xi = \xi_N$, $\eta = 0$ and $\eta = \eta_M$ are

$$\left\lfloor \frac{\partial \boldsymbol{S}(0,\eta)}{\partial \xi} \right\rfloor = \frac{p}{\xi_{p+2}} \sum_{b=1}^{m} M_{b,q}(\eta) \left(\lfloor \boldsymbol{P}_{2,b} \rfloor - \lfloor \boldsymbol{P}_{1,b} \rfloor \right), \tag{3.38}$$

$$\left\lfloor \frac{\partial \boldsymbol{S}(\xi,0)}{\partial \eta} \right\rfloor = \frac{q}{\eta_{q+2}} \sum_{a=1}^{n} N_{a,p}(\xi) \left(\lfloor \boldsymbol{P}_{a,2} \rfloor - \lfloor \boldsymbol{P}_{a,1} \rfloor \right), \tag{3.39}$$

$$\left\lfloor \frac{\partial \boldsymbol{S}(0,\eta)}{\partial \eta} \right\rfloor = \sum_{b=1}^{m} \frac{\partial M_{b,q}(\eta)}{\partial \eta} \lfloor \boldsymbol{P}_{1,b} \rfloor, \tag{3.40}$$

$$\left\lfloor \frac{\partial \boldsymbol{S}(\xi,0)}{\partial \xi} \right\rfloor = \sum_{a=1}^{n} \frac{\partial N_{a,p}(\xi)}{\partial \xi} \lfloor \boldsymbol{P}_{a,1} \rfloor, \tag{3.41}$$

$$\left\lfloor \frac{\partial \boldsymbol{S}(\xi_N,\eta)}{\partial \xi} \right\rfloor = \frac{p}{\xi_N - \xi_n} \sum_{b=1}^{m} M_{b,q}(\eta) \left(\lfloor \boldsymbol{P}_{n,b} \rfloor - \lfloor \boldsymbol{P}_{n-1,b} \rfloor \right), \tag{3.42}$$

$$\left\lfloor \frac{\partial \boldsymbol{S}(\xi,\eta_M)}{\partial \eta} \right\rfloor = \frac{q}{\eta_M - \eta_m} \sum_{a=1}^{n} N_{a,p}(\xi) \left(\lfloor \boldsymbol{P}_{a,m} \rfloor - \lfloor \boldsymbol{P}_{a,m-1} \rfloor \right), \tag{3.43}$$

$$\left\lfloor \frac{\partial \boldsymbol{S}(\xi_N,\eta)}{\partial \eta} \right\rfloor = \sum_{b=1}^{m} \frac{\partial M_{b,q}(\eta)}{\partial \eta} \lfloor \boldsymbol{P}_{n,b} \rfloor, \tag{3.44}$$

$$\left\lfloor \frac{\partial \boldsymbol{S}(\xi,\eta_M)}{\partial \xi} \right\rfloor = \sum_{a=1}^{n} \frac{\partial N_{a,p}(\xi)}{\partial \xi} \lfloor \boldsymbol{P}_{a,m} \rfloor. \tag{3.45}$$

3.2 NURBS

NURBS is the acronym for Non-Uniform Rational B-Spline. The first feature means that the knot vectors for NURBS are not necessarily uniform. This implies that the size of the knot spans may change. The other statement is that a NURBS is rational. The rationality occurs when the B-spline in $\mathbb{R}^{d+1}$ is projectively transformed into $\mathbb{R}^{d}$ to become a NURBS.

The theory in this chapter is taken from [8].

3.2.1 Control Points

The control points used for NURBS have, in addition to the coordinates used for B-splines, an extra component which is the weighting of each control point. This is called h_a.

$$\lfloor \boldsymbol{P}_a \rfloor = \lfloor x_a \quad y_a \quad z_a \quad h_a \rfloor \tag{3.46}$$

The control points used for B-splines are now altered to become *homogeneous coordinates* [8]. These control points are weighted coordinates, in the sense that

$$\lfloor \boldsymbol{P}^h_a \rfloor \in \mathbb{R}^{3+1}, \quad \lfloor \boldsymbol{P}^h_a \rfloor = \lfloor x_a h_a \quad y_a h_a \quad z_a h_a \quad h_a \rfloor. \tag{3.47}$$

The control points for the NURBS are thus defined as

$$\lfloor \boldsymbol{P}_a \rfloor_g = \frac{\lfloor \boldsymbol{P}^h_a \rfloor_g}{h_a}, \quad \lfloor \boldsymbol{P}_a \rfloor \in \mathbb{R}^3, \tag{3.48}$$

where $g = 1, 2, 3$ and denotes the component number of the control point. We can then define the weighting function:

$$H(\xi) = \sum_{a=1}^{n} N_{a,p}(\xi) h_a. \tag{3.49}$$

To get representation in 1D or 2D, either two or one of the coordinates are held constant for all control points.

3.2.2 NURBS Basis Functions

The NURBS basis functions are defined as

$$R_{a,p}(\xi) = \frac{N_{a,p}(\xi) h_a}{H(\xi)} = \frac{N_{a,p}(\xi) h_a}{\sum_{a=1}^{n} N_{a,p}(\xi) h_a}, \tag{3.50}$$

where $N_{a,p}(\xi)$ is as defined in Section 3.1.2. For multiple dimensions they are defined as

$$R^{p,q}_{a,b}(\xi,\eta) = \frac{N_{a,p}(\xi) M_{b,q}(\eta) h_{a,b}}{H(\xi,\eta)} = \frac{N_{a,p}(\xi) M_{b,q}(\eta) h_{a,b}}{\sum_{a=1}^{n} \sum_{b=1}^{m} N_{a,p}(\xi) M_{b,q}(\eta) h_{a,b}}, \tag{3.51}$$

$$R^{p,q,r}_{a,b,c}(\xi,\eta,\zeta) = \frac{N_{a,p}(\xi) M_{b,q}(\eta) L_{c,r}(\zeta) h_{a,b,c}}{H(\xi,\eta,\zeta)} = \frac{N_{a,p}(\xi) M_{b,q}(\eta) L_{c,r}(\zeta) h_{a,b,c}}{\sum_{a=1}^{n} \sum_{b=1}^{m} \sum_{c=1}^{l} N_{a,p}(\xi) M_{b,q}(\eta) L_{c,r}(\zeta) h_{a,b,c}}. \tag{3.52}$$

This fraction is rational because the denominator is a function.

The one-dimensional expression shows easily how the B-splines are a special form of NURBS. We see that when all the weighting factors are equal, $h_a = h$ can be placed outside the summation sign, and given that $\sum_{a=1}^{n} N_{a,p} = 1$, Equation (3.50) reduces to $R_{a,p}(\xi) = N_{a,p}(\xi)$, which is the same as for B-splines.

3.2.3 Construction of NURBS

The construction of NURBS is similar to the construction of B-splines, however, the NURBS basis functions are used instead of the non-rational basis functions.

$$\lfloor \boldsymbol{C}(\xi) \rfloor = \sum_{a=1}^{n} R_{a,p}(\xi) \lfloor \boldsymbol{P}_a \rfloor = \lfloor \boldsymbol{R} \rfloor [\boldsymbol{P}] \tag{3.53}$$

$$\lfloor \boldsymbol{S}(\xi,\eta) \rfloor = \sum_{a=1}^{n} \sum_{b=1}^{m} R_{a,b}^{p,q}(\xi,\eta) \lfloor \boldsymbol{P}_{a,b} \rfloor = \lfloor \boldsymbol{R} \rfloor [\boldsymbol{P}] \tag{3.54}$$

$$\lfloor \boldsymbol{S}(\xi,\eta,\zeta) \rfloor = \sum_{a=1}^{n} \sum_{b=1}^{m} \sum_{c=1}^{l} R_{a,b,c}^{p,q,r}(\xi,\eta,\zeta) \lfloor \boldsymbol{P}_{a,b,c} \rfloor = \lfloor \boldsymbol{R} \rfloor [\boldsymbol{P}] \tag{3.55}$$

An example of a NURBS curve with

$$\boldsymbol{P}^h = \begin{bmatrix} 1 & 1 & 0.8 \\ 2 & 1 & 1 \\ 2 & 2 & 1.2 \\ 1 & 2 & 1 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{P} = \begin{bmatrix} 1.25 & 1.25 \\ 2 & 1 \\ 1.67 & 1.67 \\ 1 & 2 \end{bmatrix} \tag{3.56}$$

and basis functions as in Figure 3.1 is shown in Figure 3.3. This figure also includes the B-spline from which the NURBS is projectively transformed.

3.3 Patches and Elements

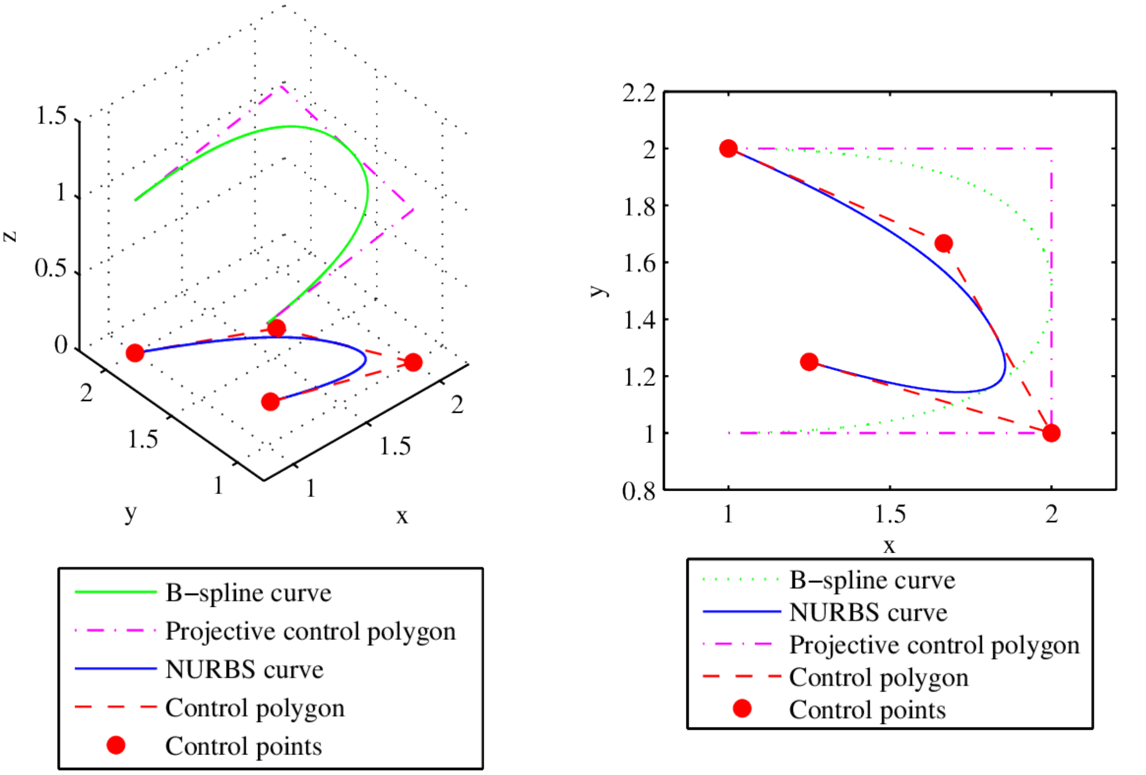
The definition of an element in an IGA varies, but for the purpose of this thesis it has been chosen to define an element as a knot span. Shown in Figure 3.6 is an example of an element in a patch with knot vectors as in the example in Section 3.1.2 in two directions.

As seen from Figure 3.1, the bandwidth of the splines comprises both the elements, but only one patch. The possibility of using Gauss quadrature rule for integration on element level is present [8] because the splines are C^∞ polynomials at the knot intervals. This will be further discussed in Section 5.

Figure 3.6 also shows that given the definition of the knot vectors as orthogonal in the parameter space, the parametric patch is always rectangular. This means that only rectangular elements can be used.

Alternatively to interpolation functions that are used in traditional FEA, splines are local to patches rather than elements which means that the bandwidth for a spline comprise the width of several elements. A structure can be divided into either one, two or several patches depending on the details of the analysis. For instance, a difficult geometry, such as a hole or a corner, could require multiple patches.

For arbitrary topologies, the option of making each patch consisting of only one element is highly relevant. With such a definition, unstructured meshes can be constructed, and the isogeometric concept can hence be just as adaptable as unstructured meshes created in traditional FEA. Isogeometric patches such as these are called Bézier elements, and are made from B-spline patches. An example with $p = 2$ is shown in Figure 3.4. However, such an approach will include severe increase of time spent on modelling.



(a) NURBS and B-spline in 3D

(b) NURBS and B-spline in the xy-plane

Figure 3.3: Example of B-spline with corresponding NURBS

A brief introduction to multiple patches and continuity between them will be given in Section 3.3.2.

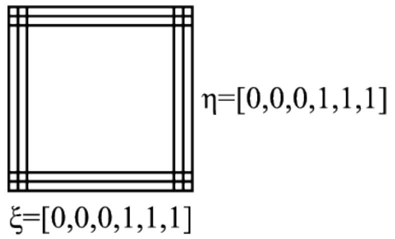


Figure 3.4: Patch consisting of one element, $p = 2$

3.3.1 Refinement

The different methods for refining a patch in IGA include knot insertion and order elevation or a combination of these [2]. As for the refinement in FEA, the refinement of a patch will give better accuracy of the solution. The following methods change neither the geometry nor the parametrization of the splines.

Knot insertion

Knot insertion as a refinement method can include insertion of new knots, or multiplicity of already existing knots. When adding new knots without increasing the degree of the spline, the number of basis functions used must also be increased, as shown in Equation (3.2). Additionally it is shown that the number of basis functions used to construct the spline must be equal to the number of control points, so that additional control points must be created. This is done by the formula

$$\bar{\boldsymbol{P}} = \boldsymbol{T}^p \boldsymbol{P}, \tag{3.57}$$

where

$$T^0_{\alpha i} = \begin{cases} 1 & \bar{\xi}_\alpha \in [\xi_i, \xi_{i+1}), \\ 0 & \text{otherwise}, \end{cases} \tag{3.58}$$

and

$$T^{b+1}_{\alpha i} = \frac{\bar{\xi}_{\alpha+b} - \xi_i}{\xi_{i+b} - \xi_i} T^b_{\alpha i} + \frac{\xi_{i+b+1} - \bar{\xi}_{\alpha+b}}{\xi_{i+b+1} - \xi_{i+1}} T^b_{\alpha i} \tag{3.59}$$

for $b = 0, 1, \ldots, p-1$ in direction of ξ and $b = 0, 1, \ldots, q-1$ in direction of η and so on. In this way the continuity of the spline is maintained, even though the continuity of the basis functions may have been reduced if the insertion has been done by knot multiplicity.

The knot insertion in a patch can be compared to the h-refinement method described in Section 2.1.1, since the method partitions the spline into new elements. However, in h-refinement, the continuity between the new elements is C^0 whereas the continuity between the new elements when applying knot insertion is C^{p-s}, where s is the resulting multiplicity of the knot after insertion. When a new knot is inserted, s clearly is one.

Order elevation

When the order of the spline is increased without adding new knots, the refinement method is called order elevation. The process starts with increasing the multiplicity of all existing knots until they are equal to the polynomial degree, making the spline C^0 continuous across knots. This is equivalent to dividing the spline into multiple splines extending only one knot span. Then the polynomial degree of each of these splines is increased to the desired order, and the multiplication of the knots is decreased to reintroduce the original continuity. Equation (3.2) implies that when increasing the number of knots and the polynomial degree, the number of basis functions used to construct the spline must also be increased.

p-refinement described in Section 3.3.1 and order elevation have noticeable similarities, but as the order elevation starts with a C^0 continuity, the p-refinement can be carried out with any basis of shape functions.

k-refinement

k-refinement is a combination of knot insertion and order elevation, however, it must be done in a specific order. If knot insertion is performed, thereby inserting a new knot, the continuity at this knot is C^{p-1}. By then performing order elevation including both increase of the polynomial degree and multiplication of knots, the continuity at the new knot is still C^{p-1}. If the order of these doings is reversed, first the polynomial order is increased to $\tilde{p}$, then a new knot is inserted, the continuity of the new knot is $C^{\tilde{p}-1}$. Thus both the number of knots and the polynomial degree are increased. Because of this, the number of basis functions used to construct the spline must also be increased, cf. Equation (3.2). It is this latter sequence that is referred to as k-refinement.

3.3.2 Continuity

Continuity of B-spline patches with equal edge refinement

As seen in Section 3.1.2 the basis functions are piecewise polynomials joined at the knots. This makes the B-spline C^∞ continuous between the knots and C^{p-s} continuous at a knot with multiplication s. Between the patches, however, the B-splines are not continuous unless certain requirements are fulfilled. In this section the continuity conditions between two B-spline surfaces will be derived [8].

To demonstrate the continuity conditions for B-spline surfaces, two patches with a common edge are considered. It is from now on assumed that this edge is $\xi = \xi_N$ for the first patch and $\xi = 0$ for the second patch. This implies that the B-splines in direction of η at the edge of both patches are equal.

To fulfil the C^0 requirement stated in Section 3.1.2, the control points on the edge have to coincide.

$$\boldsymbol{P}^1_{n,b} = \boldsymbol{P}^2_{1,b} \tag{3.60}$$

Further continuity is not required, but may enhance numerical accuracy [9]. Properties resulting from higher continuity will not be discussed very closely, but Hughes, Cottrell and Bazilevs [10] have shown that higher continuity require more computational time[1] than lower order analyses. The key factor is the ability the direct solvers have to optimize solution of analyses where the basis functions have support on several elements. The bandwidth of a system represents the number of knots on which a given basis function overlaps with other basis functions, and determines the non-zero entries in the global matrices. The maximum bandwidth of an IGA is $p-1$ and the bandwidth does not increase as the continuity of the basis is increased as long as the degree is kept constant. However, for direct solvers, the minimum bandwidth of the matrices increases and approaches $p-1$

[1] and also more memory, but this will not be further discussed

as the continuity increases, leading to fewer opportunities of condensation of degrees of freedom.

C^1 continuity is obtained by the following steps:

$$\boldsymbol{S}^1(\xi_N,\eta) = \boldsymbol{S}^2(0,\eta), \tag{3.61}$$

$$M^1_{b,q}(\eta) = M^2_{b,q}(\eta), \quad b = 1,2,\dots,m. \tag{3.62}$$

C^1 continuity implies that the tangent vector for both patches must be equal. From Equations (3.40) and (3.44) and the coinciding control points, we get

$$\left.\begin{aligned} \frac{\partial \boldsymbol{S}^1(\xi_N,\eta)}{\partial\eta} &= \sum_{b=1}^{m} \frac{\partial M^1_{b,q}(\eta)}{\partial\eta}\boldsymbol{P}^1_{n,b} \\ \frac{\partial \boldsymbol{S}^2(0,\eta)}{\partial\eta} &= \sum_{b=1}^{m} \frac{\partial M^2_{b,q}(\eta)}{\partial\eta}\boldsymbol{P}^2_{1,b} \end{aligned}\right\} \frac{\partial \boldsymbol{S}^1(\xi_N,\eta)}{\partial\eta} = \frac{\partial \boldsymbol{S}^2(0,\eta)}{\partial\eta}, \tag{3.63}$$

and the continuity with respect to η is thus fulfilled at all times.

Equations (3.38) and (3.42) give

$$\frac{\partial \boldsymbol{S}^1(\xi_N,\eta)}{\partial\xi} = \frac{p}{\xi_N - \xi_n}\sum_{b=1}^{m} M^1_{b,q}(\eta)\left(\boldsymbol{P}^1_{n,b} - \boldsymbol{P}^1_{n-1,b}\right), \tag{3.64}$$

$$\frac{\partial \boldsymbol{S}^2(0,\eta)}{\partial\xi} = \frac{p}{\xi_{p+2}}\sum_{b=1}^{m} M^2_{b,q}(\eta)\left(\boldsymbol{P}^2_{2,b} - \boldsymbol{P}^2_{1,b}\right). \tag{3.65}$$

The degree of the B-splines and the value of each knot are constant, so the condition

$$\frac{\partial \boldsymbol{S}^1(\xi_N,\eta)}{\partial\xi} = \frac{\partial \boldsymbol{S}^2(0,\eta)}{\partial\xi}, \tag{3.66}$$

is fulfilled if

$$\frac{p}{\xi_{p+2}}\left(\boldsymbol{P}^2_{2,b} - \boldsymbol{P}^1_{n,b}\right) = \frac{p}{\xi_N - \xi_n}\left(\boldsymbol{P}^1_{n,b} - \boldsymbol{P}^1_{n-1,b}\right). \tag{3.67}$$

Here, the property of $\boldsymbol{P}^2_{1,b} = \boldsymbol{P}^1_{n,b}$ is used.

Continuity of NURBS Patches

As shown in Section 3.2.2 are the NURBS generalizations of B-splines, thus all of the continuity requirements stated above also apply for NURBS.

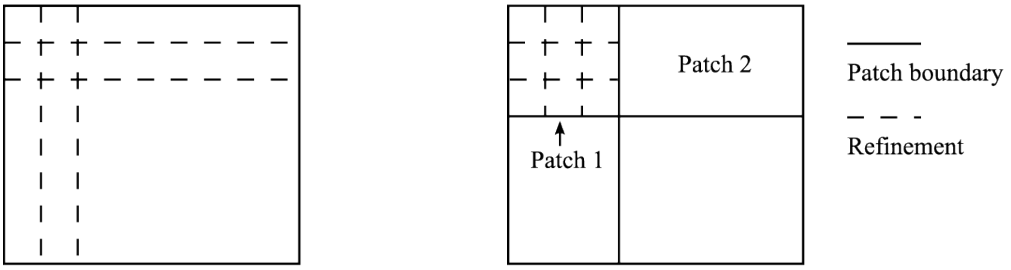
For a NURBS surface the continuity condition is applied to the homogeneous coordinates so that

$$\frac{p}{\xi_{p+2}}\left(\boldsymbol{P}^{h,2}_{2,b} - \boldsymbol{P}^{h,1}_{n,b}\right) = \frac{p}{\xi_N - \xi_n}\left(\boldsymbol{P}^{h,1}_{n,b} - \boldsymbol{P}^{h,1}_{n-1,b}\right). \tag{3.68}$$

Continuity of B-spline Patches with Different Edge Refinement

The isogeomtric concept also allows for continuity between two patches with different refinement at the adjoining edges. However, the control points for each edge must still coincide. The method described below also assumes that the refinement is done by knot insertions from two initially equal patches [2].

With these measures, the refinement shown in Figure 3.5b is possible, in contrast to the refinement when using only one patch, shown in Figure 3.5a.



(a) Refinement with one patch

(b) Refinement with multiple patches

Figure 3.5: Refinement with one and multiple patches

Before describing this technique, the related vectors are arranged in order after whether or not they correspond to the adjoining edge. Subscript e represents features lying on the edge and o features not lying on the edge, superscript 1 and 2 represent patches 1 and 2 respectively.

$$\{\boldsymbol{P}^1\} = \begin{Bmatrix} \boldsymbol{P}_o^1 \\ \boldsymbol{P}_e^1 \end{Bmatrix}, \quad \{\boldsymbol{P}^2\} = \begin{Bmatrix} \boldsymbol{P}_o^2 \\ \boldsymbol{P}_e^2 \end{Bmatrix}, \quad [\tilde{\boldsymbol{T}}] = \begin{bmatrix} \tilde{\boldsymbol{T}}_o & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{T}}_e \end{bmatrix}, \tag{3.69}$$

$$\{\boldsymbol{d}^1\} = \begin{Bmatrix} \boldsymbol{d}_o^1 \\ \boldsymbol{d}_e^1 \end{Bmatrix}, \quad \{\boldsymbol{d}^2\} = \begin{Bmatrix} \boldsymbol{d}_o^2 \\ \boldsymbol{d}_e^2 \end{Bmatrix}, \quad [\boldsymbol{K}^1] = \begin{bmatrix} \boldsymbol{K}_{oo}^1 & \boldsymbol{K}_{oe}^1 \\ \boldsymbol{K}_{eo}^1 & \boldsymbol{K}_{ee}^1 \end{bmatrix}, \quad [\boldsymbol{K}^2] = \begin{bmatrix} \boldsymbol{K}_{oo}^2 & \boldsymbol{K}_{oe}^2 \\ \boldsymbol{K}_{eo}^2 & \boldsymbol{K}_{ee}^2 \end{bmatrix}. \tag{3.70}$$

were $[\tilde{\boldsymbol{T}}]$ is *the multi-dimensional generalization of the extension operator* [2] defined in Equation (3.57).

C^0-continuity is maintained by introducing

$$\{\boldsymbol{P}_e^1\} = \{\boldsymbol{P_e}^2\}, \tag{3.71}$$

and hence

$$\{\bar{\boldsymbol{P}}_e^2\} = [\tilde{\boldsymbol{T}}_e]\{\boldsymbol{P}_e^1\}, \tag{3.72}$$

from Equation 3.57. This constraint is also applied to the degrees of freedom so that instead of calculating the field variables from the solution of patch 2, the variables are

$$\{\bar{\boldsymbol{d}}_e^2\} = [\tilde{\boldsymbol{T}}_e]\{\boldsymbol{d}_e^1\}. \tag{3.73}$$

Inserting this into the equilibrium equation for each patch gives

$$[\boldsymbol{K}]\{\boldsymbol{D}\} = \{\boldsymbol{R}\}, \tag{3.74}$$

for the total system, with

$$[\boldsymbol{K}] = \begin{bmatrix} \boldsymbol{k}_{oo}^1 & \boldsymbol{k}_{oe}^1 & \boldsymbol{0} \\ \boldsymbol{k}_{eo}^1 & \boldsymbol{k}_{ee}^1 + \tilde{\boldsymbol{T}}_e^T \boldsymbol{k}_{ee}^2 \tilde{\boldsymbol{T}}_e & \tilde{\boldsymbol{T}}_e^T \boldsymbol{k}_{eo}^2 \\ \boldsymbol{0} & \boldsymbol{k}_{oe}^2 \tilde{\boldsymbol{T}}_e & \boldsymbol{k}_{oo}^2 \end{bmatrix}, \tag{3.75}$$

$$\{\boldsymbol{D}\} = \begin{Bmatrix} \boldsymbol{d}_o^1 \\ \boldsymbol{d}_e^1 \\ \boldsymbol{d}_o^2 \end{Bmatrix}, \quad \{\boldsymbol{R}\} = \begin{Bmatrix} \boldsymbol{r}_o^1 \\ \boldsymbol{r}_e^1 + \tilde{\boldsymbol{T}}_e^T \boldsymbol{r}_e^2 \\ \boldsymbol{r}_o^2 \end{Bmatrix}. \tag{3.76}$$

The terms linking the degrees of freedom not lying on the shared edge at patch 2 with the degrees of freedom and the reaction forces lying on the shared edge at patch 1 are premultiplied by $\tilde{\boldsymbol{T}}_e^T$ to obtain the equal number of equations resulting in the somewhat more complex middle term of $[\boldsymbol{K}]$.

3.4 Discretization

Solving a problem with IGA is in many respects similar to the process of solving a problem with FEA. The structure must be discretized and the value of the degrees of freedom must be calculated. This can be done as in the FEA, fulfilling the equilibrium equations.

Only solid representation of a structure analyzed with IGA will be described in this thesis, and the discrete displacement vector is thus defined as

$$\{\boldsymbol{d}\} = \lfloor u_{a,b,c} \quad v_{a,b,c} \quad w_{a,b,c} \rfloor^T, \tag{3.77}$$

where $a = 1, 2, \cdots, n_{cp}$, $b = 1, 2, \cdots, m_{cp}$ and $c = 1, 2, \cdots, l_{cp}$ is the numbering of control points. The degrees of freedom are located in the control points and the formula

$$u_j = \sum_a \sum_b \sum_c R_{a,b,c}^{p,q,r} d_j^{a,b,c} \tag{3.78}$$

gives the distribution of the displacements of the structure in terms of ξ, η and ζ. These are the same NURBS basis functions used to describe the geometry in Section 3.2.

The Jacobian matrix for IGA is important to discuss. Since the matrix must be created for each element, it is important to recognize which element the bandwidth of the NURBS basis functions comprise when transforming the expression from physical to parametric space, and then only include the NURBS basis functions relevant to the computation of each element. These basis functions and their corresponding control points, can be gathered in altered matrices, $\tilde{\boldsymbol{R}}$ and $\tilde{\boldsymbol{P}}$. The Jacobian matrix for transformation from physical to parametric space is then applied to these basis functions only.

A Jacobian matrix for transformation from parameter space to the parent element used in Gauss integration also has to be constructed . The parent element compared to the parameter space is illustrated in Figure 3.6. The figure shows that the elements in parameter space always are rectangular which implies that the Jacobian used in transformation from parameter space to the parent element always is constant.

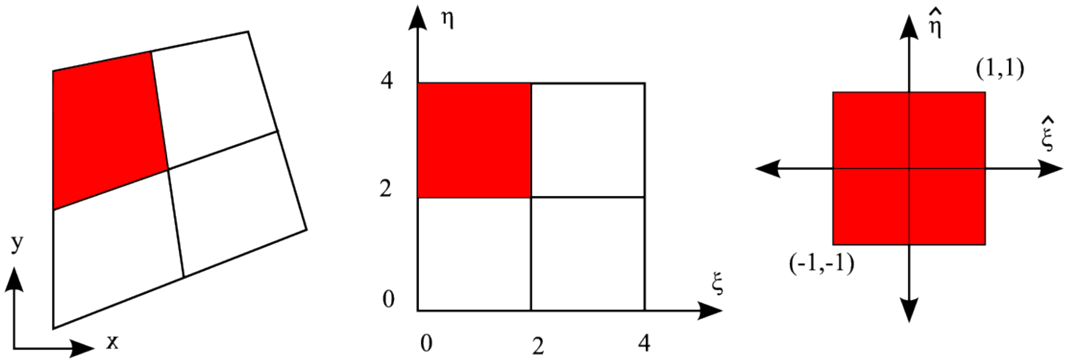


Figure 3.6: Transformation from physical space through parameter space to parent element

Chapter 4

Governing Equations for Finite Deformation Solids and Shells

The basic formulations of continuum mechanics have been used as foundation for this chapter. Details can be found in [7].

4.1 Reference System

The description of motion for a particle in a solid structure can be described by the Lagrangian description. The Lagrangian description refers the physical quantities to a chosen reference configuration. All particles in a solid structure have a position denoted by a vector of coordinates. The position of a particle in the initial configuration is denoted $\{\boldsymbol{X}\} = \lfloor X \quad Y \quad Z \rfloor^T$. The coordinates are rectangular Cartesian coordinates. As the solid structure deforms, the deformed configuration is obtained. For the particle described above, the position vector in the deformed configuration is a function of the initial position and the elapsed time:

$$\{\boldsymbol{x}\} = \{\boldsymbol{x}(\boldsymbol{X}, t)\} = \{\boldsymbol{X}\} + \{\boldsymbol{u}(\boldsymbol{X}, t)\}, \tag{4.1}$$

where $\{\boldsymbol{u}(\boldsymbol{X}, t)\} = \lfloor u(\boldsymbol{X}, t) \quad v(\boldsymbol{X}, t) \quad w(\boldsymbol{X}, t) \rfloor^T$ is the translational displacements in direction of the respective coordinates. For simplicity, $(\boldsymbol{X}, t)$ is usually not written.

4.2 Kinematics

The base of the global Cartesian coordinate system [11] consists of

$$\{\boldsymbol{i}_1\} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \quad \text{and} \quad \{\boldsymbol{i}_2\} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}, \quad \{\boldsymbol{i}_3\} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}. \tag{4.2}$$

The corresponding Cartesian coordinates, $\{\boldsymbol{x}\}$, are expressed as

$$\{\boldsymbol{x}\} = x^i \{\boldsymbol{i}_i\}. \tag{4.3}$$

In the succeeding derivations, upper case letters refer to the undeformed configuration and lower case letters to the deformed configuration. The derivations apply to both configurations, but are only shown for the deformed configuration.

For a solid structure, the orthogonal Cartesian coordinate system is used both for the global and the local bases. For a shell, referred to by the mid-surface of the structure, a local basis using curvilinear coordinates is adequate to use [8]. The representation uses the covariant and the contravariant bases, $\{\boldsymbol{g}_i\}$ and $\{\boldsymbol{g}^i\}$, and the contravariant and the covariant coordinates, θ^i and θ_i to express the position vector:

$$\{\boldsymbol{x}\} = \theta^i\{\boldsymbol{g}_i\} = \theta_i\{\boldsymbol{g}^i\}. \tag{4.4}$$

The covariant base vectors are defined as

$$\{\boldsymbol{g}_i\} = \left\{\frac{\partial \boldsymbol{x}}{\partial \theta^i}\right\}. \tag{4.5}$$

The relation between the covariant and contravariant base vectors is defined by the Kronecker delta:

$$\{\boldsymbol{g}_i\} \cdot \{\boldsymbol{g}^j\} = \delta_i^j = \begin{cases} 0 & i \neq j, \\ 1 & i = j, \end{cases} \tag{4.6}$$

which implies $\{\boldsymbol{g}_i\} \perp \{\boldsymbol{g}^i\}$. For a surface, the third covariant base vector is always perpendicular to $\{\boldsymbol{g}_1\}$ and $\{\boldsymbol{g}_2\}$ because it is the normal to the surface defined by the curvilinear coordinates θ^1 and θ^2. $\{\boldsymbol{g}_3\}$ is expressed by the formula for a vector perpendicular to two others:

$$\{\boldsymbol{g}_3\} = \left\{\frac{\boldsymbol{g}_1 \times \boldsymbol{g}_2}{|\boldsymbol{g}_1 \times \boldsymbol{g}_1|}\right\}. \tag{4.7}$$

Due to this definition

$$\{\boldsymbol{g}_3\} = \{\boldsymbol{g}^3\}. \tag{4.8}$$

The change in the base vectors due to the deformation of the structure can be expressed by the deformation gradient, $[\boldsymbol{F}]$. The deformation gradient expresses the change in distance between two particles from the initial to the deformed configuration:

$$[F_{ij}] = [\boldsymbol{F}] = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} & \frac{\partial x}{\partial Z} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} & \frac{\partial y}{\partial Z} \\ \frac{\partial z}{\partial X} & \frac{\partial z}{\partial Y} & \frac{\partial z}{\partial Z} \end{bmatrix} = \begin{bmatrix} 1 + \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & 1 + \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \\ \frac{\partial w}{\partial X} & \frac{\partial w}{\partial Y} & 1 + \frac{\partial w}{\partial Z} \end{bmatrix} \tag{4.9}$$

and the relation between the base vectors in the initial and current configuration is

$$\{\boldsymbol{g}_i\} = [\boldsymbol{F}] \cdot \{\boldsymbol{G}_i\}, \tag{4.10}$$

$$\{\boldsymbol{g}^i\} = [\boldsymbol{F}]^{-T} \cdot \{\boldsymbol{G}^i\}. \tag{4.11}$$

The definitions can be used to derive expressions for the infinitesimal volume and surface elements for a shell. This will not be done in this thesis, details may be found in [12].

4.3 Strain Measures

Describing the strains in a solid structure can be done in several ways. No matter what measure that is chosen, the strain measure must fulfil two conditions [13]:

- predict zero strains for arbitrarily rigid-body motions
- reduce to the infinitesimal strains if the nonlinear terms are neglected

In addition, it should go towards $+\infty$ for "infinite stretching" and $-\infty$ for "full compression".

The simplest strain measure, and most common for geometrically linear problems, is the *engineering strain*:

$$[\epsilon_{E,ij}] = [\boldsymbol{\epsilon}_E] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = [\boldsymbol{\partial}]\{\boldsymbol{u}\} = [\partial_{ij} u_j]. \tag{4.12}$$

The engineering strain uses the initial configuration as reference configuration. Also the *logarithmic strain* uses the initial configuration as reference configuration. The logarithmic strain is the only strain measure that fulfils both the requirements and recommendations given above. The relation between the engineering and logarithmic strain is

$$[\epsilon_{L,ij}] = [\boldsymbol{\epsilon}_L] = [\log(1 + \epsilon_{E,ij})]. \tag{4.13}$$

The strain measure commonly used for finite strains is the *Green strain.* The Green strain takes into account the finite deformations and gives a nonlinear relation between the deformations and the strains, but it does not go towards $-\infty$ for "full compression". However, it has been shown to have small error if the strains are not too large [13]. The Green strain may be expressed as

$$[\epsilon_{G,ij}] = [\boldsymbol{\epsilon}_G] = \begin{bmatrix} \epsilon_{GXX} & \epsilon_{GXY} & \epsilon_{GXZ} \\ \epsilon_{GYX} & \epsilon_{GYY} & \epsilon_{GYZ} \\ \epsilon_{GZX} & \epsilon_{GZY} & \epsilon_{GZZ} \end{bmatrix} = \frac{1}{2}\left([\boldsymbol{F}]^T[\boldsymbol{F}] - \lceil \boldsymbol{I} \rfloor\right), \tag{4.14}$$

with $[\boldsymbol{F}]$ as in Equation (4.9). The uppercase indices indicate that the initial configuration is the reference configuration. The Green strain matrix is symmetric hence the terms can be expressed as:

$$\begin{Bmatrix} \epsilon_{GXX} \\ \epsilon_{GYY} \\ \epsilon_{GZZ} \\ \epsilon_{GXY} \\ \epsilon_{GYZ} \\ \epsilon_{GZX} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial X} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial X}\right)^2 + \left(\frac{\partial v}{\partial X}\right)^2 + \left(\frac{\partial w}{\partial X}\right)^2\right] \\ \frac{\partial v}{\partial Y} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial Y}\right)^2 + \left(\frac{\partial v}{\partial Y}\right)^2 + \left(\frac{\partial w}{\partial Y}\right)^2\right] \\ \frac{\partial w}{\partial Z} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial Z}\right)^2 + \left(\frac{\partial v}{\partial Z}\right)^2 + \left(\frac{\partial w}{\partial Z}\right)^2\right] \\ \frac{1}{2}\left(\frac{\partial u}{\partial Y} + \frac{\partial v}{\partial X}\right) + \frac{1}{2}\left[\left(\frac{\partial u}{\partial X}\right)\left(\frac{\partial u}{\partial Y}\right) + \left(\frac{\partial v}{\partial X}\right)\left(\frac{\partial v}{\partial Y}\right) + \left(\frac{\partial w}{\partial X}\right)\left(\frac{\partial w}{\partial Y}\right)\right] \\ \frac{1}{2}\left(\frac{\partial v}{\partial Z} + \frac{\partial w}{\partial Y}\right) + \frac{1}{2}\left[\left(\frac{\partial u}{\partial Y}\right)\left(\frac{\partial u}{\partial Z}\right) + \left(\frac{\partial v}{\partial Y}\right)\left(\frac{\partial v}{\partial Z}\right) + \left(\frac{\partial w}{\partial Y}\right)\left(\frac{\partial w}{\partial Z}\right)\right] \\ \frac{1}{2}\left(\frac{\partial w}{\partial X} + \frac{\partial u}{\partial Z}\right) + \frac{1}{2}\left[\left(\frac{\partial u}{\partial Z}\right)\left(\frac{\partial u}{\partial X}\right) + \left(\frac{\partial v}{\partial Z}\right)\left(\frac{\partial v}{\partial X}\right) + \left(\frac{\partial w}{\partial Z}\right)\left(\frac{\partial w}{\partial X}\right)\right] \end{Bmatrix}. \tag{4.15}$$

Another way of defining the Green strain is by the change of length of a material fibre from the undeformed to the deformed configuration:

$$ds^2 - ds_0^2 = 2\{d\boldsymbol{X}\}^T[\boldsymbol{\epsilon}_G]\{d\boldsymbol{X}\}. \tag{4.16}$$

A strain measure that uses the current configuration as reference configuration is the *Almansi strain measure.* This measure is on a similar form as the Green strain, but uses as stated the current configuration as the reference configuration [14]:

$$[\epsilon_{A,ij}] = [\boldsymbol{\epsilon}_A] = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{F}\boldsymbol{F}^T). \tag{4.17}$$

4.4 Stress Measures

There are multiple options when considering what stress measure to choose for a solid structural problem [13]. An important requirement is that the stress and strain measure must have the same reference state. The surface traction vector is defined as

$$\{\boldsymbol{\phi}\} = \left\{\frac{d\boldsymbol{p}}{da}\right\}, \tag{4.18}$$

where $\{\boldsymbol{p}\}$ is the forces acting on the surface and da is an infinitesimal surface element in the current configuration.

The *Cauchy stress tensor*, $[\boldsymbol{\sigma}]$, also known as the true stress, uses the current configuration as reference configuration. The definition of the Cauchy stress is obtained by solving

$$\{\boldsymbol{\phi}\} = [\boldsymbol{\sigma}]\{\boldsymbol{n}\}, \tag{4.19}$$

where $\{\boldsymbol{n}\}$ is the outward normal to the infinitesimal surface element da. This stress measure is often used in combination with the logarithmic strain and it is energy conjugate to the Almansi strain.

The 2^{nd} *Piola Kirchhoff (PK2)* stress tensor is work conjugate to the Green strain and relates the current force increment to the initial infinitesimal surface element, dA. The transformation from the Cauchy stress to the PK2 stress tensor is defined through Nanson's formula:

$$[\boldsymbol{S}] = J[\boldsymbol{F}]^{-1}[\boldsymbol{\sigma}][\boldsymbol{F}]^{-T}, \tag{4.20}$$

where J is the determinant of the deformation gradient given in Equation (4.9) and maps a volume element from the initial to the current configuration:

$$dv = J\,dV. \tag{4.21}$$

A stress tensor given by the product of the Cauchy stress and J, is the *Kirchhoff stress tensor*:

$$[\boldsymbol{\tau}] = J[\boldsymbol{\sigma}]. \tag{4.22}$$

The stresses, now referred to as σ, can be divided into two parts; a deviatoric and a hydrostatic part. The deviatoric part represents the stresses causing the structure to alter its shape, whereas the hydrostatic stress component only changes the structures volume. The definitions of the two components are given in Equations (4.23) and (4.24).

$$\sigma_h = \frac{1}{3}\sigma_{kk}, \tag{4.23}$$

$$\sigma'_{ij} = \sigma_{ij} - \sigma_h \delta_{ij}, \quad \delta ij = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases} \tag{4.24}$$

The principal values of the stresses are calculated from the formula [15]

$$\sigma^3 - I_1\sigma^2 + I_2\sigma - I_3 = 0, \tag{4.25}$$

where the three solutions, σ_1, σ_2 and σ_3, of the third degree equation are the maximum, medium and minimum principal stresses,

$$\sigma_1 > \sigma_2 > \sigma_3, \tag{4.26}$$

and I_1, I_2 and I_3 are the stress invariants:

$$I_1 = \sigma_{kk}, \tag{4.27}$$

$$I_2 = \frac{1}{2}\left(\sigma_{ii}\sigma_{jj} - \sigma_{ij}\sigma_{ji}\right), \tag{4.28}$$

$$I_3 = \frac{1}{6}\left(\sigma_{ii}\sigma_{jj}\sigma_{kk} + 2\sigma_{ij}\sigma_{ik}\sigma_{ki} - 3\sigma_{ij}\sigma_{ij}\sigma_{kk}\right). \tag{4.29}$$

4.5 Constitutive Relations

It has been assumed isotropy and homogeneity for the constitutive relations.

4.5.1 Elastic Material

For an elastic material the strain increment is completely elastic:

$$\Delta\epsilon_{ij} = \Delta\epsilon^e_{ij} + \Delta\epsilon^p_{ij} = \Delta\epsilon^e_{ij}, \tag{4.30}$$

which means that all deformations are reversible. This implies that the incremental stress-strain relation can be written as

$$\Delta\sigma_{ij} = c_{ijkl}\Delta\epsilon_{ij}, \tag{4.31}$$

and the total stress-strain relation as

$$\sigma_{ij} = c_{ijkl}\epsilon_{ij}. \tag{4.32}$$

Here, $[\boldsymbol{\sigma}]$, $[\boldsymbol{c}]$ and $[\boldsymbol{\epsilon}]$ are defined as:

$$\{\boldsymbol{\sigma}\} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{Bmatrix}, \quad [\boldsymbol{c}] = \begin{bmatrix} c_{xxxx} & c_{xxyy} & c_{xxzz} & c_{xxxy} & c_{xxyz} & c_{xxzx} \\ c_{yyxx} & c_{yyyy} & c_{yyzz} & c_{yyxy} & c_{yyyz} & c_{yyzx} \\ c_{zzxx} & c_{zzyy} & c_{zzzz} & c_{zzxy} & c_{zzyz} & c_{zzzx} \\ c_{xyxx} & c_{xyyy} & c_{xyzz} & c_{xyxy} & c_{xyyz} & c_{xyzx} \\ c_{yzxx} & c_{yzyy} & c_{yzzz} & c_{yzxy} & c_{yzyz} & c_{yzzx} \\ c_{zxxx} & c_{zxyy} & c_{zxzz} & c_{zxxy} & c_{zxyz} & c_{zxzx} \end{bmatrix}, \quad \{\boldsymbol{\epsilon}\} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{zx} \end{Bmatrix}. \tag{4.33}$$

For the linear elastic material, $[\boldsymbol{c}]$ is a function of the Young's modulus, E, and the Poisson ratio, ν, of the material:

$$[\boldsymbol{c}] = [\boldsymbol{E}] = \begin{bmatrix} (1-\nu)c_{YP} & \nu c_{YP} & \nu c_{YP} & 0 & 0 & 0 \\ \nu c_{YP} & (1-\nu)c_{YP} & \nu c_{YP} & 0 & 0 & 0 \\ \nu c_{YP} & \nu c_{YP} & (1-\nu)c_{YP} & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{YP} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{YP} & 0 \\ 0 & 0 & 0 & 0 & 0 & G_{YP} \end{bmatrix}, \tag{4.34}$$

where c_{YP} and G_{YP} are:

$$c_{YP} = \frac{E}{(1+\nu)(1-2\nu)}, \quad G_{YP} = \frac{E}{2(1+\nu)}. \tag{4.35}$$

4.5.2 Plastic Material

For a plastic material the stress increment has one elastic part and one plastic part. The plastic part occurs when the total stress state increases beyond the yield limit of the material. The stress increment can then be expressed in terms of the total and plastic strain increment as

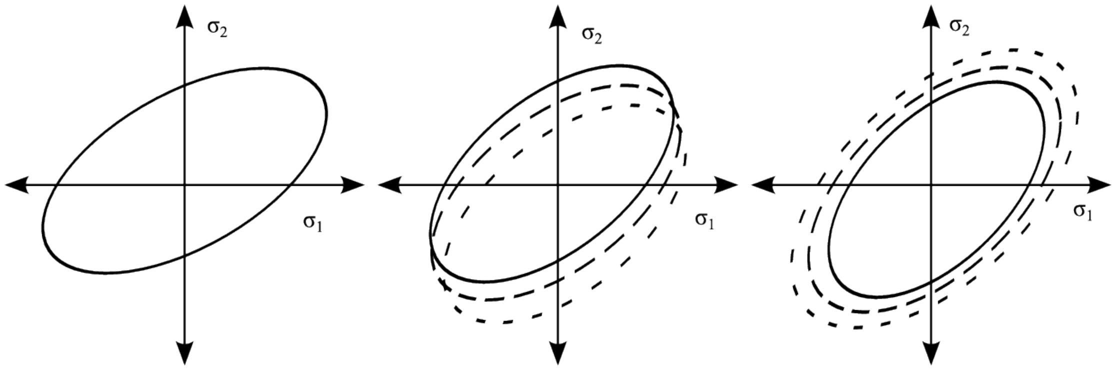
$$\Delta\sigma_{ij} = c_{ijkl}(\Delta\epsilon_{ij} - \Delta\epsilon_{ij}^{p}). \tag{4.36}$$

For an elastic-plastic material there are three essentials to be considered when performing a strength analysis, namely the yield criterion, the flow rule and the hardening rule [16]. An elastic-plastic material can either be elastic-perfectly-plastic, which means that the material does not harden when plastic loading occurs, or it can be associated with either isotropic or kinematic hardening, or a combination of these. The hardening rule determines the development of the yield surface when plastic flow occurs, and the three cases are shown in Figure 4.1.

The yield function expressing the shape of the yield surface is called Y and is written as

$$Y = Y\left(\{\boldsymbol{\sigma}\}, \{\boldsymbol{\alpha}\}, W_p\right) = 0, \tag{4.37}$$

where $\{\boldsymbol{\alpha}\}$ and W_p denotes the kinematic and isotropic hardening parameters respectively. $\{\boldsymbol{\alpha}\}$ is a vector describing the translation of the yield surface as shown in Figure 4.1b and



(a) No hardening **(b)** Kinematic hardening **(c)** Isotropic hardening

Figure 4.1: Hardening rules for elastic-plastic materials in two dimensions

W_p is the work done by the plastic flow to extend the yield surface as shown in Figure 4.1c.

In this thesis, the material model is assumed to follow the Johnson-Cook strength constitutive law [17] which defines the yield stress as

$$\sigma_y = \sigma_0 \left[1 + \frac{A_{JC}}{\sigma_0} p^{n_{JC}}\right] \left[1 + B_{JC} \log\left(\frac{\dot{p}}{\dot{p}_0}\right)\right] \left[1 - T_H^m\right]. \tag{4.38}$$

The middle and last brackets take into account the equivalent plastic strain rate and temperature dependence of the material respectively. The equivalent plastic strain rate is assumed to be constant and the temperature considered in this thesis is room-temperature which implies that both these brackets can be neglected. The yield stress can then be defined as

$$\sigma_y = \sigma_0 + A_{JC} p^{n_{JC}}, \tag{4.39}$$

where σ_0 is the zero-plastic-strain yield strength, A_{JC} is a strain hardening constant, p is the equivalent plastic strain and n_{JC} is a strain hardening exponent. The Johnson-Cook model represents isotropic hardening which means that the yield function can be expressed as

$$Y = \sigma_{eq} - \sigma_y(R), \tag{4.40}$$

where R is defined by the last term in Equation (4.39), and the Johnson-Cook model with constant strain rate and room temperature is thus shown to be a power law. The equivalent stress is the magnitude of the stress state and can be defined by the yield criterion of choice. The von Mises yield criterion has been chosen for this thesis and can be expressed as

$$\sigma_{eq} = \sqrt{\frac{3}{2}\sigma'_{ij}\sigma'_{ij}}, \tag{4.41}$$

where σ'_{ij} is the stress deviator as given in Equation (4.24). p is defined as the integral of the equivalent plastic strain rate [16],

$$p = \int_0^t \dot{p} dt, \tag{4.42}$$

and the equivalent plastic strain rate may be defined as

$$\dot{p} = \sqrt{\frac{2}{3}\dot{\epsilon}^p_{ij}\dot{\epsilon}^p_{ij}}. \tag{4.43}$$

By the definitions of the von Mises stress and the equivalent plastic strain rate it is shown that these variables posses definite positivity.

For an elastic-plastic material, the state of stresses may either be on the yield surface or inside its boundary. It is physically impossible to create a stress state that lies outside the yield surface because the stresses will redistribute. This implies that when an elastic-plastic material is plastically loaded, the change in the yield function must remain zero,

$$dY = 0. \tag{4.44}$$

For a material employing the von Mises yield criterion, the expression for a plastic strain increment is

$$d\epsilon^p_{ij} = \frac{\partial Y}{\partial \sigma_{ij}} d\lambda, \tag{4.45}$$

where λ is the plastic multiplier. This property is derived from an *associated flow rule*, which means that the yield function is the flow rule. These properties are utilized to be able to find an expression for c_{ijkl}, and thus to determine the magnitude of the stress increment. This will be further discussed in Section 5.3.

4.6 Equilibrium Equations

The equilibrium equation can be obtained in several manners and have varying form. Here both the strong and weak form of the equilibrium equation will be discussed.

4.6.1 Strong Form

The strong form of the equilibrium equation is derived from the equilibrium of traction and body forces. The sum of the moments they produce must be equal to zero for a solid structure at rest:

$$\int_{\tilde{A}} \{\mathbf{\Phi}\} d\tilde{A} + \int_{\tilde{V}} \{\mathbf{\Omega}\} d\tilde{V} = \{\mathbf{0}\}, \tag{4.46}$$

where $\{\mathbf{\Phi}\}$ and $\{\mathbf{\Omega}\}$ is the traction and body forces respectively. $d\tilde{A}$ and $d\tilde{V}$ represent the infinitesimal surface and volume element of the structure at the reference configuration of choice [1]. The divergence theorem [18] can be used to show that

$$\int_{\tilde{V}} div\{\boldsymbol{\sigma}\} + \{\mathbf{\Omega}\} d\tilde{V} = \{\mathbf{0}\}, \tag{4.47}$$

which has to hold for any arbitrary volume element $d\tilde{V}$. Thus:

$$div\{\boldsymbol{\sigma}\} + \{\mathbf{\Omega}\} = \mathbf{0}. \tag{4.48}$$

This form of the equilibrium equation can in general not be solved exactly, hence discretized methods as described in this theses may be adequate to use.

[1] The reference configuration for the forces and geometry elements must coincide.

4.6.2 Weak Form

The weak form of the equilibrium equation means that the equilibrium is not satisfied point-wise in the structure, but in an integral form only. This is often the case with discretized structures. In this thesis the *principle of virtual displacements* [2] will be applied to derive an equilibrium equation. The principle of virtual work indicates that the sum of the internal and external virtual work has to cancel each other out:

$$\delta W_{int} + \delta W_{ext} = 0. \tag{4.49}$$

The internal virtual work is the work done by the stresses in the structure when an infinite virtual displacement is applied. The external virtual work is the work done by the applied forces and the body forces due to the applied infinite virtual displacement. The equations applies to every reference configuration chosen (this will be further discussed in the following sections). Here, both the use of the initial and current configuration as the reference configuration is shown. This is done by implementing the Cauchy stress and the true strains and the 2^{nd} Piola Kirchhoff stress and the Green strain respectively:

$$\int_v \{\delta\boldsymbol{\epsilon}\}^T[\boldsymbol{\sigma}]\, dv = \int_v \{\delta\boldsymbol{u}\}^T\{\boldsymbol{\omega}\}\, dv + \int_a \{\delta\boldsymbol{u}\}^T\{\boldsymbol{\phi}\}\, da, \tag{4.50}$$

$$\int_V \{\delta\boldsymbol{\epsilon_G}\}^T[\boldsymbol{S}]\, dV = \int_V \{\delta\boldsymbol{u}\}^T\{\boldsymbol{\Omega}\}\, dV + \int_A \{\delta\boldsymbol{u}\}^T\{\boldsymbol{\Phi}\}\, dA. \tag{4.51}$$

4.7 Variational Description for Finite Deformation

Theory in this section has been compiled from [11].

Total Lagrangian

Measures using the Total Lagrangian (TL) formulation as basis uses the initial configuration as reference configuration. The strain measure normally associated with the TL formulation is the Green strain, and the stress measure the PK2 stress tensor. With this formulation, the increments in both strains and stresses are simply the difference between the current and next configuration:

$$\{d\boldsymbol{\epsilon}_G^{n+1}\} = \{\boldsymbol{\epsilon}_G^{n+1}\} - \{\boldsymbol{\epsilon}_G^{n}\}, \tag{4.52}$$

$$[d\boldsymbol{S}^{n+1}] = [\boldsymbol{S}^{n+1}] - [\boldsymbol{S}^{n}], \tag{4.53}$$

where

$$[d\boldsymbol{S}^{n+1}] = \{d\boldsymbol{\epsilon}_G^{n+1}\}[\boldsymbol{C}_T] \tag{4.54}$$

as in Section 5.3.

[2] The symbol δ will be used to indicate virtual quantities.

Updated Lagrangian Formulation

The Updated Lagrangian formulation (UL) uses the current configuration as reference configuration. The total strains and stresses are hence the true strains and the Cauchy stresses, assumed that the deformations from one configuration to the next are small. However, the strain increments must be expressed as the Green strain and PK2 stress with the current configuration as reference configuration because the UL formulation does not take into account geometric stiffness effects. The incremental stress-strain relation then becomes

$$[d\boldsymbol{S}_n^{n+1}] = \{d\boldsymbol{\epsilon}_{G,n}^{n+1}\}[\boldsymbol{C}_T], \tag{4.55}$$

where the lower index n indicates that $\boldsymbol{S}$ and $\boldsymbol{\epsilon}_G$ is referred to the current configuration. The total stress is the Cauchy stress [3] from the previous configuration added to the stress increment:

$$[\boldsymbol{S}_n^{n+1}] = [\boldsymbol{\sigma}^n] + [d\boldsymbol{S}_n^{n+1}]. \tag{4.56}$$

Malvern [11] states that this stress measure is approximately equal to the Cauchy stress measure with the configuration $n+1$ as reference, and transformation from the PK2 to the Cauchy stress tensor is therefore not necessary:

$$[\boldsymbol{S}_n^{n+1}] \approx [\boldsymbol{\sigma}^{n+1}]. \tag{4.57}$$

4.8 Finite Element Equations

Theory in this section is compiled from [7].

4.8.1 Finite Element Formulations for the Total Lagrangian Formulation

The strain and stress measures for a TL formulation can be expressed as:

$$\{\boldsymbol{S}\} = \lfloor S_{XX} \quad S_{YY} \quad S_{ZZ} \quad S_{XY} \quad S_{YZ} \quad S_{ZX} \rfloor^T, \tag{4.58}$$

$$\{\boldsymbol{\epsilon}_G\} = \lfloor \epsilon_{GXX} \quad \epsilon_{GYY} \quad \epsilon_{GZZ} \quad 2\epsilon_{GXY} \quad 2\epsilon_{GYZ} \quad 2\epsilon_{GZX} \rfloor^T. \tag{4.59}$$

The doubling of the shear components is done to enable this formulation.

The following steps are done to derive an expression for the element stiffness matrices; the *material stiffness* $[\boldsymbol{K}_M^{AB}]$ and the *geometric stiffness* $[\boldsymbol{K}_G^{AB}]$. The indices AB denotes the stiffness in node A due to the influence from node B.

The material stiffness of an element is derived from the expression for internal virtual work in Equation (4.51) and may also be called the elastic stiffness. The integrand can in combination with Equations (4.58) and (4.59) be expressed as

$$\delta\epsilon_{G,IJ}S_{IJ} = \{\delta\boldsymbol{\epsilon_G}\}^T\{\boldsymbol{S}\}. \tag{4.60}$$

[3] It has been chosen not to include lower indices for the Cauchy stress tensor because the Cauchy stress measure implies that it is the current configuration that is referred to, and this is shown by the upper index.

By combining Equations (4.9) and (4.15), the variation of the Green strain can be expressed as

$$\{\delta\boldsymbol{\epsilon}_G\} = \begin{Bmatrix} F_{iX}\delta u_{i,X} \\ F_{iY}\delta u_{i,Y} \\ F_{iZ}\delta u_{i,Z} \\ F_{iX}\delta u_{i,Y} + F_{iY}\delta u_{i,X} \\ F_{iY}\delta u_{i,Z} + F_{iZ}\delta u_{i,Y} \\ F_{iZ}\delta u_{i,X} + F_{iX}\delta u_{i,Z} \end{Bmatrix}. \tag{4.61}$$

Solid elements

For an element with three degrees of freedom in each node, the variation of the Green strain in matrix form can in combination with the definition of the displacements as in Equation (2.3) for the traditional FEA be expressed as

$$\begin{aligned} \{\delta\boldsymbol{\epsilon}_G\} &= \begin{bmatrix} F_{xX}N_{A,X} & F_{yX}N_{A,X} & F_{zX}N_{A,X} \\ F_{xY}N_{A,Y} & F_{yY}N_{A,Y} & F_{zY}N_{A,Y} \\ F_{xZ}N_{A,Z} & F_{yZ}N_{A,Z} & F_{zZ}N_{A,Z} \\ F_{xX}N_{A,Y} + F_{xY}N_{A,X} & F_{yX}N_{A,Y} + F_{yY}N_{A,X} & F_{zX}N_{A,Y} + F_{zY}N_{A,X} \\ F_{xY}N_{A,Z} + F_{xZ}N_{A,Y} & F_{yY}N_{A,Z} + F_{yZ}N_{A,Y} & F_{zY}N_{A,Z} + F_{zZ}N_{A,Y} \\ F_{xZ}N_{A,X} + F_{xX}N_{A,Z} & F_{yZ}N_{A,X} + F_{yX}N_{A,Z} & F_{zZ}N_{A,X} + F_{zX}N_{A,Z} \end{bmatrix} \begin{Bmatrix} \delta D_x^A \\ \delta D_y^A \\ \delta D_z^A \end{Bmatrix} \\ &= [\hat{\boldsymbol{B}}_A]\{\delta\boldsymbol{D}^A\}. \end{aligned} \tag{4.62}$$

The difference for the isogeometric concepts only includes substitution of the interpolation functions with the basis functions, $R_{a,b,c}^{p,q,r}$. For all definitions, it is important to have control over which space the functions are defined in. The NURBS basis functions are defined in the parameter space, hence the Jacobian matrix must also be included. By expressing the stresses in terms of strains and the tangential material matrix, see Equation (5.33), the material stiffness matrix may be obtained by the formula

$$\left[\boldsymbol{K}_M^{AB}\right] = \int_V [\hat{\boldsymbol{B}}_A]^T[\boldsymbol{C}_T][\hat{\boldsymbol{B}}_B]\, dV. \tag{4.63}$$

The formula for the geometric stiffness matrix is given as

$$\left[\boldsymbol{K}_G^{AB}\right] = K_G^{AB}\lceil \boldsymbol{I} \rfloor \quad \text{where } \lceil \boldsymbol{I} \rfloor = \lceil 1^{AB} \rfloor \text{ and } K_G^{AB} = \int_V N_{A,I}S_{IJ}N_{B,J}\, dV. \tag{4.64}$$

The resulting stiffness matrix is called the *tangent stiffness* matrix:

$$\left[\boldsymbol{K}_T^{AB}\right] = \left[\boldsymbol{K}_M^{AB}\right] + \left[\boldsymbol{K}_G^{AB}\right], \tag{4.65}$$

and is the derivative of the internal forces w.r.t. the displacements:

$$\left[\boldsymbol{K}_T^{AB}\right] = \left[\frac{\partial \boldsymbol{R}_A^{int}}{\partial \boldsymbol{D}_B}\right]. \tag{4.66}$$

In addition, if the external forces are non-conservative, the tangent stiffness also includes a term derived from the derivative of the external forces with respect to the displacements:

$$\left[\boldsymbol{K}_T^{AB}\right] = \left[-\frac{\partial \boldsymbol{R}_A^{res}}{\partial \boldsymbol{D}_B}\right] = \left[\frac{\partial \boldsymbol{R}_A^{int}}{\partial \boldsymbol{D}_B}\right] - \left[\frac{\partial \boldsymbol{R}_A^{ext}}{\partial \boldsymbol{D}_B}\right] = \left[\boldsymbol{K}_M^{AB}\right] + \left[\boldsymbol{K}_G^{AB}\right] + \left[\boldsymbol{K}_l^{AB}\right], \tag{4.67}$$

where the external forces are expressed as[4]

$$\{\boldsymbol{R}_A^{ext}\} = \int_V [\boldsymbol{N}_A]^T \{\boldsymbol{\Omega}\}\, dV + \int_A [\boldsymbol{N}_A]^T \{\boldsymbol{\Phi}\}\, dA. \tag{4.68}$$

The internal forces are expressed as

$$\left\{\boldsymbol{R}_A^{int}\right\} = \int_V [\hat{\boldsymbol{B}}_A]^T \{\boldsymbol{S}\}\, dV. \tag{4.69}$$

An advantage of this formulation is that the derivation of the tangent stiffness matrix is straight forward. However, $[\hat{\boldsymbol{B}}_A]$ is full and therefore much more comprehensive than for a linear formulation. It is desirable to obtain a strain-displacement matrix that is as simple as possible to reduce computational time, since this computation must be done for each node in each element.

Shell elements

The definition of the deformation of shell elements, differs some from the solid definition, as shown in Equation (2.10). Equation (4.61) will in addition to the translational degrees of freedom include the rotational degrees of freedom and will make the expression for the variation of the Green strain far more complex. The details of the derivation may be found in [5], but also culminates in

$$\{\delta \boldsymbol{\epsilon}_G\} = [\hat{\boldsymbol{B}}_A]\{\delta \boldsymbol{D}^A\}, \tag{4.70}$$

however, now with more complex elements of $[\hat{\boldsymbol{B}}_A]$.

4.8.2 Finite Element Formulation in the Current Configuration

To derive the element stiffness matrices with the current configuration as reference, all the terms referred to the initial configuration must be transformed to refer to the current configuration. This includes the stresses, strains and integrals.

The transformation from initial to current configuration is done by the chain rule:

$$\frac{\partial(\,\cdot\,)}{\partial X_I} = \frac{\partial(\,\cdot\,)}{\partial x_i}\frac{\partial x_i}{\partial X_I} = \frac{\partial(\,\cdot\,)}{\partial x_i} F_{iI}, \tag{4.71}$$

and the relation between a volume element in the current and initial configuration is as given in Equation (4.21).

[4] The integration surface also includes the shell edges. This applies to all surface integrals unless other is stated.

The *push forward* of the variation of the strains is used to derive the expression for the strains in current configuration:

$$\delta E_{IJ} = \frac{1}{2}\left(\delta u_{i,j} + \delta u_{j,i}\right) F_{iI} F_{jJ} = \delta\epsilon_{ij} F_{iI} F_{jJ}. \tag{4.72}$$

By this equation it is shown that the strain in deformed configuration, ϵ_{ij}, is in the same form as the engineering strain with respect to the updated coordinates. This means that $\{\delta\boldsymbol{\epsilon}\} = \lfloor \delta\epsilon_{xx} \quad \delta\epsilon_{yy} \quad \delta\epsilon_{zz} \quad 2\delta\epsilon_{xy} \quad 2\delta\epsilon_{yz} \quad 2\delta\epsilon_{zx} \rfloor^T$ can be expressed as

$$\{\delta\boldsymbol{\epsilon}\} = [\boldsymbol{B}_A]\{\delta\boldsymbol{D}^A\}. \tag{4.73}$$

Solid elements

For solid elements $[\boldsymbol{B}_A]$ is identical to the small deformations strain-displacement matrix:

$$[\boldsymbol{B}_A] = \begin{bmatrix} N_{A,x} & 0 & 0 \\ 0 & N_{A,y} & 0 \\ 0 & 0 & N_{A,z} \\ N_{A,y} & N_{A,x} & 0 \\ 0 & N_{A,z} & N_{A,y} \\ N_{A,z} & 0 & N_{A,x} \end{bmatrix}. \tag{4.74}$$

Also for the current configuration, the interpolation functions are substituted with the NURBS basis functions to represent an isogeometric formulation.

The material stiffness is derived from the incremental form of the equilibrium equation given in Section 4.6.2, where the stress increment is as in Equation (4.54) and the volume transformation as in Equation (4.21). This gives:

$$\int_V \delta E_{IJ} C_{IJKL} \Delta E_{KL}\, dV = \int_V \delta\epsilon_{ij} F_{iI} F_{jJ} C_{IJKL} F_{kK} F_{lL} \Delta\epsilon_{kl}\, dV = \int_v \delta\epsilon_{ij} c_{ijkl} \Delta\epsilon_{kl}\, dv, \tag{4.75}$$

where

$$c_{ijkl} = \frac{1}{J} F_{iI} F_{jJ} F_{kK} F_{lL} C_{IJKL}. \tag{4.76}$$

By substituting the expression for the strains and writing the equation on matrix form, the expression for the material stiffness is obtained:

$$\left[\boldsymbol{K}_M^{AB}\right] = \int_v [\boldsymbol{B}_A]^T [\boldsymbol{c}_T] [\boldsymbol{B}_B]\, dv. \tag{4.77}$$

The geometric stiffness is obtained by transforming Equation (4.64):

$$\left[\boldsymbol{K}_G^{AB}\right] = K_G^{AB} \lceil \boldsymbol{I} \rfloor, \tag{4.78}$$

where

$$K_G^{AB} = \int_V N_{A,I} S_{IJ} N_{B,J}\, dV = \int_V N_{A,i} F_{iI} J \frac{1}{F_{iI} F_{jJ}} \sigma_{ij} N_{B,j} F_{jJ}\, dV = \int_v N_{A,i} \sigma_{ij} N_{B,j}\, dv. \tag{4.79}$$

The internal forces are expressed as

$$\left\{\boldsymbol{R}_A^{int}\right\} = \int_v [\boldsymbol{B}_A]^T \{\boldsymbol{\sigma}\}\, dv. \tag{4.80}$$

The total tangential stiffness matrix does also in the current configuration include the stiffness component due to follower forces:

$$\left[\boldsymbol{K}_T^{AB}\right] = \left[-\frac{\partial \boldsymbol{R}_A^{res}}{\partial \boldsymbol{D}_B}\right] = \left[\frac{\partial \boldsymbol{R}_A^{int}}{\partial \boldsymbol{D}_B}\right] - \left[\frac{\partial \boldsymbol{R}_A^{ext}}{\partial \boldsymbol{D}_B}\right] = \left[\boldsymbol{K}_M^{AB}\right] + \left[\boldsymbol{K}_G^{AB}\right] + \left[\boldsymbol{K}_l^{AB}\right], \tag{4.81}$$

where the external forces are expressed as

$$\{\boldsymbol{R}_A^{ext}\} = \int_v [\boldsymbol{N}_A]^T \{\boldsymbol{\omega}\}\, dv + \int_a [\boldsymbol{N}_A]^T \{\boldsymbol{\phi}\}\, da. \tag{4.82}$$

As mentioned in the previous section, a simple form of the strain-displacement matrix is desirable. For a formulation in the current configuration, the strain-displacement matrix, $[\boldsymbol{B}_A]$, has a simple form, in fact it is identical to the small deformations strain-displacement matrix which means that half of its entries are zero, and hence can be neglected in a product including this matrix. This is utilized when computing the tangential stiffness matrix and the internal forces and reduces the computational time, and thus the cost of the analysis. A disadvantage with this approach is that the tangential material matrix must be transformed to the current configuration for each node, each increment and each iteration.

Shell elements

The definition of the deformation of shell elements given in Equation (2.10) gives the expression for $[\boldsymbol{B}_A]$ [5]

$$[\boldsymbol{B}_A] = \begin{bmatrix} \boldsymbol{\Gamma} & 0 & 0 \\ 0 & \boldsymbol{\Gamma} & 0 \\ 0 & 0 & \boldsymbol{\Gamma} \end{bmatrix} \begin{bmatrix} N_{A,\xi} & 0 & 0 & -\zeta t_A N_{A,\xi} l_{2A}/2 & \zeta t_A N_{A,\xi} l_{1A}/2 \\ N_{A,\eta} & 0 & 0 & -\zeta t_A N_{A,\eta} l_{2A}/2 & \zeta t_A N_{A,\eta} l_{1A}/2 \\ 0 & 0 & 0 & -t_A N_A l_{2A}/2 & t_A N_A l_{1A}/2 \\ 0 & N_{A,\xi} & 0 & -\zeta t_A N_{A,\xi} m_{2A}/2 & \zeta t_A N_{A,\xi} m_{1A}/2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -t_A N_A n_{2A}/2 & t_A N_A n_{1A}/2 \end{bmatrix}, \tag{4.83}$$

with $[\boldsymbol{\Gamma}]$ as in Equation (2.20).

Chapter 5

Solution of Finite Element Analyses

The solution of a FEA is obtained by solving the equilibrium equations from Section 4.6 with respect to the unknown displacements. Both the internal forces and the tangent stiffness matrix are expressed as in Section 4.8. The global matrices are assembled in a standard manner according to a connectivity table.

For a linear analysis, the solution can be found directly, solving the system stiffness relationship:

$$[\boldsymbol{K}]\{\boldsymbol{D}\} = \{\boldsymbol{R}\} \rightarrow \{\boldsymbol{D}\} = [\boldsymbol{K}]^{-1}\{\boldsymbol{R}\}, \tag{5.1}$$

$\{\boldsymbol{R}\}$ being the external forces and $[\boldsymbol{K}]$ the stiffness matrix. A nonlinear analysis must solve the nonlinear equations

$$\{\boldsymbol{R}^{res}\} = \{\boldsymbol{R}\} = \{\boldsymbol{R}^{ext}\} - \{\boldsymbol{R}^{int}\}. \tag{5.2}$$

The iterative expression may be obtained through a Taylor series expansion, see Section 5.1, of the equilibrium equations:

$$\left\{\boldsymbol{R}_n^i\right\} = \left[\boldsymbol{K}_{Tn}^i\right]\left\{\Delta\boldsymbol{D}_n^i\right\}. \tag{5.3}$$

As mentioned in Section 4.8 it is important to recognize which space the matrices are assembled in. The transformation between the spaces is different from the traditional FEA to the IGA. To be able to numerically integrate the integrals containing $[\hat{\boldsymbol{B}}_A]$ in an IGA both the transformation from the physical to the parameter space, and also the transformation from the parameter space to the parent element must be taken into account. This will include two Jacobian matrices. These integrals can be solved using the Gauss quadrature rule [5] for solid elements and the Simpson's rule in the thickness direction for shell elements. The number of quadrature points utilizing Gauss' quadrature is usually $p+1$ within each knot in each direction. However, ongoing research is currently trying to discover the most efficient numerical rules. Hughes, Reali and Sangalli [19] give one example; the "half-point" rule. For a traditional FEA only the transformation from the physical space to the parameter space is present because the parameter space defines the parent element.

5.1 Newton-Raphson

To derive the formula for a Newton-Raphson incremental-iterative solution procedure, the Taylor series expansion of a function $f = 0$ about a point x_{i+1} must be shown:

$$f_{i+1} = f(x_{i+1}) = f_i + \left(\frac{df}{dx}\right)_i \Delta x + \left(\frac{1}{2}\frac{d^2 f}{dx^2}\right)_i (\Delta x)^2 + \text{h.o.t} = 0. \tag{5.4}$$

The equation is on residual form and is linearized so that

$$f_i = -\left(\frac{df}{dx}\right)_i \Delta x_{i+1} = -\left(\frac{df}{dx}\right)_i (x_{i+1} - x_i), \tag{5.5}$$

$$x_{i+1} = x_i + \frac{f_i}{\left(\frac{df}{dx}\right)_i}. \tag{5.6}$$

With this simplification the nonlinear equation as functions of x is transformed to a linear equation as functions of the correction of x, Δx. The residual is the difference between f_{i+1}, which is assumed to be in equilibrium with x, and the current iteration, f_i. The solution of each increment will combined result in the equilibrium path. For an unconditionally increasing equilibrium path, the Newton-Raphson method is stable, but it is not capable of tracing the equilibrium path beyond critical points, unless an increment control, like the arc-length methods described in Section 5.5, is adopted. In general the residuals will never be exactly equal to zero and a convergence criterion is necessary to determine when to end the iteration process. Different convergence criteria are presented in Section 5.6.

There are several versions of the Newton-Raphson method available. For a *full Newton-Raphson* analysis, the term $(df/dx)_i$ is updated at every increment, and the method is consequently suitable for highly nonlinear problems which include large changes in incline of the equilibrium path. This is computationally expensive, but the method has quadratic convergence rate. A *modified Newton-Raphson* method updates the mentioned term only a given number of times. The *initial stiffness method* never updates the term after it is assembled initially, whereas the most commonly used modified method updates the term once per increment. By never updating the term the computational expenses are reduced, but the method may often lead to slow convergence rates, hence it is only suitable for mildly nonlinear problems. By updating the term once, a faster convergence rate is obtained.

5.2 Numerical Integration

The integrals in the equilibrium equations may be solved by a numerical integration rule. Two rules will be described in this thesis; Gauss' quadrature rule and Simpson's rule. Gauss' quadrature rule is the most common numerical integration rule, whereas Simpson's rule is often used if the values of the variables are required on the borders of the element integrated. Gauss' quadrature rule [5] solves the integrals by the formula

$$I = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \Phi(\xi,\eta,\zeta)\, d\xi d\eta d\zeta \approx \sum_i \sum_j \sum_k W_i W_j W_k \Phi(\xi_i,\eta_j,\zeta_k). \tag{5.7}$$

Here I represents the solution of the integral, W_i, W_j and W_k are weight factors and ξ_i, η_j and ζ_k are the sampling point locations. The sampling points are located using different fractions of the element size, and are never located at the outermost surfaces of an element. As a consequence, the highest values of for instance plastic strain, may not be detected as this often is located furthest away from the neutral axis of an element. Whether the expression is integrated with full or reduced integration is determined by the analyst. Simpson's rule [20] divides the integral into n equally sized subdivisions, n being an even number, and is used in the through thickness direction in the integration of shells in this thesis. The approximation of the integral is

$$I = \int_{-1}^{1} \Phi(\zeta)\, d\zeta \approx \frac{h}{3}\left[\Phi_0 + 4\Phi_1 + 2\Phi_2 + 4\Phi_3 + 2\Phi_4 + \cdots + 2\Phi_{n-2} + 4\Phi_{n-1} + \Phi_n\right], \tag{5.8}$$

where

$$h = \zeta_k - \zeta_{k-1}. \tag{5.9}$$

5.3 Return Mapping

For elastic-plastic materials, a fundamental problem occurs when strains and stresses are to be updated; how much of the strain increment is elastic and how much is plastic? To be able to determine the ratio between the two, a return mapping algorithm can be applied.

Starting from the last known converged solution, the increment in the displacements is obtained by the equation

$$[\boldsymbol{K}_{T,n-1}]\{\Delta\boldsymbol{D}_n\} = \{\boldsymbol{R}_n^{res}\}, \tag{5.10}$$

where the residual force vector is

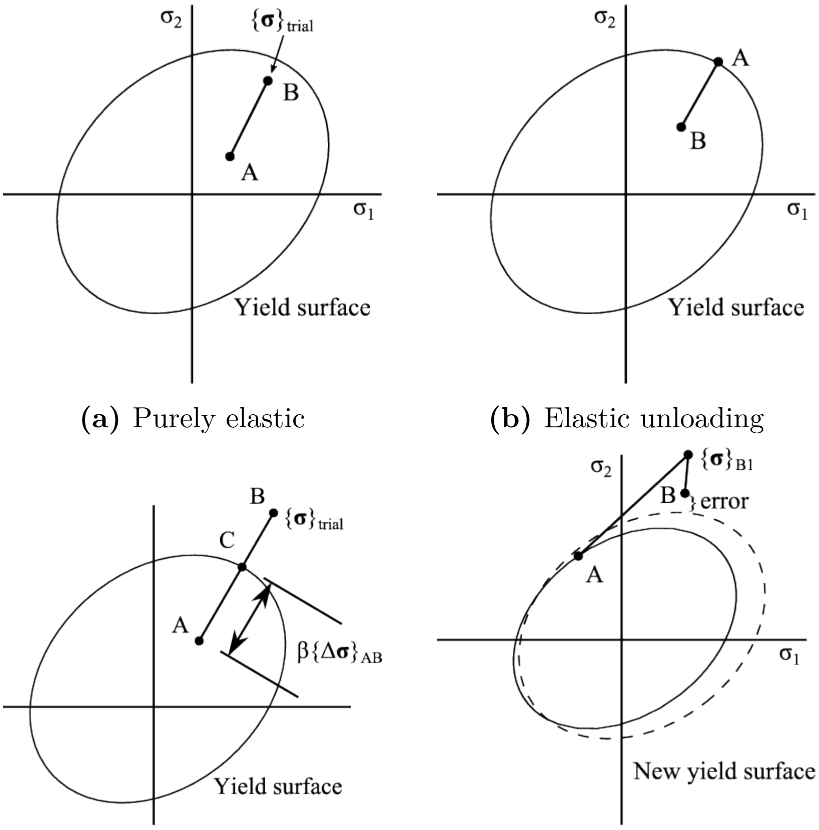
$$\{\boldsymbol{R}_n^{res}\} = \left\{\boldsymbol{R}_n^{ext}\right\} - \left\{\boldsymbol{R}_{n-1}^{int}\right\} = \left\{\boldsymbol{R}_n^{ext}\right\} - \sum_{e=1}^{N_{els}} \left\{\boldsymbol{r}_{n-1}^{int}\right\}_e, \tag{5.11}$$

and the expressions for $[\boldsymbol{K}_{T,n-1}]$ and $\left\{\boldsymbol{r}_{n-1}^{int}\right\}$ depend on the reference configuration chosen and can be found in Section 4.8. e refers to the element number and N_{els} is the total number of elements. The strain increment is computed at each integration point in each finite element and is obtained from

$$\{\Delta\boldsymbol{\epsilon}_n\} = [\tilde{\boldsymbol{B}}_{a,n-1}]\{\Delta\boldsymbol{d}_n^a\}, \tag{5.12}$$

where $[\tilde{\boldsymbol{B}}_{a,n-1}]$ also depends on the reference configuration chosen and can be found in Section 4.8.

There are four possible stress state changes for an incremental step as shown in Figure 5.1 [21].



(a) Purely elastic

(b) Elastic unloading

(c) Partially plastic

(d) Purely plastic

Figure 5.1: Stress increment options

5.3.1 Increment from Elastic Stress State

For the last known converged solution, henceforth called A, there are two possibilities as to which stress state the material is in, either elastic or plastic. If A is in the elastic range, hence $Y < 0$, a first trial increment is assumed to be entirely elastic, bringing the stress state to B. The trial stress is then obtained from

$$\{\boldsymbol{\sigma}\}_{trial} = \{\boldsymbol{\sigma}\}_A + \{\Delta\boldsymbol{\sigma}\}_{AB} = \{\boldsymbol{\sigma}\}_A + [\boldsymbol{E}]\{\Delta\boldsymbol{\epsilon}\}_{AB}. \tag{5.13}$$

$[\boldsymbol{E}]$ is the elastic material matrix from Equation (4.34). For the trial stress, the yield criterion must me controlled. If $Y < 0$, the strain increment is purely elastic and $\{\boldsymbol{\sigma}\}_B = \{\boldsymbol{\sigma}\}_{trial}$, as shown in Figure 5.1a. This also applies if the increment is in the direction of unloading. If $Y > 0$, the strain increment is partially plastic and the fraction, β, of the strain increment that is elastic has to be determined, as shown in Figure 5.1c. The stress increment is now reduced so that the stress state lies on a point C on the yield surface. β is found by solving the yield criterion:

$$Y_C = Y(\{\boldsymbol{\sigma}\}_C) = 0, \tag{5.14}$$

where

$$\{\boldsymbol{\sigma}\}_C = \{\boldsymbol{\sigma}\}_A + \beta\{\Delta\boldsymbol{\sigma}\}_{AB} = \{\boldsymbol{\sigma}\}_A + \beta[\boldsymbol{E}]\{\Delta\boldsymbol{\epsilon}\}_{AB}. \tag{5.15}$$

The first approximation of β is chosen to be

$$\beta \approx \frac{Y_A}{Y_A - Y_{trial}} = \beta_1. \tag{5.16}$$

To provide more accurate solutions a secant iteration as shown in Figure 5.2 may be used, which results in the iterative β-values:

$$\beta_{i+1} = \beta_i - \frac{Y_{Ci}}{\Delta Y_{Ci}/\Delta\beta}, \quad \frac{\Delta Y_{Ci}}{\Delta\beta} = \frac{Y_{C(i-1)} - Y_{Ci}}{\beta_{i-1} - \beta_i}. \tag{5.17}$$

With the constitutive relations given in Section 4.5.2, the equation system to solve is then:

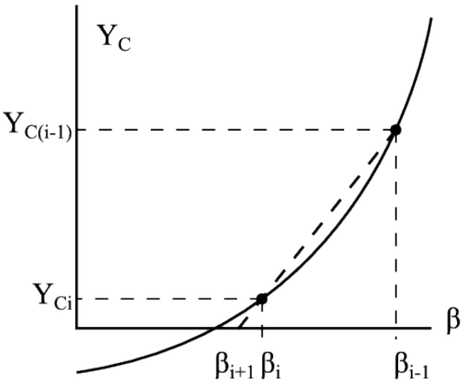


Figure 5.2: Secant iteration

$$\{\boldsymbol{\sigma}\}_{Ci} = \{\boldsymbol{\sigma}\}_A + \beta_i[\boldsymbol{E}]\{\Delta\boldsymbol{\epsilon}\}_{AB}, \tag{5.18}$$

$$p_{Ci} = p_A + (1-\beta_i)\{\Delta\boldsymbol{\epsilon}\}_{AB}, \tag{5.19}$$

$$Y_{Ci} = \sigma_{eq}(\{\boldsymbol{\sigma}\}_{Ci}) - (\sigma_0 + A_{JC}p_{Ci}^{n_{JC}}). \tag{5.20}$$

p_A is the equivalent plastic strain stored in the element in stress state A. The iterations are terminated with an appropriate convergence criterion.

5.3.2 Increment from Plastic Stress State

This state is depicted in Figure 5.1d. To determine the size of the plastic strain increment,

$$\{\Delta\boldsymbol{\epsilon}^p\}_{AB} = \left\{\frac{\partial Y}{\partial\boldsymbol{\sigma}}\right\}\Delta\lambda_p, \tag{5.21}$$

the size of the plastic multiplier, $\Delta\lambda_p$, must be determined. This is done using the property given in Equation (4.44) combined with the definition of the yield function from Section 4.5.2 which gives:

$$dY = \left\{\frac{\partial Y}{\partial\boldsymbol{\sigma}}\right\}^T \{d\boldsymbol{\sigma}\} + \frac{\partial Y}{\partial W_p}dW_p = 0. \tag{5.22}$$

Further calculation gives expression for $\lfloor \boldsymbol{P}_\lambda \rfloor$ which relates the plastic multiplier to the total strain increment:

$$\Delta\lambda_p = \lfloor \boldsymbol{P}_\lambda \rfloor \{\Delta\boldsymbol{\epsilon}\}_{AB}, \tag{5.23}$$

$$\lfloor \boldsymbol{P}_\lambda \rfloor = \frac{\left\{\frac{\partial Y}{d\boldsymbol{\sigma}}\right\}^T [\boldsymbol{E}]}{\left\{\frac{\partial Y}{d\boldsymbol{\sigma}}\right\}^T [\boldsymbol{E}] \left\{\frac{\partial Y}{d\boldsymbol{\sigma}}\right\} - \frac{\partial Y}{dW_p}\{\boldsymbol{\sigma}\}^T \left\{\frac{\partial Y}{d\boldsymbol{\sigma}}\right\}}. \tag{5.24}$$

As initial step, $\lfloor \boldsymbol{P}_\lambda \rfloor$ is evaluated at A. If then $\Delta\lambda_p < 0$, the increment equals elastic unloading and the stress at B is

$$\{\boldsymbol{\sigma}\}_B = \{\boldsymbol{\sigma}\}_A + [\boldsymbol{E}]\{\Delta\boldsymbol{\epsilon}\}_{AB}. \tag{5.25}$$

On the other hand, if $\Delta\lambda_p > 0$, the increment is entirely plastic and the development of the yield surface must be taken into account. The plastic strain increment is then expressed as

$$\{\Delta\boldsymbol{\epsilon}^p\}_{AB} = \left[(1-\gamma)\left\{\frac{\partial Y}{\partial\boldsymbol{\sigma}}\right\}_A \lfloor \boldsymbol{P}_\lambda \rfloor_A + \gamma\left\{\frac{\partial Y}{\partial\boldsymbol{\sigma}}\right\}_B \lfloor \boldsymbol{P}_\lambda \rfloor_B\right] \{\Delta\boldsymbol{\epsilon}\}_{AB} \tag{5.26}$$

The value of γ is chosen by the analyst. For $\gamma = 0$ the method results in forward Euler, for $\gamma = 1$ the result is backward Euler. To improve the result, the above equation is implemented in predictor-corrector form:

$$\{\Delta\boldsymbol{\epsilon}^p\}_{AB1} = \left\{\frac{\partial Y}{\partial\boldsymbol{\sigma}}\right\}_A \lfloor \boldsymbol{P}_\lambda \rfloor_A \{\Delta\boldsymbol{\epsilon}\}_{AB}, \tag{5.27}$$

$$\{\Delta\boldsymbol{\sigma}\}_{AB1} = [\boldsymbol{E}]\left(\{\Delta\boldsymbol{\epsilon}\}_{AB} - \{\Delta\boldsymbol{\epsilon}^p\}_{AB1}\right), \tag{5.28}$$

$$\{\boldsymbol{\sigma}\}_{B1} = \{\boldsymbol{\sigma}\}_A + \{\Delta\boldsymbol{\sigma}\}_{AB1}. \tag{5.29}$$

These values are used to determine a new value of $\lfloor \boldsymbol{P}_\lambda \rfloor = \lfloor \boldsymbol{P}_\lambda \rfloor_{Bi}$ which then is used to calculate new values of the stress:

$$\{\Delta\boldsymbol{\epsilon}^p\}_{ABi} = \left\{\frac{\partial Y}{\partial\boldsymbol{\sigma}}\right\}_{Bi} \lfloor \boldsymbol{P}_\lambda \rfloor_{Bi} \{\Delta\boldsymbol{\epsilon}\}_{AB}, \tag{5.30}$$

$$\{\Delta\boldsymbol{\sigma}\}_{ABi} = [\boldsymbol{E}]\left(\{\Delta\boldsymbol{\epsilon}\}_{AB} - \{\Delta\boldsymbol{\epsilon}^p\}_{ABi}\right), \tag{5.31}$$

$$\{\boldsymbol{\sigma}\}_{Bi} = \{\boldsymbol{\sigma}\}_A + \{\Delta\boldsymbol{\sigma}\}_{ABi}, \tag{5.32}$$

$$[\boldsymbol{c}_{TBi}] = [\boldsymbol{E}]\left(\lceil \boldsymbol{I} \rfloor - \left\{\frac{\partial Y}{\partial\boldsymbol{\sigma}}\right\}_{Bi} \lfloor \boldsymbol{P}_\lambda \rfloor_{Bi}\right). \tag{5.33}$$

The form of the material tangent matrix is equal to the linear material matrix:

$$[\boldsymbol{c}_T] = \begin{bmatrix} c_{Txxxx} & c_{Txxyy} & c_{Txxzz} & 0 & 0 & 0 \\ c_{Tyyxx} & c_{Tyyyy} & c_{Tyyzz} & 0 & 0 & 0 \\ c_{Tzzxx} & c_{Tzzyy} & c_{Tzzzz} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{Txyxy} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{Tyzyz} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{Tzxzx} \end{bmatrix}. \tag{5.34}$$

5.4 Imposing Constraints

There are several possibilities when imposing constraints, each with its pros and cons. The three most commonly used methods of imposing constraints are by *transformation*, by *Lagrange multipliers* and by *penalty functions* [22].

5.4.1 Transformation

When using transformation to impose constraints, the nodes chosen as slave nodes are eliminated by rearranging the order of the degrees of freedom and making the slave nodes dependent on the master nodes. When a node is shared by more than two elements, the node chosen as slave node may only be slave of one other node. A node may be master node for several slave nodes, and a node may be master to a slave node which acts as a master node in relation to a third node. The relations described above can be explained by help of Figure 5.3. Node $A1$ may be master for both $A2$ and $A3$, or it may be master for $A2$ that is master for $A3$. It is important to notice that both $A1$ and $A2$ can not be master for $A3$. A master node is a node where the degrees of freedom are not constrained.

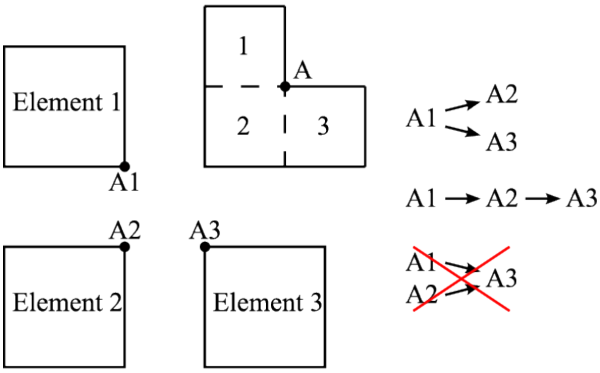


Figure 5.3: Examples of constraint relations

Generally the constraint equation yields

$$\{\boldsymbol{C}\} = [\boldsymbol{G}]\{\boldsymbol{D}\} - \{\boldsymbol{Q}\} = \{0\} \Leftrightarrow [\boldsymbol{G}]\{\boldsymbol{D}\} = \{\boldsymbol{Q}\}. \tag{5.35}$$

$[\boldsymbol{G}]$ is a matrix containing the constraint gradient coefficients of the constraint equations. The size of $[\boldsymbol{G}]$ is $n_c \times n_{dof}$ where n_c is the number of constraint equations and n_{dof} is the number of degrees of freedom included in the constraint equations. $\{\boldsymbol{Q}\}$ is a vector containing constants. When the degrees of freedom are arranged according to the slave/master-distribution ($\{\boldsymbol{D}_s\}$ and $\{\boldsymbol{D}_m\}$), and the new vector of degrees of freedom is inserted into the global equation,

$$[\boldsymbol{K}]\{\boldsymbol{D}\} = \{\boldsymbol{R}\}, \tag{5.36}$$

the new equation yields

$$[\boldsymbol{K}^*_{mm}]\{\,\boldsymbol{D}_m\} = \{\boldsymbol{R}^*_m\}\,, \tag{5.37}$$

where m denotes the features related to the master nodes and

$$[\boldsymbol{K}^*_{mm}] = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{G}_s^{-1}\boldsymbol{G}_m \end{bmatrix}^T [\boldsymbol{K}] \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{G}_s^{-1}\boldsymbol{G}_m \end{bmatrix} \quad \text{and} \tag{5.38}$$

$$\{\boldsymbol{R}^*_m\} = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{G}_s^{-1}\boldsymbol{G}_m \end{bmatrix}^T \left(\{\boldsymbol{R}\} - [\boldsymbol{K}] \begin{Bmatrix} \boldsymbol{0} \\ \boldsymbol{G}_s^{-1}\boldsymbol{Q} \end{Bmatrix} \right). \tag{5.39}$$

The constraints imposed by transformation in this thesis will only be linear, hence, only linear constraints will be described.

In FE programs, the constraints are imposed on element level, thus, the equations described above will be local to each element. The equilibrium equation will as an example yield

$$[\boldsymbol{k}^*_{mm}]\{\,\boldsymbol{d}_m\} = \{\boldsymbol{r}^*_m\}\,. \tag{5.40}$$

5.4.2 Lagrange Multipliers

The general constraint equation given in Equation (5.35) also applies for the method of Lagrange multipliers. However, the constraints are imposed by augmenting the potential energy with the energy from the constraints:

$$\Pi_p^* = \Pi_p + \{\boldsymbol{\lambda}\}^T\{\boldsymbol{C}\}. \tag{5.41}$$

Because the Lagrange multiplier method imposes the constraints exactly, the potential energy remains constant due to $\{\boldsymbol{C}\} = \{\boldsymbol{0}\}$. The initial potential energy is expressed as

$$\Pi_p = \frac{1}{2}\{\boldsymbol{D}\}^T[\boldsymbol{K}]\{\boldsymbol{D}\} - \{\boldsymbol{D}\}^T\{\boldsymbol{R}\}, \tag{5.42}$$

and by making the potential energy stationary, the relations below are derived:

$$[\boldsymbol{K}]\{\boldsymbol{D}\} + [\boldsymbol{G}]^T\{\boldsymbol{\lambda}\} = \{\boldsymbol{R}\}, \tag{5.43}$$

$$[\boldsymbol{G}]\{\boldsymbol{D}\} = \{\boldsymbol{Q}\}. \tag{5.44}$$

The matrix form of the equations yield

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{G}^T \\ \boldsymbol{G} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{D} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{R} \\ \boldsymbol{Q} \end{Bmatrix}. \tag{5.45}$$

The method of imposing constraints described above only applies to linear constraints. If the constraints are nonlinear, the constraint equations must be stated on incremental form, and the product of the matrices containing the multipliers and the constraint gradients is added to the internal force vector.

An advantage of imposing linear constraints with Lagrange multipliers in contrast to transformations is that the stiffness matrix is not altered. This saves computational time. Another advantage is that the linear constraints are imposed exactly, which eliminates a possible source of error, and that the Lagrange multipliers can be interpreted physically, e.g. as the forces necessary to obtain the constraints. Also for a static nonlinear analysis the displacement constraints will be satisfied exactly.

5.4.3 Penalty Functions

As for the Lagrage multiplier method, the penalty functions augment the potential energy:

$$\Pi_p^* = \Pi_p + \frac{1}{2}\{\boldsymbol{C}\}^T \lceil \boldsymbol{\alpha} \rfloor \{\boldsymbol{C}\}, \tag{5.46}$$

where $\lceil \boldsymbol{\alpha} \rfloor$ is the diagonal matrix containing the penalty numbers. The penalty numbers determines the magnitude of the penalty if the constraint is violated. Making the potential energy stationary gives the expressions

$$[\boldsymbol{K}^*] = [\boldsymbol{K} + \boldsymbol{G}^T \boldsymbol{\alpha} \boldsymbol{G}], \tag{5.47}$$

$$\{\boldsymbol{R}^*\} = \{\boldsymbol{R} + \boldsymbol{G}^T \boldsymbol{\alpha} \boldsymbol{Q}\}, \tag{5.48}$$

and the equation

$$[\boldsymbol{K}^*]\{\boldsymbol{D}\} = \{\boldsymbol{R}^*\} \tag{5.49}$$

needs to be solved. An advantage of the penalty method is that is does not introduce additional unknowns, as the Lagrange multipliers do. However, by defining the penalty numbers with too high value, ill-conditioning may occur, and it is the analyst responsibility to determine the values. On the other hand, if the penalty numbers are too low, the constraint will not be satisfied sufficiently.

5.5 Arc-Length Methods

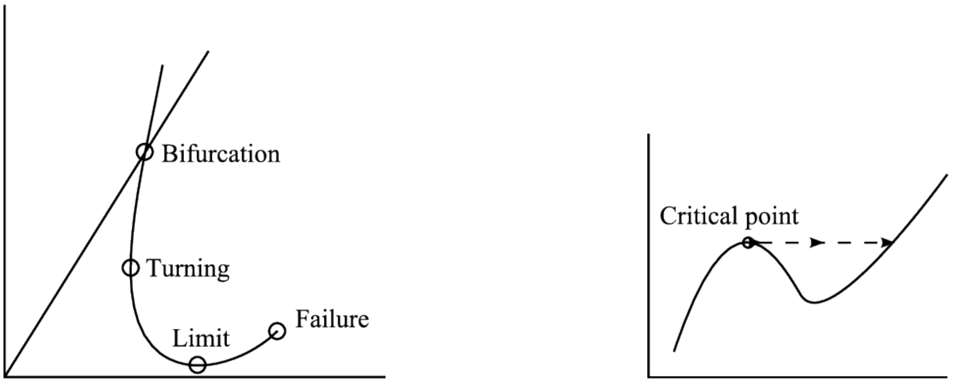
An arc-length method is a method used to trace the equilibrium path by controlling the load and displacement increments. The need for incremental control may occur if the equilibrium path includes critical points such as limit, bifurcation, turning and/or failure points as shown in Figure 5.4a. Critical points are characterized either by singular tangent stiffness matrix, vertical tangent or a sudden stop of the equilibrium path. In such cases, a solution algorithm without incremental control will not be able to follow the equilibrium path, and problems like snap-through may occur as shown in Figure 5.4b.

Arc-length methods require an additional unknown, $\Delta\lambda$, and an additional constraint equation to determine the expression for the arc-length l. The basis of the constraint equation are the vectors $\{\boldsymbol{Z}\} = \{\lambda, \{\boldsymbol{D}\}\}$ and $\{\bar{\boldsymbol{Z}}\}$. $\{\boldsymbol{Z}\}$ represents the converged solution whereas $\{\bar{\boldsymbol{Z}}\}$ is the tangent to an increment solution which varies with the type of arc-length method used. The constraint equation and expression for the arc length are:

$$\{\boldsymbol{C}(\Delta \boldsymbol{Z})\} = \left\{\boldsymbol{C}(\Delta\lambda, \Delta\{\boldsymbol{D}\}^T)\right\} = 0, \tag{5.50}$$

$$l^2 = \{\Delta \boldsymbol{D}\}^T\{\Delta \boldsymbol{D}\} + \psi^2 \Delta\lambda^2. \tag{5.51}$$

ψ may vary from 0 to ∞, where $\psi = 0$ results in displacement control and $\psi = \infty$ results in load control. This constant is included to determine the ratio of load/displacement in the increment control and to transform the term $\psi^2\Delta\lambda^2$ to the same denomination as the degrees of freedom. The different steps in an analysis using incremental control are shown in Figure 5.5. In the predictor step the load and/or displacement increment is prescribed



(a) Critical points of an equilibrium path **(b)** Example of solution obtained with solution algorithm without incremental control

Figure 5.4: Equilibrium path

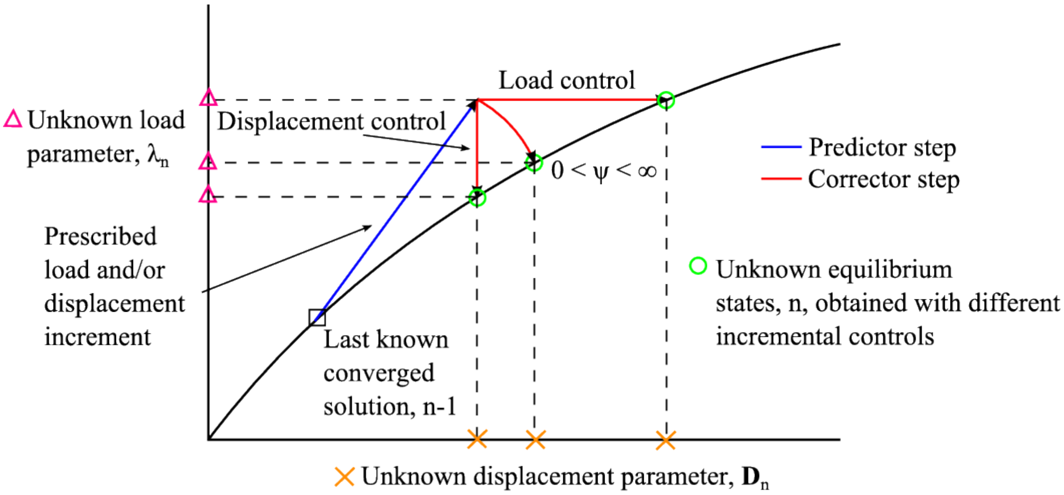

Load control
Unknown load parameter, λ_n
Displacement control
0 < ψ < ∞
Predictor step
Corrector step
Prescribed load and/or displacement increment
Unknown equilibrium states, n, obtained with different incremental controls
Last known converged solution, n-1
Unknown displacement parameter, D_n

Figure 5.5: Steps in analysis with incremental control

such that $\left\{\boldsymbol{Z}_n^0\right\} = \left(\lambda_n^0, \{\boldsymbol{D}_n^0\}\right)^T$. The increment is added to the last known converged solution on the equilibrium path, $\{\boldsymbol{Z}_{n-1}\} = (\lambda_{n-1}, \{\boldsymbol{D}_{n-1}\})^T$. The correction phase consists of Newton-Raphson iterations, see Section 5.1, to solve the force-displacement equations given in Section 4.8 on incremental form and with an augmented tangent stiffness matrix:

$$\left[\hat{\boldsymbol{K}}_{Tn}\right]\left\{\Delta\boldsymbol{Z}_n^i\right\} = \left\{\Delta\boldsymbol{R}_n^i\right\} \quad \text{and} \quad \left\{\boldsymbol{C}(\boldsymbol{Z}_n^i)\right\} = 0, \tag{5.52}$$

$$\left[\hat{\boldsymbol{K}}_{Tn}\right] = \left[\hat{\boldsymbol{K}}_T(\boldsymbol{Z}_n^{i-1})\right] = \left[\left\{\frac{\partial \boldsymbol{R}^{int}(\boldsymbol{Z}_n^{i-1})}{\partial\lambda}\right\}\left\{\frac{\partial \boldsymbol{R}^{int}(\boldsymbol{Z}_n^{i-1})}{\partial \boldsymbol{D}}\right\}\right], \tag{5.53}$$

$$\left\{\Delta\boldsymbol{R}_n^i\right\} = \left\{\boldsymbol{R}^{ext}(\boldsymbol{Z}_n^{i-1})\right\} - \left\{\boldsymbol{R}^{int}(\boldsymbol{Z}_n^{i-1})\right\}. \tag{5.54}$$

There are several possible formulations for arc-length methods. A selection of them is described below and shown in Figure 5.6.

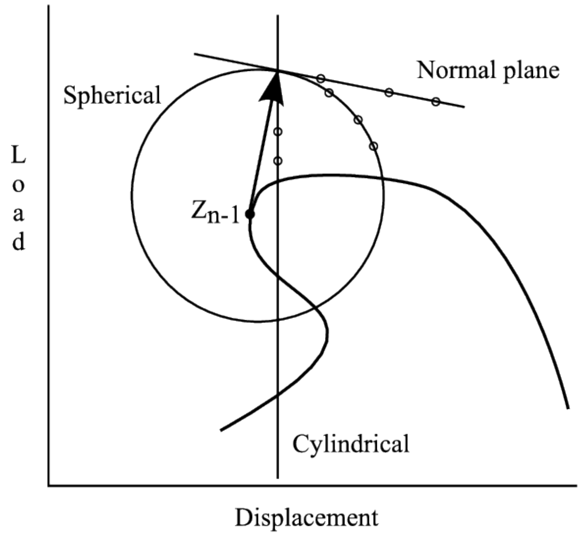


Figure 5.6: Arc-length methods

Theory is taken from [23].

5.5.1 Normal Plane Arc-Length Method

In the normal plane (NP) arc-length method the arc-length is the distance from the last converged solution, $\{\boldsymbol{Z}_{n-1}\}$, to the normal plane of the tangent solution from $\{\boldsymbol{Z}_{n-1}\}$. The constraint is expressed as

$$\left\{\bar{\boldsymbol{Z}}_{n-1}\right\}^T \left\{\boldsymbol{Z}_n^i - \boldsymbol{Z}_{n-1}\right\} - l = 0, \tag{5.55}$$

where $\{\bar{\boldsymbol{Z}}_{n-1}\}$ is the tangent to the equilibrium path at point $n-1$. The vectors include values of $\Delta\lambda$ and $\{\Delta\boldsymbol{D}\}$ which make it possible to control these quantities.

The NP arc-length method may also be updated (UNP) which means that the normal plane at all times is normal to the tangent to the equilibrium path at the i'th iteration at point n. The constraint will for the UNP method be defined as

$$\left\{\bar{\boldsymbol{Z}}_n^i\right\}^T \left\{\boldsymbol{Z}_n^i - \boldsymbol{Z}_{n-1}\right\} - l = 0. \tag{5.56}$$

5.5.2 Spherical Arc-Length Method

In a spherical arc-length method, the iteration path is a sphere. The constraint equation is expressed as

$$\left\{\boldsymbol{Z}_n^i - \boldsymbol{Z}_{n-1}\right\}^T \left\{\boldsymbol{Z}_n^i - \boldsymbol{Z}_{n-1}\right\} - l^2 = 0. \tag{5.57}$$

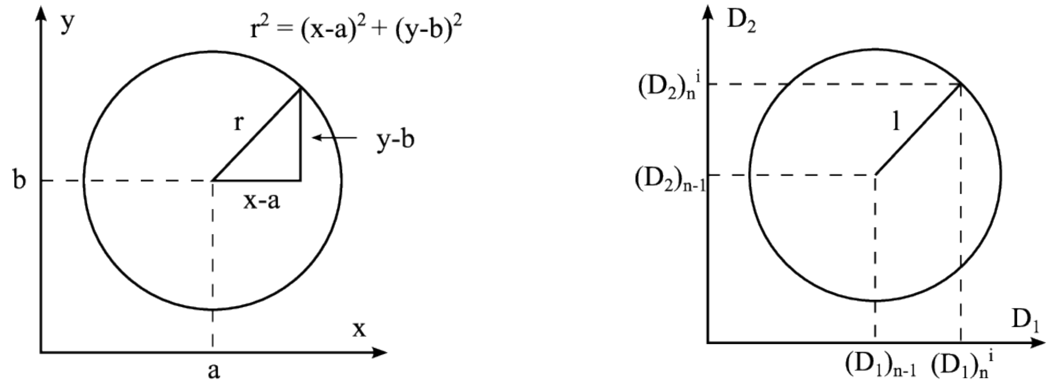
The constraint leads to a quadratic expression for $\Delta\lambda$. The $\Delta\lambda$ giving a positive angle of rotation is the one chosen [24]. This requirement is stated to avoid doubling back of the solution.

5.5.3 Cylindrical Arc-Length Method

For a cylindrical arc-length method the iteration path follows a cylinder parallel to the λ-axis. The outline of the cylinder coincides with the tangent solution from the last known converged solution. The constrain equation is:

$$\left\{\boldsymbol{D}_n^i - \boldsymbol{D}_{n-1}\right\}^T \left\{\boldsymbol{D}_n^i - \boldsymbol{D}_{n-1}\right\} - l^2 = 0. \tag{5.58}$$

The constraint equation can be explained by an example consisting of two degrees of freedom as shown in Figure 5.7. The equation for a circle is shown in Figure 5.7a and is independent of the load parameter.



(a) Equation for circle

(b) Example of analysis with two degrees of freedom

Figure 5.7: Explanation of constraint equation for cylindrical arc-length method

The cylindrical arc-length method is a displacement control method ($\psi = 0$) which means that the analyst does not have any control of the size of the load increment. Methods such as this are not able to trace the equilibrium path beyond turning points and the analyst must therefore assume that the equilibrium path has traceable geometry. This can be seen from Figure 5.6: if the tangent had started before the turning point, the vertical outline of the cylinder would not have crossed the equilibrium path.

5.6 Convergence Criteria

A convergence criterion is used to determine how close to the equilibrium path an iterated solution lies. When the solution fulfils the specified convergence criteria, the iterations are stopped and the solution is said to be converged. It is important that the convergence criteria are neither too loose nor too tight. If the criteria are too loose the results may be inaccurate, but if the criteria are too tight the analysis may be uneconomical due to too many iterations. Convergence criteria can be based on different parameters, but the most commonly used are displacements, residual forces and energy. In general, it is recommended that a convergence criterion consisting of a combination of these parameters are used.

The total value of a vector, $\|\{\boldsymbol{A}^{i+1}\}\|$, consisting of n_{dof} entries at increment n and iteration i is given as

$$\|\{\boldsymbol{A}_n^{i+1}\}\| = \frac{1}{n_{dof}}\sqrt{\sum_{k=1}^{n_{dof}} A_k^{i+1}} \tag{5.59}$$

Theory in this section is taken from [11] and [23].

5.6.1 Displacement Based Convergence Criterion

For a convergence criterion based on displacements, the displacement increment is compared to the total displacements. It is only the translational degrees of freedom that are taken into account when applying a displacement based convergence criterion. The criterion may be expresses as

$$\|\{\boldsymbol{D}_n^{i+1}\}\| - \|\{\boldsymbol{D}_n^{i}\}\| \leq \varepsilon_D \|\{\boldsymbol{D}_n^{i+1}\}\| \tag{5.60}$$

where ε_D is a constant chosen by the analyst and normally ranges from 10^{-2} to 10^{-6}. A displacement based convergence criterion may mislead to a slow convergence rate and is in general not advisable, at least not alone.

5.6.2 Residual Force Based Convergence Criterion

The formulation of a residual force based convergence criterion is similar to the displacement based criterion. The increment in the residual forces is compared to the size of the residual forces at increment $i + 1$.. As for the displacement criterion, the residual force criterion may be expressed as

$$\|\{\boldsymbol{R}_n^{i+1}\}\| - \|\{\boldsymbol{R}_n^{i}\}\| \leq \varepsilon_F \|\{\boldsymbol{R}_n^{i+1}\}\|. \tag{5.61}$$

It is only the force components that are taken into account when calculating $\|\{\boldsymbol{R}_n\}\|$. ε_F is usually in the range 10^{-1} to 10^{-3}. Contrary to the displacement based convergence criterion, the residual force based criterion usually produces a far more reliable result because it directly controls the equilibrium of the forces.

5.6.3 Energy Based Convergence Criterion

A combination of the displacement and residual force based convergence criteria is the energy based criterion. This combines both the displacement and residual increments so that the convergence criterion is expressed as the work done over the iterative displacements:

$$\{\Delta \boldsymbol{R}_n^{i+1}\}^T \{\Delta \boldsymbol{D}_n^{i+1}\} \leq \varepsilon_E \{\Delta \boldsymbol{R}_n^{0}\}^T \{\Delta \boldsymbol{D}_n^{0}\}. \tag{5.62}$$

With this formulation both the translations and the rotational degrees of freedom are taken into account and combined with the force and moment components respectively. ε_E is usually in the range 10^{-2} to 10^{-4}.

5.7 Analysis with both Material and Geometric Nonlinearity

For an analysis with elastic-plastic material and large deformations, nonlinearity is induced both by the material and the geometry. This implies that return mapping, solved by Newton-Raphsons method, must be used to determine the plastic part of the strain increments, and Newton-Raphsons method must also be applied to solve the nonlinear force-displacement equations given in Section 4.8. With prescribed displacements, the Newton-Raphson method can be combined with the cylindrical arc-length method to control the displacement increments.

5.8 Abaqus/Standard

5.8.1 Solution Algorithm

The default nonlinear solution algorithm in Abaqus/Standard [25] is the full Newton-Raphson method described in Section 5.1. For a continuum problem including stress-strain relations, the equations are composed of terms as functions of displacement and rotation. The equations are on the form

$$[\boldsymbol{K}_T]\{\Delta\boldsymbol{D}\} = \{\boldsymbol{R}^{res}\}\,, \tag{5.63}$$

where $\{\boldsymbol{R}^{res}\}$ defines the residual forces necessary to obtain equilibrium:

$$\{\boldsymbol{R}^{res}\} = \left\{\boldsymbol{R}^{ext}\right\} - \left\{\boldsymbol{R}^{int}\right\}, \tag{5.64}$$

and $[\boldsymbol{K}_T]$ is the tangential stiffness of the system:

$$[\boldsymbol{K}_T] = \left[\frac{\partial \boldsymbol{R}^{int}}{\partial \boldsymbol{D}}\right]. \tag{5.65}$$

The motivation for choosing full Newton-Raphson in contrary to the modified Newton-Raphson methods is based on convergence rate. Nevertheless, in some cases, the tangential stiffness matrix may be difficult or impossible to determine exactly. To calculate the matrix numerically is an expensive process, which is undesirable. The full Newton-Raphson method is expensive per iteration also if the tangential stiffness matrix can be assembled analytically, because it must be assembled and inverted for every iteration. To avoid these problems, modified or approximated Newton-Raphson methods may be used. Modified methods are described in Section 5.1. Approximate Newton-Raphson methods use an approximate tangential stiffness matrix in cases where exact assembly is either impossible or unprofitable.

5.8.2 Imposing constraints

Linear constraints in Abaqus/Standard are imposed by the transformation method. Nonlinear constraints are imposed by Lagrange multipliers. Both methods are described in Section 5.4.

5.8.3 Increment Control

In most static analyses, there are no time scale, and the size of the "time" steps is determined by the necessity of accuracy in modelling the nonlinear effects. The automatic step control implemented in Abaqus/Standard [6] is based on the maximum force residuals for each iteration. The magnitude of the residuals are compared to determine if convergence is likely to appear within a certain number of iterations. The first increment size should be suggested by the analyst. If no suggestion is made, the total increment is attempted to be applied in one step. For nonlinear problems this is normally not possible, and Abaqus/Standard reduces the increment until a solution is obtained. Finding the size of the first increment requires CPU-time, which can be neglected if the analyst provide a reasonable suggestion for the initial increment.

Throughout the solution process, the iterations are stopped if the solution has not converged within 16 iterations. Abaqus/Standard then restarts the iteration process with an increment size 25% smaller than the previous. If this procedure is performed more than 5 times, the analysis is stopped. On the other hand, if two increments in a row converges within 5 iterations, the following increment size is increased by 50%.

5.8.4 Convergence

For convergence in Abaqus/Standard [6], all the entries in the residual force vector, $\{\boldsymbol{R}_n^{i+1}\}$, are controlled to be sufficiently small. The tolerance limit for the residual force vector is by default set to be $R_n^\alpha = 0.5\%$ of an average force in the structure, averaged over time:

$$\{\boldsymbol{R}_n^{i+1}\}_{max}^{\alpha} \leq R_n^\alpha \tilde{q}^\alpha, \quad \tilde{q}^\alpha \overset{\text{def}}{=} \frac{1}{N_t} \sum_{n=1}^{N_t} \bar{q}^\alpha(t|_n). \tag{5.66}$$

where n is the current increment number, i is the iteration number, α represents the degree of freedom considered, N_t is the total number of increments so far in the step including the current increment, $\bar{q}^\alpha(t|_n)$ is the value of $\bar{q}^\alpha$ at increment number n with $\bar{q}^\alpha$ as the instantaneous magnitude of the flux for field α at time t, averaged over the entire model:

$$\bar{q}^\alpha(t) \overset{\text{def}}{=} \frac{1}{\sum_{e=1}^{E} \sum_{n_e=1}^{N_e} N_{n_e}^\alpha + N_{ef}^\alpha} \left(\sum_{e=1}^{E} \sum_{n_e=1}^{N_e} \sum_{j=1}^{N_{n_e}^\alpha} |q|_{j,n_e}^\alpha + \sum_{j=1}^{N_{ef}^\alpha} |q|_j^{\alpha,ef} \right). \tag{5.67}$$

E is the number of elements in the model, N_e is the number of nodes in element e, $N_{n_e}^\alpha$ is the number of degrees of freedom of type α at node n_e of element e, $|q|_{j,n_e}^\alpha$ is the magnitude of the total flux component that element e applies at its j'th degree of freedom of type α at its n_e'th node at time t, N_{ef}^α is the number of external fluxes for field α and $|q|_j^{\alpha,ef}$ is the magnitude of the j'th external flux for field α. If this criterion is fulfilled, the last iterated displacement correction and the estimated displacement correction with one more iteration are controlled against the total displacement increment:

$$c_{max}^\alpha \leq C_n^\alpha \Delta u_{max}^\alpha \quad \text{and} \quad c_{est}^\alpha \leq C_n^\alpha \Delta u_{max}^\alpha, \tag{5.68}$$

with

$$c_{est}^{\alpha} = \frac{\left(r_{max}^{\alpha}\right)^{i}}{min\left(\left(r_{max}^{\alpha}\right)^{i-1}, \left(r_{max}^{\alpha}\right)^{i-2}\right)} c_{max}^{\alpha}. \tag{5.69}$$

If the correction is larger than 1% of the increment, one more iteration is performed. In addition, the analyst may define maximum values of the load proportionality factor and maximum values for one or all degrees of freedom to terminate the iterations. The default values in Abaqus/Standard should in general not be altered, however, they may be reset with the *CONTROLS option.

5.9 IFEM

The default solution algorithms used by IFEM are the full Newton-Raphson algorithm and the Return Mapping algorithm described in the former section. The default numerical integration scheme is Gauss' quadrature rule and the variational description is the Updated Lagrangian formulation. The convergence criterion is energy norm based.

Chapter 6

Gap K-Joint

To compare the results obtained by FEA and IGA, a gap K-joint has been analyzed. The model was introduced in the technical report *Static Strength of Tubular Joints, Phase 2. Analyses and tests of Gap and Overlap K-Joints* [26] produced by Veritec, Det Norske Veritas Industri Norge AS. Initially an attempt was made to analyse the model from the Veritec report, but limitations as to modelling boundary conditions in IFEM resulted in the model described in the following sections.

6.1 Geometry

The gap K-joint is modelled with dimensions shown in Figure 6.1. The pipe is asymmetric about the intersection of the centrelines of the two braces. This point will from now on be referred to as the intersection point. The intersection point is located 28 mm below the centreline of the chord. Both braces have outer diameter d_{out} = 365.8 mm and thickness t = 9.3 mm. The length of the right brace is 1800 mm and the length of the left brace is 1550 mm. Both lengths have been measured from the intersection point,

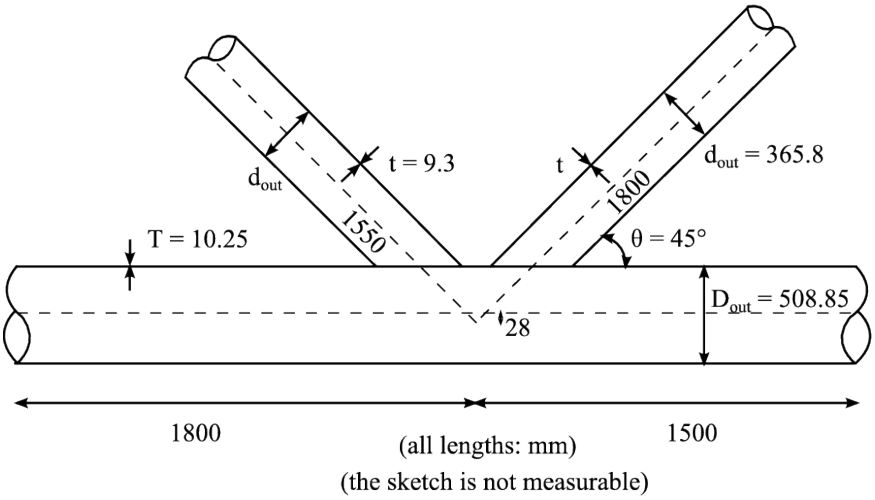


Figure 6.1: Geometry of gap K-joint

along the centreline, up to the brace ends. The incline of the braces relative to the chord is $\theta = 45°$. The outer diameter of the chord is $D_{out} = 508.85$ mm and the thickness is $t = 10.25$ mm. The length of the chord from the intersection point is 1500 mm to the right and 1800 mm to the left. The out-of-plane geometry of the K-joint is symmetric, hence only one half of the model needs to be modelled, assumed that boundary conditions representing symmetry conditions along the out-of-plane axis are applied.

For the analyses with shell elements, the shell geometry has been taken as the mid-surface geometry of the solid model in Figure 6.1. The diameter of the braces is then $d = 356.5$ mm and the diameter of the chord $D = 498.6$ mm. It is worth noticing that the K-joint is just at the limit of being able to use thin shell theory as T/D is approximately 1/24 and t/d approximately 1/19.

The analysis model is depicted in Figure 6.2[1]. The directions of the axes given in this figure will act as the definition of the axes in subsequent discussion.

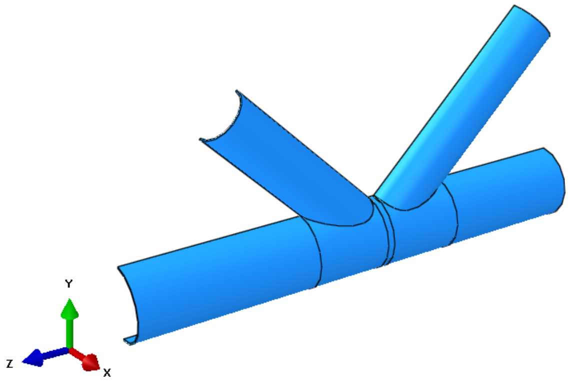


Figure 6.2: Geometry of analysis model

6.2 Assembly

The model consists of three parts; the two braces and the chord. In practice, the parts are connected with welds, which in itself is a research field. This has been neglected in this thesis, but the mesh may be more refined in the areas close to the intersection between the chord and the braces to get a better idea of the local deformations and stresses.

6.2.1 Abaqus/CAE

The constraint between the braces and the chord to simulate the connection is applied as a tie constraint. For nodes constrained by tie, all the active degrees of freedom at the slave component are constrained to be equal to the degrees of freedom at the master component. When coupling solids, the most efficient way of coupling the components for this type of structure is with a surface-to-surface representation. For the gap K-joint, the cut-out of the chord in the thickness direction will act as the master surfaces, and the

[1] The lines dividing the chord is only used for meshing purposes, and do not influence the geometry.

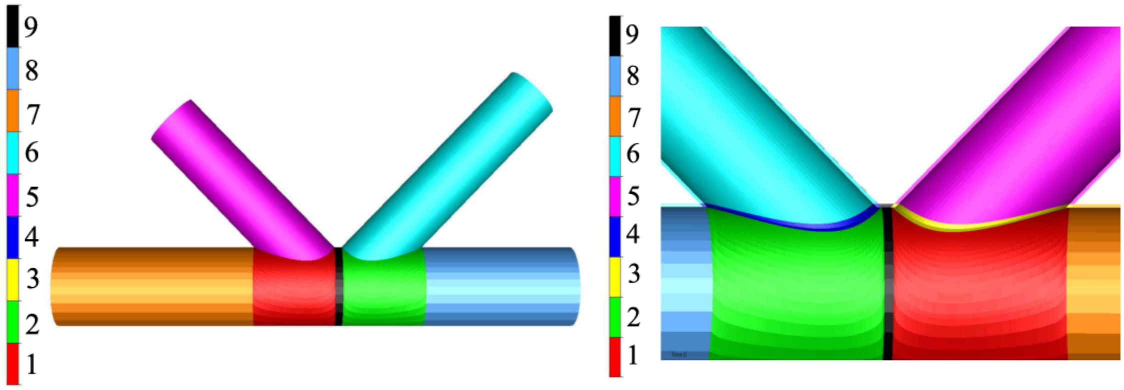
area of the surfaces of the braces that is in contact with the chord will act as the slave surfaces. A partition has been made to easily define the area of the braces that is in contact with the chord. Abaqus/CAE automatically constrains only the nodes which are in connection with the master surface initially, and the nodes within a specified position tolerance, but by creating the exact surface, a more accurate result may be obtained. The position tolerance is defined as 0.1 mm since it is only the nodes within the partition that are to be included in the constraint. This results in one complete row of elements from the bottom of the braces to be constrained to the chord.

For the model with shell elements, the bottom edge of the braces are defined as the slave surfaces, and the cut-out of the chord as the master.

In Abaqus/CAE, the meshes on the two surfaces tied together do not need to be of equal refinement. The degrees of freedom in each node at the slave surface are determined to be equal to the degrees of freedom at the node that is located closest at the master surface [6].

6.2.2 IFEM

To model the gap K-joint for analysis in IFEM, the geometry has been modelled as a conforming multipatch NURBS model. The topology of the model, i.e. the neighbouring information, is handled using in-house software. The intersection lines between the braces and the chord are the first to be defined. Then surfaces are defined using the intersection lines and the lines defining the pipe ends. The volume of the model is created by filling in the space between the defined surfaces. For the gap K-joint, there has been created 9 patches as shown in Figure 6.3. The patches are connected with the a standard master slave technique. Patch 9 acts as slave for both patch 1 and 2. Patch 1 is master also for patch 3 and 7 and patch 3 is master for patch 5. The same hierarchy applies to the right side; patch 2 is master for patches 4 and 8, and patch 4 is master for patch 6. To



(a) Front of gap K-joint

(b) Back of gap K-joint

Figure 6.3: Patches creating the gap K-joint

be able to constrain degrees of freedom of two adjoining surfaces in IFEM, the control points must be coinciding, hence the analyses conducted in this thesis always have the same mesh refinement on intersecting surfaces.

6.3 Material

The material assigned to both the chord and the braces is a multilinear elastic-plastic material with isotropic hardening defined by empiric testing of material properties [26]. The Young's modulus is 210 000 MPa, the yield strength is 360 MPa and the Poisson's ratio is 0.3. The numerical values of the stress-strain relation can be found in Table 6.1. The density of the material is $7.85 \cdot 10^{-6}$ kg/mm^3.

True strain	True equivalent plastic strain	Stress (MPa)
-	0	360
0.0271	0.0253	375
0.0444	0.0423	445
0.0725	0.0701	500
0.1066	0.1040	550
0.1720	0.1691	600
1.0000	0.9967	700

Table 6.1: Numerical values of the multilinear elastic-plastic material

6.4 Mesh

Three refinements of the FE mesh in Abaqus/CAE have been focused upon. One is a very coarse mesh, the second a standard mesh and the third a fine mesh. The second mesh is called standard because this number of unknowns is closest to the number of unknowns used for the models in IFEM. When creating the standard mesh, both the number of unknowns and the shape of the mesh was attempted to replicate the mesh used in IFEM. Since the IFEM mesh relies on the initial discretization of the geometry[2], the meshes are not identical. For the chord the construction of the IFEM mesh began with two elements in the circumferential direction, and one element lengthwise in each patch. The mesh was then refined with knot insertion in all directions. The thickness direction was represented with one element. The braces were initially constructed by 12

[2] The geometry has been defined by Kjetil Andre Johannessen and has not been available for comparison.

elements in the circumferential direction and one lengthwise, and refined as the chord. The meshes from Abaqus/CAE are shown in Appendix A.

The meshes are generated with focus on the areas close to the intersection between the braces and the chord. These areas are subject to large stress/strain-gradients. It is therefore more important to use a finer mesh in these areas, than in the areas further away where the gradients are smaller.

6.5 Elements

The pipe joint has been modelled with both solid and shell elements, however not in combination. The analyses in IFEM has only been conducted with solid elements.

The solid element used in the FEA of the gap K-joint with Abaqus/Standard is the 20-node quadratic brick element with reduced integration and hybrid linear pressure *C3D20RH* [6]. Each node possesses three translational degrees of freedom in direction of the global x-, y- and z-axis.

To improve the element performance, a hydrostatic pressure field is interpolated independently, resulting in a mixed element. However, the additional pressure degrees of freedom are condensed out on the element level so that the solid element has only displacement degrees of freedom [27]. This ensures proper deformations in case of material behaviour close to incompressibility. The deviatoric part of the stress field is integrated with full integration, whereas the hydrostatic part is integrated using reduced integration.

The shell element used in the FEA with Abaqus/Standard is the *S8R5*, an 8-node doubly curved thin shell, with reduced integration, using five degrees of freedom per node. The degrees of freedom are the translations in three perpendicular directions, and the rotations about the two in-plane axes. Simpson's integration rule has been used in the thickness direction to be able to compare the results directly to the values in the nodes of the solid elements. The Kirchhoff thin shell formulation, implying zero shear strain, is imposed numerically by the penalty method, see Section 5.4.3, as a transverse shear stiffness [6].

6.6 Load and Boundary Conditions

The load applied to the gap K-joint is a prescribed displacement of 16 mm at the upper end of the left brace. The displacement is applied as a *static, general* step, with prescribed load increments, on the local axial direction, resulting in tension in the left brace. Both in Abaqus/Standard and IFEM the load applied with a linear ramp over the time steps.

The left end of the chord is constrained against displacement in axial direction of the chord, whereas the right end of the chord is free to move. The upper end of the right brace is constrained against displacement in local axial direction. The symmetry plane is constrained not to move in the out-of-plane direction, neither to rotate about the two in-plane axes. The last constraint will only affect the shell representation, since the rotational degrees of freedom are not active in the model with solid elements.

For the natural frequency analyses, the upper end of the left brace is also constrained against displacement in local axial direction.

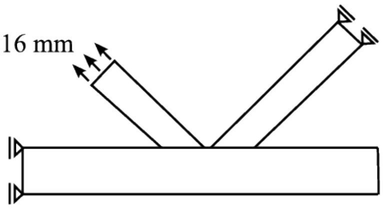


Figure 6.4: Load and boundary conditions of the gap K-joint

6.7 Warnings in Abaqus/Standard

All the warnings discussed in this chapter are referring to the standard mesh. However, the warnings have been the same for all refinements, disregarding the number and location of poor and distorted elements.

6.7.1 Poor Elements

To be able to have a certain amount of control of the structure of the FE mesh, the model must be partitioned. The partition of the model in Abaqus/CAE is performed on order to as easy as possible be able to create a mesh that is similar to the mesh of the isogeometric model. The result is partitions indicated by the internal lines in Figure 6.2.

A problem area that arises for the solid model is the area of the chord which surrounds the intersection with the braces, especially the small part between the braces. As shown in Figure 6.5a, the partition of the chord, created by sweeping the edge of the mid section between the braces along the circular edge of the chord, will result in an area, marked by a circle, where the thickness eventually goes towards zero. This generates warnings in Abaqus/Standard due to the fact that the elements used are brick elements, and thus requires six faces to be defined. For the standard mesh, the warning applies to a total of 10 elements, shown in Figure 6.5b.

(a) Partition of area between the braces

(b) Poor elements

Figure 6.5: Area of poor elements

6.7.2 Warped Elements

The default setting in Abaqus/Standard is set to control the curvature of the shell elements used in the model. The angle which is limited is the angle between the average element normal and a nodal normal, which is limited to 10°. The elements being warped in the shell model with standard mesh are shown in Figure 6.6.

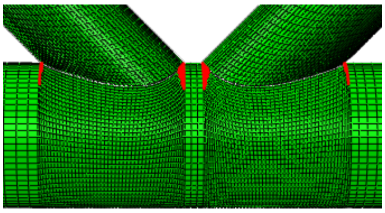
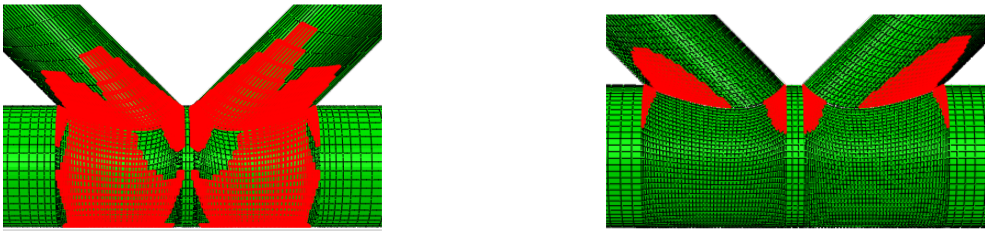


Figure 6.6: Warped elements

6.7.3 Distorted Elements

In addition to the poor elements, which are present already from the meshing face, some elements are characterized as distorted throughout the analysis. The error emerges as the analyses commence because the angles between the physical representation and the parametric representation reaches a specified limit. Consequences may be negative volume of an element at a node, which obviously is incorrect. If distorted elements are detected, Abaqus/Standard automatically reduces the next time step with a default cutback of 25%, to increase the possibility of convergence. The angles are upward limited at 135°, and downward at 45°. The distorted elements for the solid model are shown in Figure 6.7a, and for the shell model in Figure 6.7b, both for the standard mesh. The elements may also be distorted due to a large ratio between the longest and the shortest edge. The aspect ratio is limited to 10 [28]. The aspect ratio leads to warnings for the very coarse meshes because of the size of the outer elements of the braces and the chord.



(a) Solid

(b) Shell

Figure 6.7: Distorted elements

6.7.4 Boundary Conditions on Inactive Degrees of Freedom

A set of nodes, identified in Figure 6.8a, has been alarmed due to boundary conditions applied to inactive degrees of freedom. The figure depicts the standard mesh, but the nodes are the same for all mesh refinements. The degrees of freedom identified are located both at a slave surface defined by the constraint between the braces and the chord, and at the face of the braces that has symmetry boundary conditions applied. The result is that the boundary conditions will not be applied to these nodes. However, this is not a problem, because the nodes being master nodes to the alarmed nodes are located at a face of the chord which has symmetry boundary conditions applied to it, and will therefore ensure the right constraint of the braces.

(a) Location of nodes

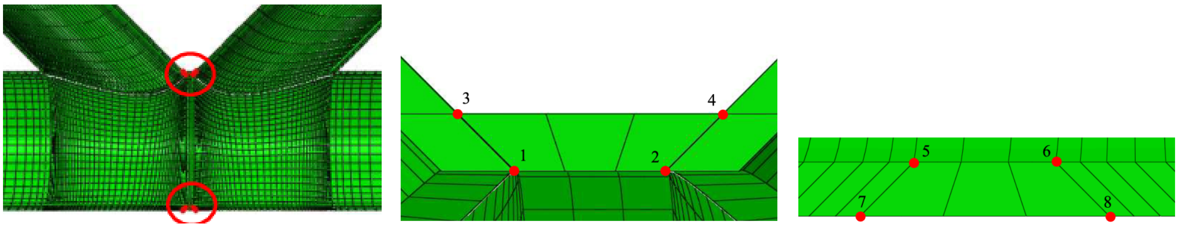
(b) Close-up of location of nodes at the braces relative to the chord

Figure 6.8: Nodes with boundary conditions applied to inactive degrees of freedom

The same warning appears in the shell model analyses on the degrees of freedom on the intersection of the braces and the chord which are located at the symmetry plane. The above description and conclusion also apply for shells.

6.8 Areas of Interest

To compare the results from Abaqus/Standard and IFEM, both local and global results will be considered. For the local results, maximum values of a selection of variables will be examined. The areas in focus will be the corners defined by the partitions/patches between the braces as shown in Figure 6.9. These nodes are easy to pick with different mesh refinements both in Abaqus/CAE and IFEM, and the areas are interesting because they are located in the areas most affected by the intersection of the braces and the chord. The nodes will be referred to as the *result nodes*, and the numbering of the nodes is shown



(a) Location of areas of interest

(b) Close-up of upper nodes of interest

(c) Close-up of lower nodes of interest

Figure 6.9: Areas of interest

in Figures 6.9b and 6.9c. A drawback of these nodes are that they are located on patch boundaries resulting in equal continuity in for the FEA and the IGA. The results and differences discussed later is thus an effect of the different continuity globally in the model, and would have been even clearer if nodes also displaying this continuity had been chosen.

The location of the nodes in the shell model is on the midsurface of the K-joint, and the same numbering as for the solid model is not automatically possible. However, the variables are chosen to be calculated at three Simpson points, which means that the section points for the nodes correspond to the nodes for the solid model. The numbering of the section points starts with number 1 at the bottom of the shell, which for the K-joint is the inside of the chord, and results in number 3 on the top of the shell, i.e. the outer surface of the chord. Thus the numbering of the section points result in the same location as for the nodes shown in Figures 6.9b and 6.9c.

Chapter 7

Results and Discussion

7.1 Analysis Data

The analyses run in Abaqus/Standard are:

1. Natural frequency analysis with very coarse, standard and fine mesh, solid elements.
2. Natural frequency analysis with very coarse and standard mesh, shell elements.
3. Five nonlinear analyses with prescribed displacements and different mesh refinements, solid elements.
4. Nonlinear analyses with prescribed displacements and very coarse and standard mesh, shell elements.

The analyses run in IFEM are[1]

5. Natural frequency analyses with $q = 2$, $q = 3$ and $q = 4$.
6. Nonlinear analyses with prescribed displacements and $q = 2$, $q = 3$ and $q = 4$.

When an Abaqus/Standard analysis is referred to without further specifications, it is the standard mesh, nonlinear analysis with solid elements that applies. As for IFEM, the standard analysis is the nonlinear analysis with $q = 2$.

	Very coarse ABQ	Standard ABQ	Fine ABQ	Very coarse ABQ shell	Standard ABQ shell	Standard IFEM
Elements	60	4056	15420	62	4335	10692
Nodes	684	38118	141429	309	17875	36387
Number of unknowns	1932	105402	393447	1854	107250	106356

Table 7.1: Mesh data

[1] To avoid confusion between the degree of the basis functions and equivalent plastic strain, will from now on the general degree of the basis functions be denoted q.

The coarsest mesh of both the solid and shell analysis with Abaqus/Standard has far too large elements and results from these analyses will only be included to show the effects of such a coarse refinement. For general analysis purposes a structure with such coarse mesh should not be used, and if so the results should be interpreted with extreme caution.

Analysis	ABQ very coarse	ABQ	ABQ fine	ABQ very coarse shell	ABQ standard shell	IFEM $q = 2$	IFEM $q = 3$	IFEM $q = 4$
Tot. CPU time (sec)	2.3	390.40	3559.5	1.7	2047.7	4922.04	18190.35	68401.28
Tot. no. of incr.	10	12	20	10	143	10	10	10
Tot. no. of ite.	24	48	93	30	586	30	29	30
Max. no. of ite. per incr.	3	5	7	4	7	4	3	4
Min. no. of ite. per incr.	2	2	2	2	2	2	2	2
Max. time step size	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Min. time step size	0.1	0.025	0.025	0.1	0.00154	0.1	0.1	0.1
No. of cutbacks	0	1	3	0	39	0	0	0
Incr. for cut-back(s)	-	7	3,5,11	-	Evenly spaced	-	-	-

Table 7.2: Analysis data

The nonlinear analyses in IFEM have initial time step of 0.1, maximum time step of 0.1 and minimum time step of 0.025. To aim at the most similar analyses these properties were also adopted for Abaqus/Standard. However, for the standard analysis with shell elements, the required time steps were less than 0.025, and the limit was reduced to 10^{-5}. The time steps relate to the prescribed displacement in the same way the load proportionality factor relates to prescribed load, which means that time = 1 equals prescribed displacement = 16 mm, and the displacement increments are calculated as the product of the time increment and the total value of the prescribed displacements. The computational details of the analyses are shown in Table 7.2.

One property of Table 7.2 is that the computational time for the IGAs is much higher than for the traditional FEAs, and the difference increases with increasing degree of the basis functions. The reason for this was explained in Section 3.3.2 and is related to the continuity between the patches and the elements as stated in Section 3.1.2. For both methods of analysis the number of interelement borders is much larger than the borders between FE meshes/patches and the dominating continuity for the traditional FE model is therefore C^0 and for the isogeometric model C^{q-1}, which results in higher solution cost for the isogemetric analyses. Because of the extent of the extra time required by the IGA (the relation between the time for the standard mesh in Abaqus/Standard and for $q = 2$ in IFEM is approximately 12.5), this may be said to be a significant drawback for the isogeometric concept.

7.2 Convergence

7.2.1 Extrapolated Exact Solution

To find the extrapolated exact solution, the Richardson's extrapolation formula has been used [5]. The formula generates an extrapolated exact solution, based on values of the field variables obtained with different mesh refinements. With h as element size, Φ as field variable and a and b as the coarse and refined mesh respectively, the Richardson's extrapolation formula yields:

$$\Phi_{exact} = \frac{\Phi_a h_b - \Phi_b h_a}{h_b - h_a}. \tag{7.1}$$

7.2.2 Solid

External Energy

To examine the convergence of the results obtained with Abaqus/Standard, the value of the external energy has been extrapolated to be able to compare an approximate exact solution to the results from the performed analyses. The characteristic element lengths are calculated as half of the longest line possible to fit inside an element [6] and are extracted from the .msg-file created for each job in Abaqus/Standard. However, the characteristic element lengths are not available in IFEM, and the results are therefore plotted against the total number of unknowns in the model. The extrapolated value is calculated using the external energy and characteristic element length from the standard and fine mesh.

The error in the external energy relative to the Richardson's extrapolated exact value, shown in Figure 7.1, is monotonically decreasing with increasing number of unknowns, which is a desirable result. However, the error is large for the two coarsest solid meshes

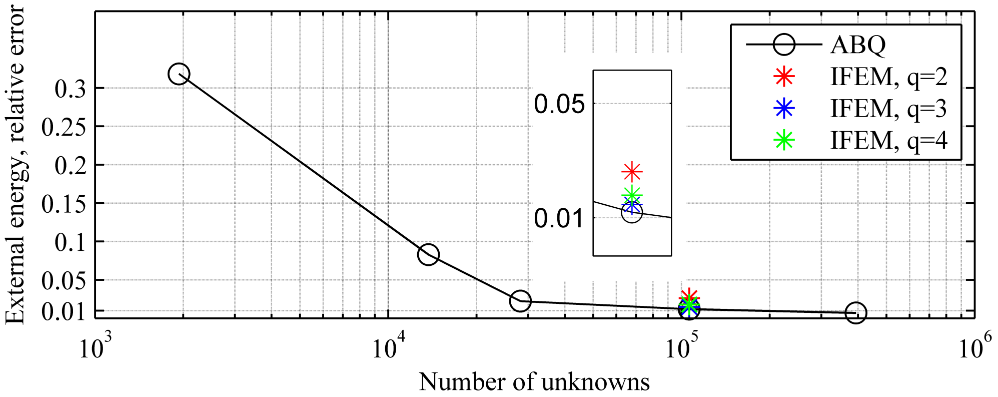


Figure 7.1: Error of external energy relative to the Richardson extrapolated exact value

and indicates that results from these analyses should be interpreted with caution. Too high external energy represents a structure which is too stiff, and the overall results are therefore not conservative. The maximum values of the displacements may nevertheless be too high, because the external energy takes into account the deformation of the entire structure. This can be exemplified with the maximum value of v, in Figure 7.6, which is highest for the very coarse solid model f. As stated in Section 7.1 the coarsest FE mesh is only included as demonstration of effects of very large elements, and the significant error in external energy substantiate that the results should not be used.

When the structure is refined beyond 28212 unknowns, the convergence rate decreases drastically. This is a normal behaviour and gives a good approximation on the number of unknowns that is required to get an approximately good result, without resulting in a too high computational time. Consequently the standard FE mesh with solid elements can represent an approximate exact solution in comparison to variables from the other analyses with standard mesh.

An uncertain factor influencing the convergence is how the exact value is calculated. In Figure 7.1, the exact value is calculated using the results from the analyses with standard and fine mesh. The development of the external energy is not linear, and by using the chosen refinements in Equation (7.1), the incline of the extension of the graph becomes closest to the last increment. Tests show that if the external energy had been calculated with other values, the error would have been even larger, because they are all produced with an extrapolated exact value lower than the currently used refinements. It was concluded that the chosen method gave the most likely result because of a tendency of decreasing rate of change in the external energy. The definition of the characteristic element length should also be discussed. Because the models contain local refinements, the characteristic element lengths do not give a very good picture of the refinement of the model. However, the lengths were only used as a measure of refinement and Equation (7.1) is therefore concluded to give a relatively good result.

Overall, it would have been desirable to approach an error further below 1%, but with the exact result also as an approximation, the error estimates may be said to be sufficient.

Other Variables

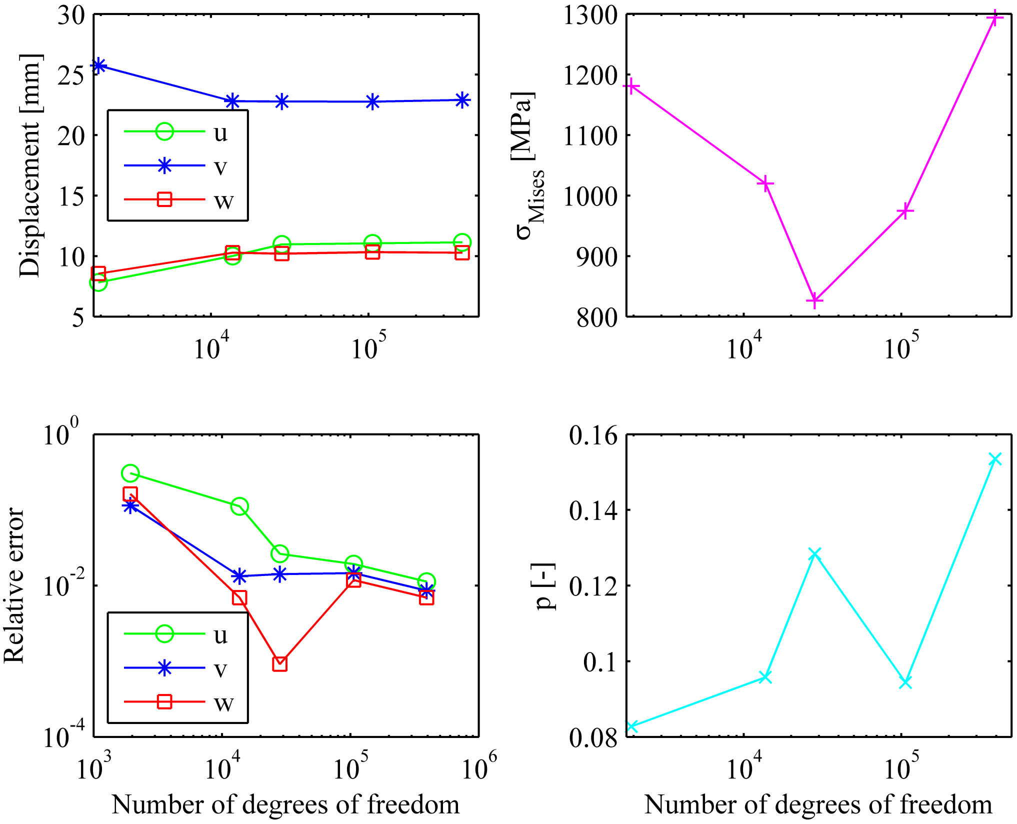


Figure 7.2: Development of field variables with mesh refinements for the solid analysis in Abaqus/Standard

To obtain a general overview of the convergence of other parameters, the maximum displacement in $x-$, $y-$ and $z-$direction have been extrapolated with Richardson's formula. So have the von Mises stress and the equivalent plastic strain. The convergence of the parameters can be seen in Figure 7.2.

The extrapolation has been made despite some of the limitations for the formula. The most important in this context is the restriction for which the values extrapolated must be located in the same point for each refinements. The location of maximum displacements, stresses and strains changes for each refinement in the gap K-joint. It is shown [5] that analyses not fulfilling the limitations can give good results anyway, and this may be the case for the displacements.

The figure shows that there is good convergence of the displacements, but the von Mises stress and equivalent plastic strain do not converge, and an error analysis has not been performed for those variables. The maximum values of the von Mises stress and the equivalent plastic strain are located in the area surrounding the intersection between the braces and the chord and can indicate that the result has an extreme dependence on the

FE mesh refinement. Additionally, both the von Mises stress and equivalent plastic strain are calculated from the first derivatives of the field variables, which means that continuity is not present in a solid analysis in Abaqus/Standard, and may also affect the results.

The error of the displacement converges to approximately 1% which is the same as for the external energy. The kink in the error plot of w is probably also explained by the location of the maximum value of w which is located in the intersection area, and the dependence on the FE mesh refinement.

7.2.3 Shell

External Energy

A convergence study of the variables from the shell analyses has also been attempted. However, the external energy did not show any signs of convergence, as can be seen in Figure 7.3.

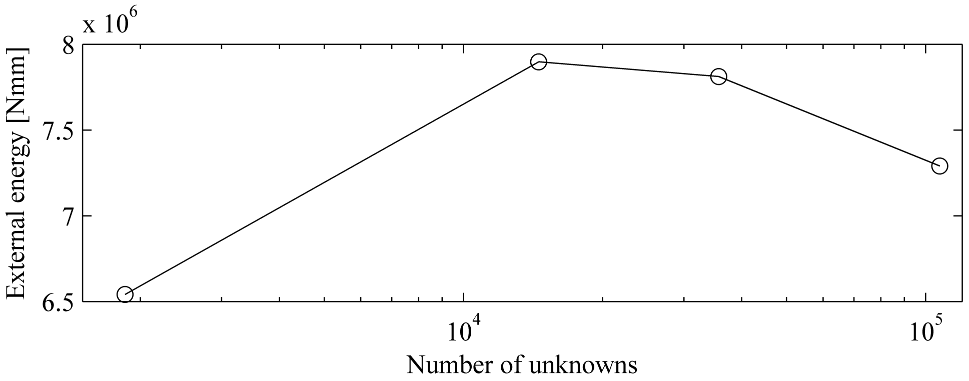


Figure 7.3: External energy of the analyses with shell elements

The most likely reason for this is the inability of the shell elements to represent the three dimensional stress state in the intersection area of the K-joint, as described in Section 7.1. Because of the diverging values of external energy the other field variables were not examined for convergence for the shell model.

7.3 Natural Frequencies

A good estimate of the similarities between different models can be the natural frequencies. If the natural frequencies of models with various discretizations are approximately equal, the models may be assumed to behave relatively similar. The error between the natural frequencies for the models in Abaqus/Standard with very coarsened, standard and fine mesh, the Abaqus/Standard shell model with very coarse and standard mesh and the models in IFEM with $q = 2$, $q = 3$ and $q = 4$ are shown in Figure 7.4. The error is relative to the natural frequency of the analysis in IFEM with $q = 2$.

The natural frequency for the very coarse mesh with solid elements is far too high for modes 2 and 6, and far too low for mode 10. It is earlier stated that the coarsest

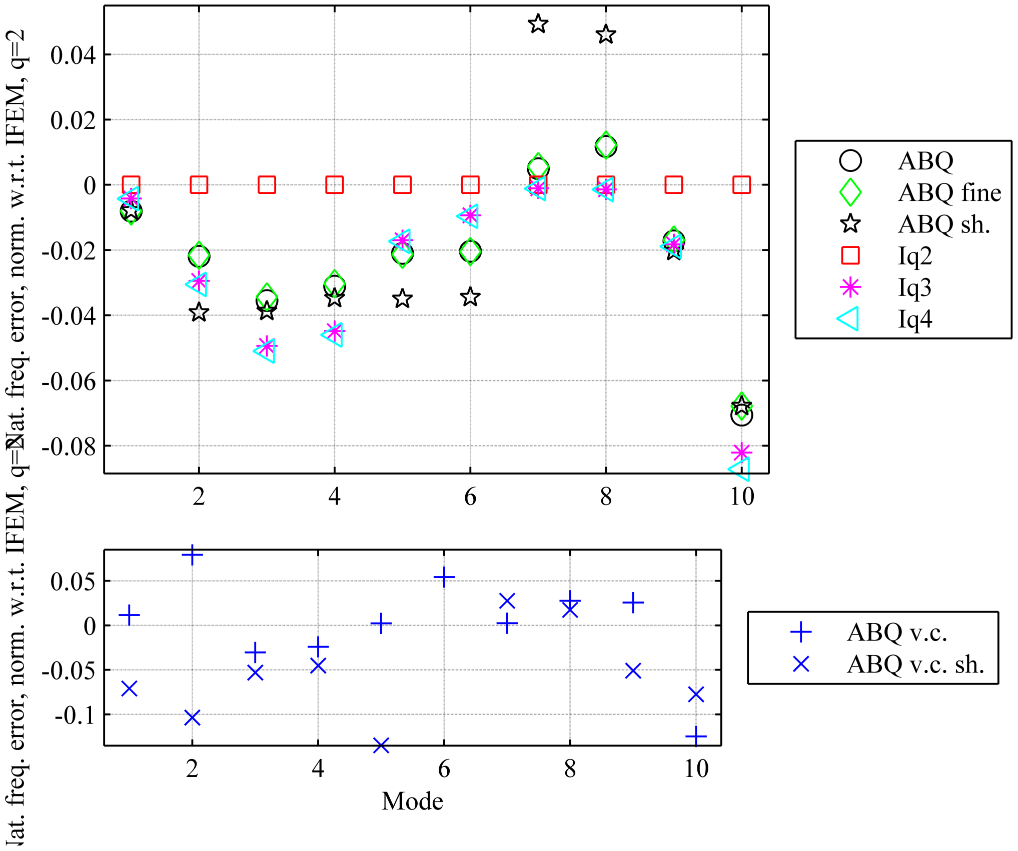
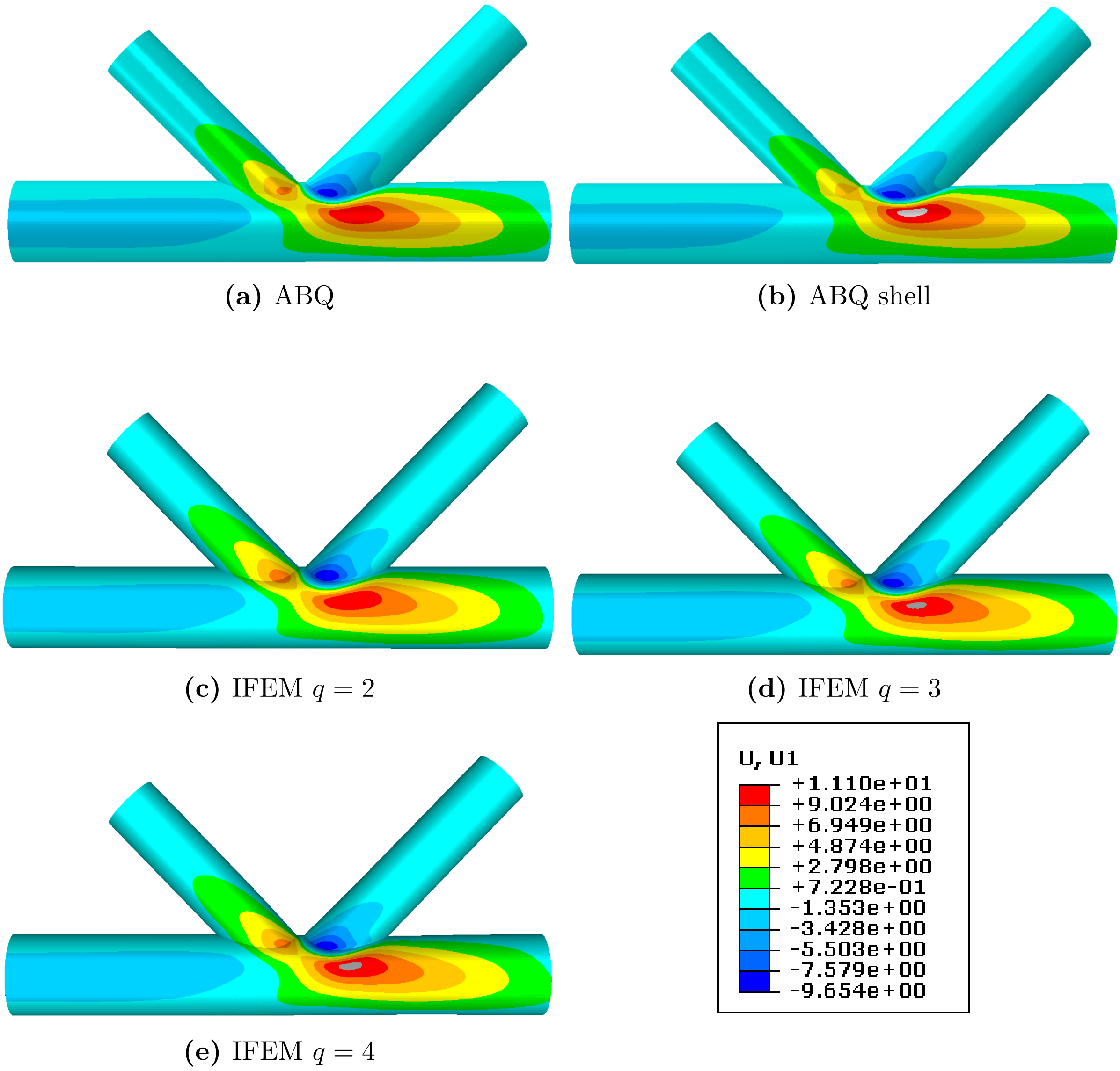


Figure 7.4: Error of the natural frequencies relative to the values from model in IFEM with $q = 2$.

meshes are only included for brief comparison, and the errors in the natural frequencies are all contributing to the conclusion that such coarse meshes can not be used. Also the very coarse mesh with shell elements exhibit severe errors in the natural frequencies. It becomes clear by looking at the plots of deformation, stresses and strains in Figures 7.5 through 7.7 and Figures 7.11 through 7.14, that very large elements will experience trouble obtaining a good result, as the magnitude of the stresses and strains in certain areas changes rapidly. With only a limited number of integration points per element, the values become significantly inaccurate. Also the standard mesh, shell model in Abaqus/Standard exhibits errors generally larger than the other models.

All the other solid models exhibit generally softer behaviour than the standard mesh model in IFEM with $q = 2$. As a rule of thumb, the softer the model is the more accurate representation it will provide, which may indicate that a standard, traditional FE method or higher order isogeometric models are preferable over an isogeometric model with quadratic basis.

Overall the differences between the natural frequencies are small, which indicate that the models behave relatively similar. This may also be substantiated by looking at the contour plots of the displacements. In order to make a comparison of the results easier, the scale of the contour plots has been modified to correspond to the contour plot for the analysis with $q = 2$ in GLview Inova. In some cases this results in values for the other analyses that are outside the contour range. Larger values are shown in grey and smaller values are shown in black[2]. Some of the values may be located at internal nodes and hence not visible at the plots. Figure 7.5 depicts both the relatively similar behaviour represented by the deformation patterns, while the magnitude of the displacement shows that there are differences in stiffness of the models. The out of bound values in IFEM with $q = 3$ and $q = 4$ are shown with the maximum/minimum colors of the contour plot.



(a) ABQ

(b) ABQ shell

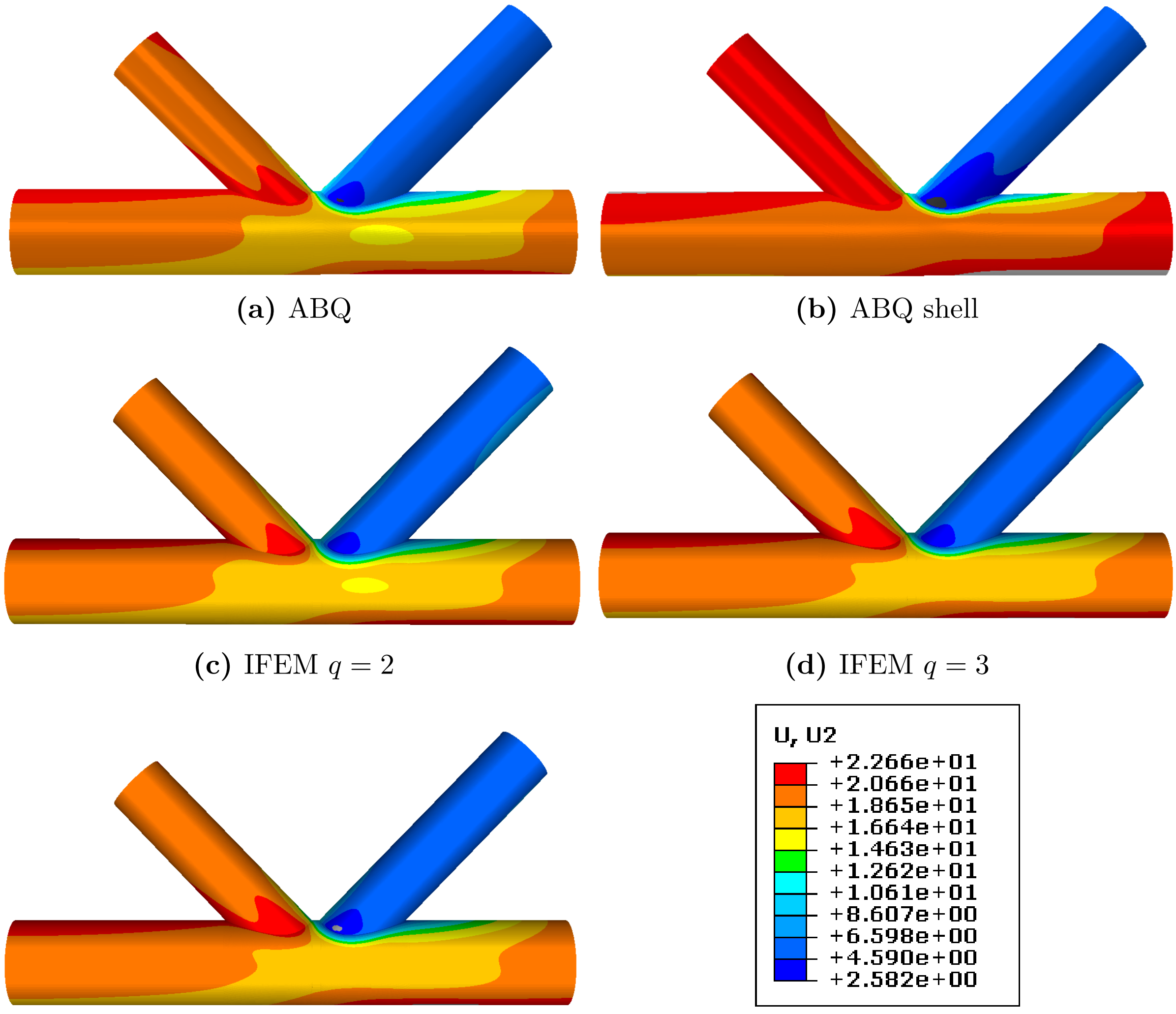
(c) IFEM $q = 2$

(d) IFEM $q = 3$

(e) IFEM $q = 4$

Figure 7.5: Displacement in global x-direction, u

[2] Keep in mind that as the absolute value of a negative result increases, the value will decrease.



(a) ABQ

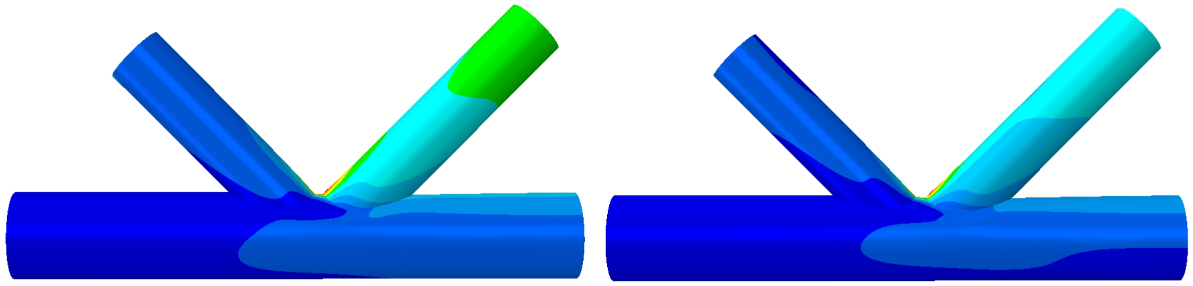
(b) ABQ shell

(c) IFEM $q = 2$

(d) IFEM $q = 3$

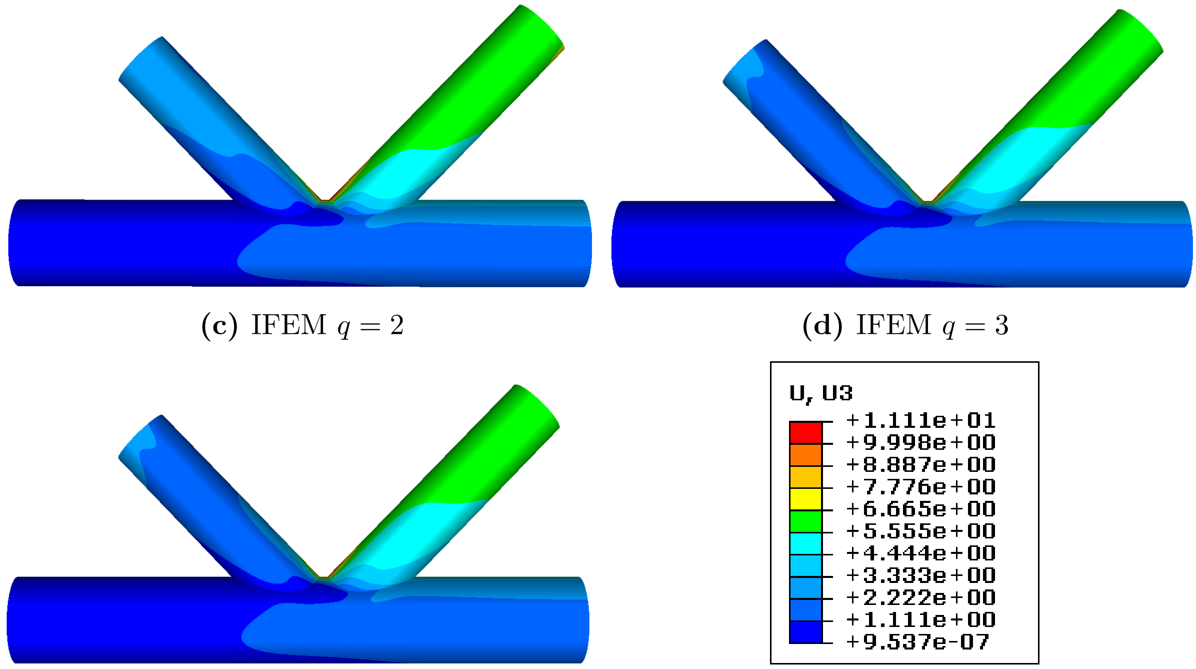
(e) IFEM $q = 4$

Figure 7.6: Displacement in global y-direction, v



(a) ABQ

(b) ABQ shell



(c) IFEM $q = 2$

(d) IFEM $q = 3$

(e) IFEM $q = 4$

Figure 7.7: Displacement in global z-direction, w

7.4 Nodal Results

Figure 7.8 shows the value of a selection of central local results. The results are located in result nodes 1 through 8. The variables chosen are displacements globally in direction of y and z: v and w, the von Mises stress as defined in Equation (4.41): σ_M, the equivalent plastic strain as defined in Equation (4.43): p, the deviatoric stress from Equation (4.23): σ_h , and the maximum, medium and minimum principal stress values calculated as in Equation (4.25): σ_1, σ_2 and σ_3. The values of u are not included because all the nodes are located on the symmetry plane, hence the displacement in $x-$direction is prohibited. The tabulated values are given in Appendix C.

The figure shows that the local results can be split into two groups; the results for the upper group of nodes and the lower group of nodes, as shown in Figure 6.9. The stress state in the upper group has reached plasticity, shown by the non-zero values of p, while the material in the lower group of nodes still is in the elastic range.

The results for the coarsest mesh in Abaqus/Standard differ a great deal from the other results for displacement, as expected. For the other variables, large differences appear in the plastic area.

It is clear that the accuracy of the results depends on the stress state of the material, shown by the scattering of the stress and strain results for the plastic area in contrast to the gathered stress and strain results for the elastic area. One reason for this may be that the difference in equivalent plastic strain propagates to the other parameters. The

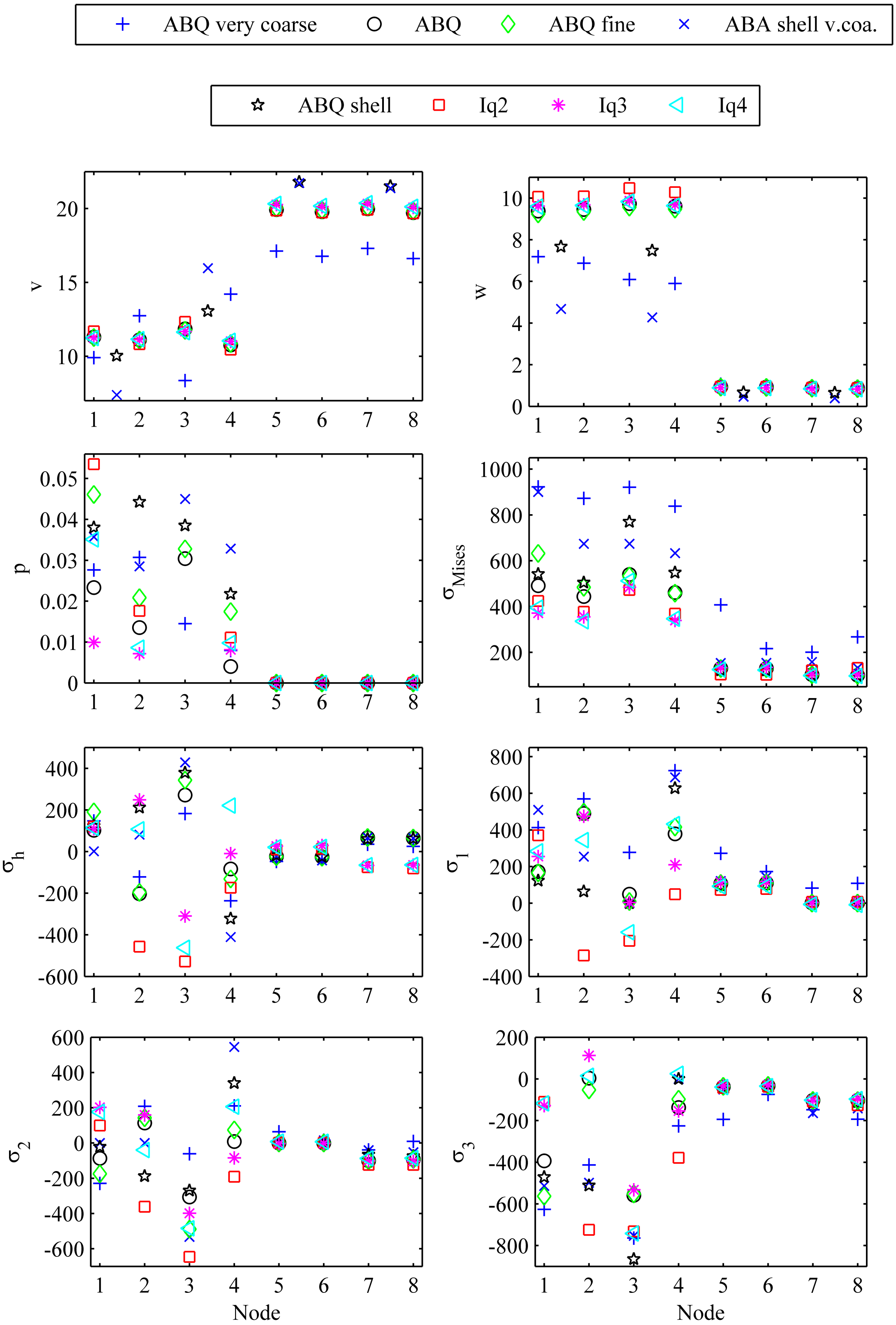


Figure 7.8: Local results in node 1 through 8 according to Figure 6.9 for the analyses in Abaqus/Standard with very coarse, standard and fine solid mesh, the Abaqus/Standard shell analyses with very coarse and standard mesh and the analyses in IFEM with $q = 2$, $q = 3$ and $q = 4$. All displacements in mm, all stresses in MPa. Displacement components in global directions. NB: equivalent strain in nodes 1 and 2 for the very coarse solid mesh is manually averaged.

calculation of the von Mises stress in a plastic area is dependent on the equivalent plastic strain shown by Equation (4.41), and in combination with different calculation methods for derived variables in Abaqus/Standard and IFEM, see Section 7.5, a difference in the equivalent plastic strain will affect the stress values. Abaqus/Standard and IFEM apply the same solution algorithm for calculating the elastic/plastic strain ratio in an increment, and the calculation of the stresses and strains in the integration points should therefore not be a source of difference (disregarded the shape functions).

7.5 Differences Relating to the Basis Functions

As an introduction to the discussion, a summary of the differences and similarities of traditional FEA and IGA may be adequate to provide. The basis for Tables 7.3 and 7.4 was created as a part of the conclusion of the project being a precursor to this thesis.

Isogeometric analysis	**Finite element analysis**
Exact geometry	Interpolative geometry
Not interpolating	Interpolating
Convex hull property	Oscillating
Degrees of freedom in the control points	Degrees of freedom in the nodes
CAD model = analysis model	Conversion from CAD model to analysis model is required
Refinement of surfaces in one direction extends over the whole patch	Refinement only affects the refined area
Support on limited number of knots, but multiple elements	Support on limited number of nodes and only on one element
Refinement does not change the geometry	Refinement changes the geometry
C^{q-s} continuity at knots	No certainty for continuity
Non-negativity	Positive and negative

Table 7.3: Differences between IGA and traditionFEA

Many of the properties are basis for the differences seen in the global and local results.

The most prominent difference between the traditional FEA and the IGA is the shape functions used. The traditional FEA uses lower order Lagrange polynomials for solid elements whereas the IGA applies NURBS as basis functions. The major difference affecting the results between the two is the continuity. The continuity requirements for the methods are given in Sections 2.2.1 and 3.1.2 respectively and results for a solid analysis in C^0 continuity over element borders for traditional FEA and C^{q-1} for IGA with basis functions of degree q. This generates differences in the calculation of the stress and strain

Code architecture for analysis (not discussed)
Isoparametric implementations
Gauss'/Simpons's integration rule
Partition of unity
Affine transformation possible

Table 7.4: Similarities between IGA and traditional FEA

variables for the FEA and the IGA.

Another difference is the ability the NURBS basis functions may have to exactly represent conic sections, which is profitable in the way of giving the possibility to run analyses on models equal to the physical geometry. The interpolation functions used in traditional FEA can only approximate such geometry, and if two equal analyses performed on two different geometries were to be compared it is clear that the results would be more correct with exact geometry. An especially large effect can be apparent for problems where buckling may occur, because of the sensibility of geometric irregularities. The gap K-joint is made up of a total of three pipes, all with circular geometry, and all being on the limit of being thin shells, which is especially vulnerable to buckling. However, a force-displacement plot from the upper end of the left brace in the standard solid analysis in Abaqus/Standard, shown in Figure 7.9, shows that the K-joint does not exhibit buckling behavoiur, and the geometric differences will therefore not be further discussed.

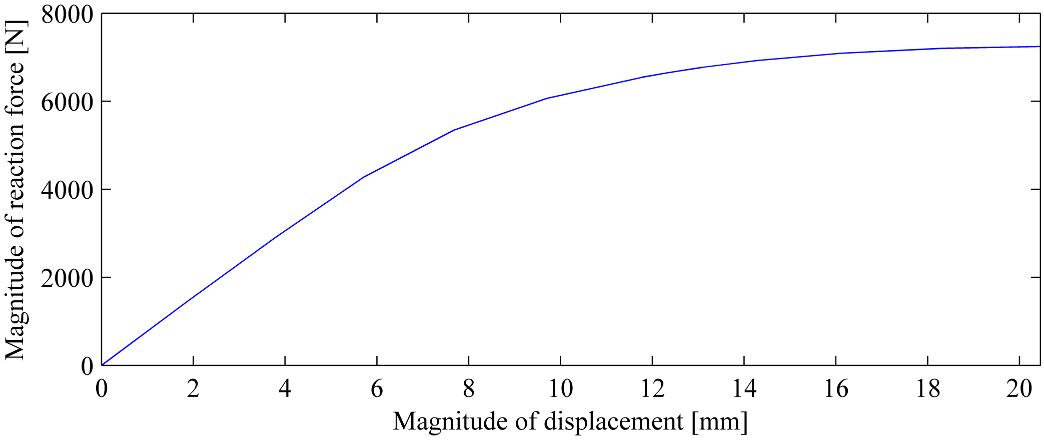


Figure 7.9: Plot of force vs. displacement of the upper end of the left brace.

A third difference, not relating to the theory of traditional FEA and IGA, but to the manners of the software used to perform the analyses is the difference in handling derived values such as von Mises stress and principal stresses. In Abaqus/Standard the stress components calculated in the integration points are extrapolated to the nodes. The extrapolation is done according to the shape functions, i.e. lower order Lagrange polynomials for the solid analysis, as shown in Section 2.2. The von Mises stress, which is derived by the formula given in Equation (4.41), is calculated with the extrapolated results, directly in the nodes. To create a smooth plot, all values are averaged over element and mesh borders. In IFEM the derived variables are treated differently; they are calculated in the integration points in the same way as the component variables and then extrapolated to the nodes according to the NURBS basis functions. A smooth plot of the IG results is created by performing an L2-projection. The post-processed results are plotted in visualization points chosen by the analyst. The more visualization points, the smoother the result plot will be. The results are averaged over the patch borders.

This will in total include three sources of difference for derived variables; the basis functions used to calculate the field variables, the continuity requirement of the field variables and the treatment method of the derived variables. When several sources of difference are summed, the total result may either include an accumulation of the differences, or the differences in each procedure may cancel each other out.

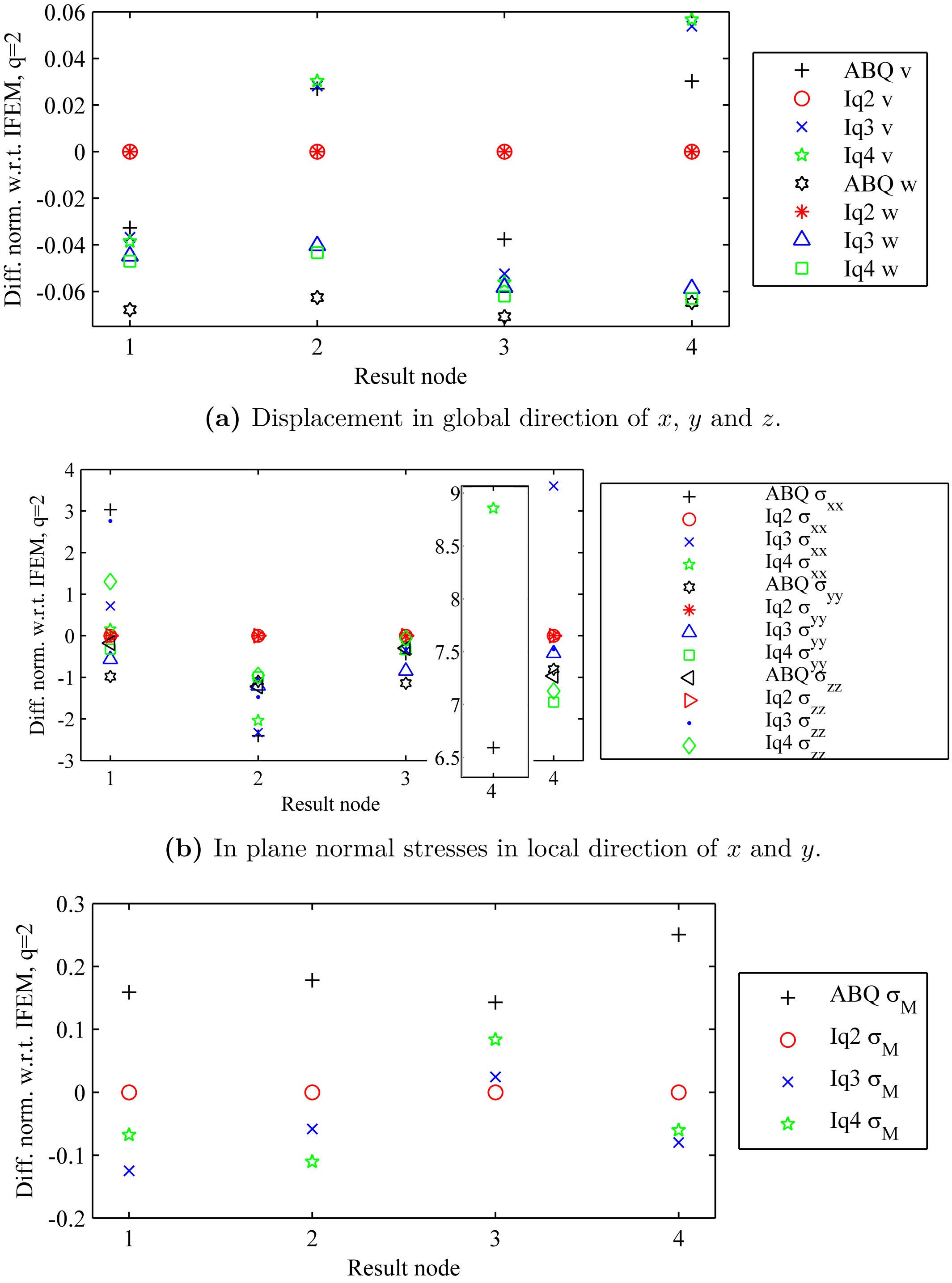
To show the effect of the differences addressed, the values of displacement, normal stresses and von Mises stress have been plotted for the area between the chord and the braces being result nodes 1 through 4 from Figure 6.9b, located on the chord. The differences between the values are shown in Figure 7.10. The differences have been normalized with respect to the results from IFEM with $q = 2$ to be able to both compare the results between Abaqus/Standard and IFEM and between the analyses in IFEM with basis functions of varying degree.

Figures 7.10a and 7.10b show that it is a much larger difference between the values of the stress components than for the values of the displacement. This is as expected since the only difference between the calculation of the displacement lies in the shape functions. For the stress components another source of difference is included, namely the continuity. In IFEM, the continuity across the element borders is C^1, C^2 and C^3 for the analyses with $q = 2$, $q = 3$ and $q = 4$ respectively. This results in stress components being continuous because they are functions of the first derivative of the field variables, and also continuous rates of stress for $q = 3$ and $q = 4$. For the traditional solid FEA, the continuity is only C^0, and the stress components are therefore not continuous.

Figure 7.10a shows that the difference between the displacement values differs between being positive and negative for all the components, implying that there is no basis of stating that one method is more conservative than the other. Also the difference in the von Mises stress values in Figure 7.10c varies between being positive and negative, hence is the stress calculation not definite conservative for neither of the methods.

The difference in handling the derived variables in Abaqus/Standard and IFEM will, in addition to the different shape functions and continuity, affect the von Mises stress results. Figure 7.10c shows that instead of accumulating the differences, as mentioned as one possibility earlier, the differences have almost cancelled each other out, resulting in differences much smaller than for the stress components. However, the differences are still greater than for the displacement components.

The above differences are visualized in Figures 7.5 through 7.7 and Figures 7.11 through 7.14.

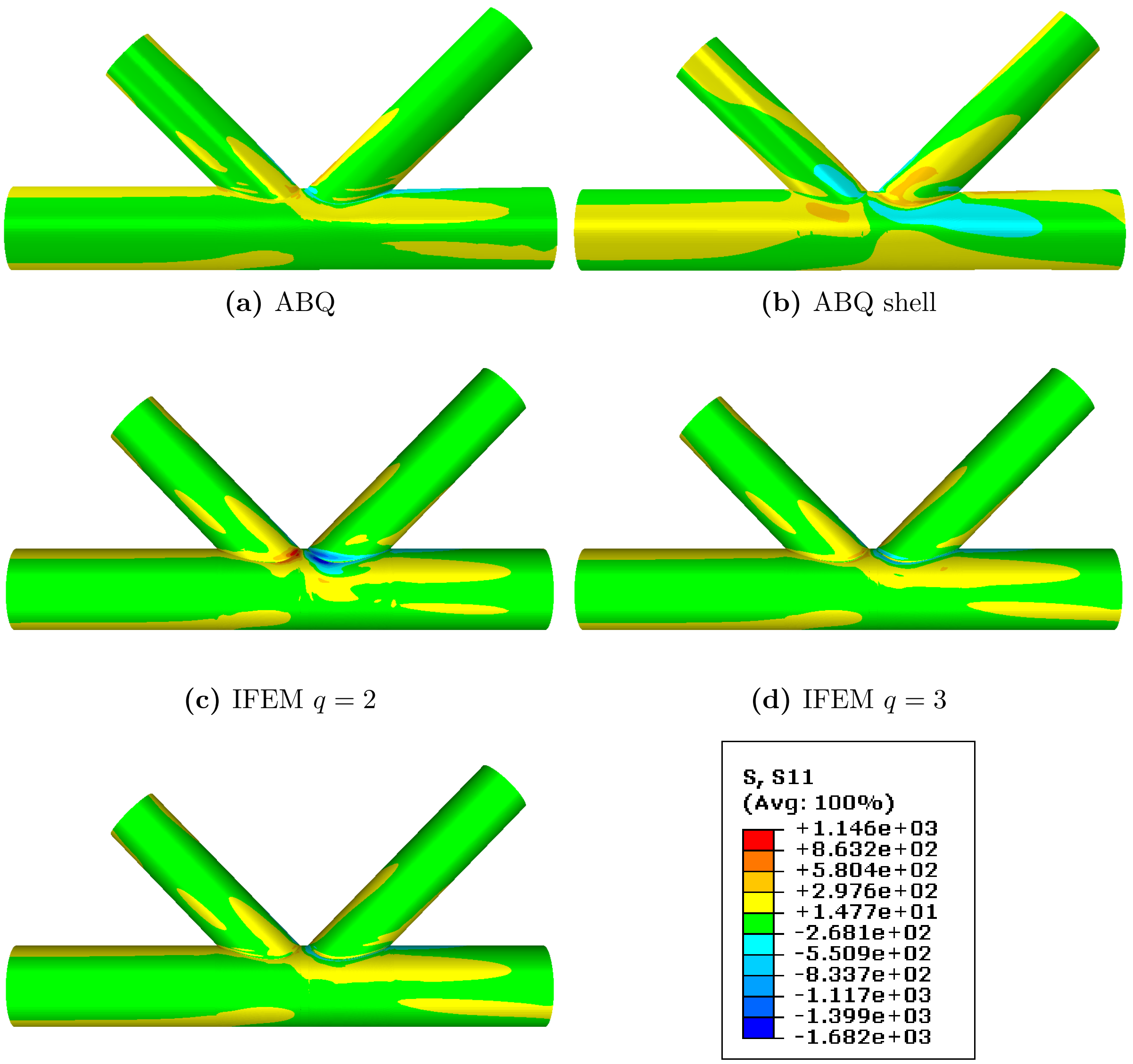


(a) Displacement in global direction of x, y and z.

(b) In plane normal stresses in local direction of x and y.

(c) Von Mises stress.

Figure 7.10: Differences in local results between traditional solid FEA and IGA.



(a) ABQ

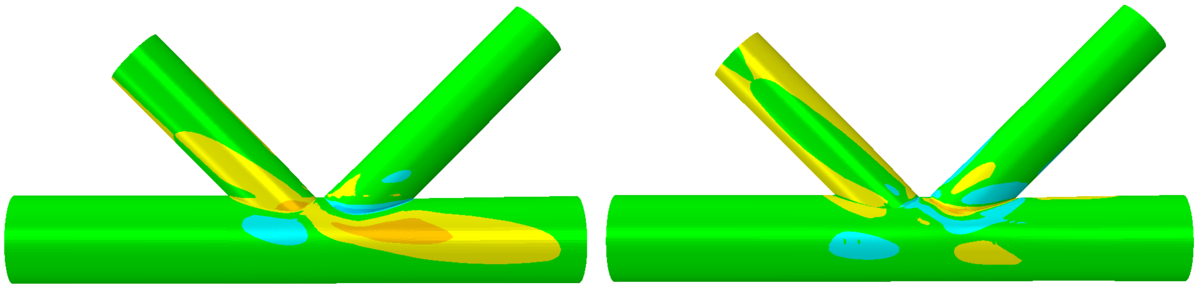
(b) ABQ shell

(c) IFEM $q = 2$

(d) IFEM $q = 3$

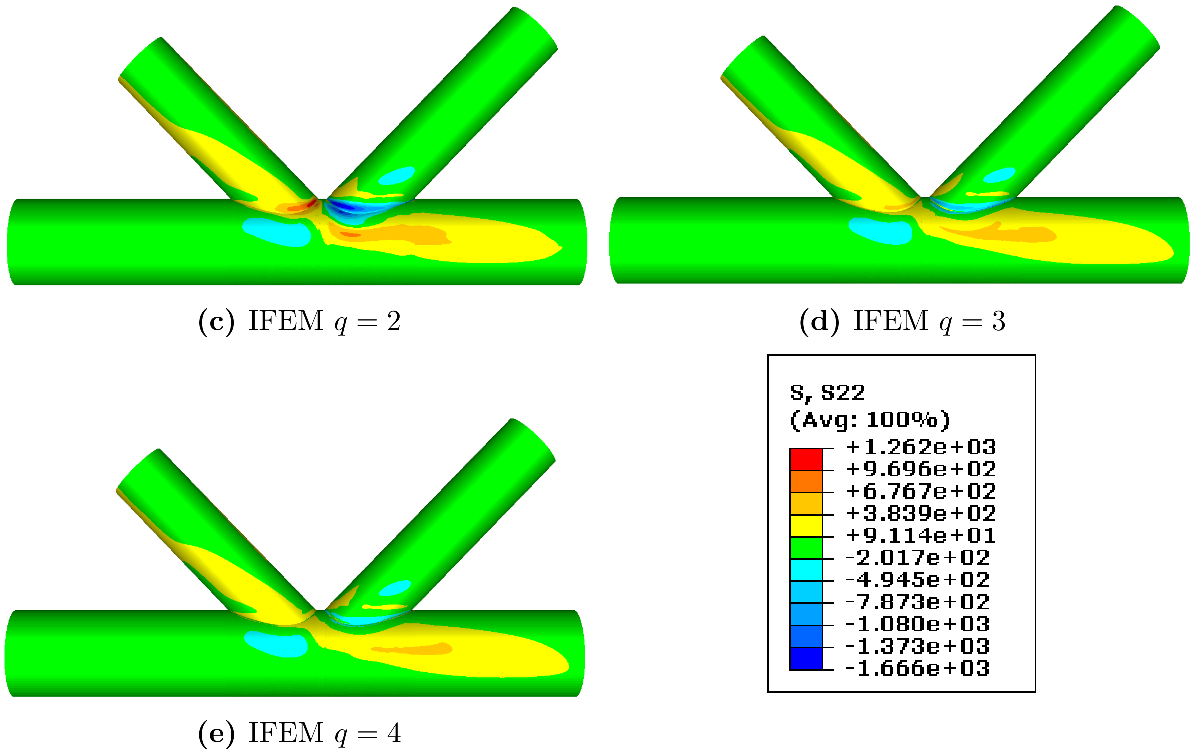
(e) IFEM $q = 4$

Figure 7.11: Stress in local x-direction, σ_{xx}



(a) ABQ

(b) ABQ shell

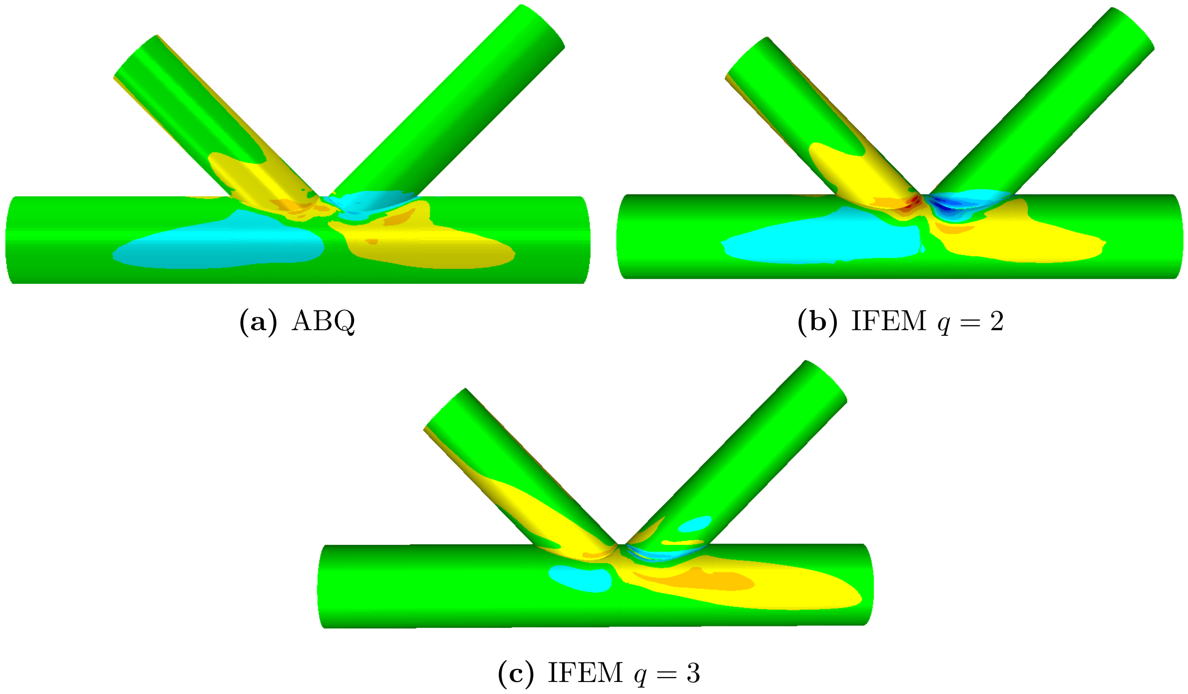


(c) IFEM $q = 2$

(d) IFEM $q = 3$

(e) IFEM $q = 4$

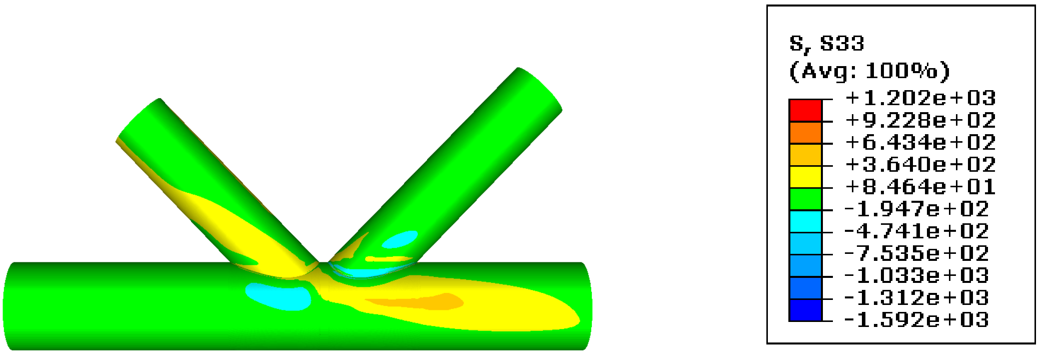
Figure 7.12: Stress in local y-direction, σ_{yy}



(a) ABQ

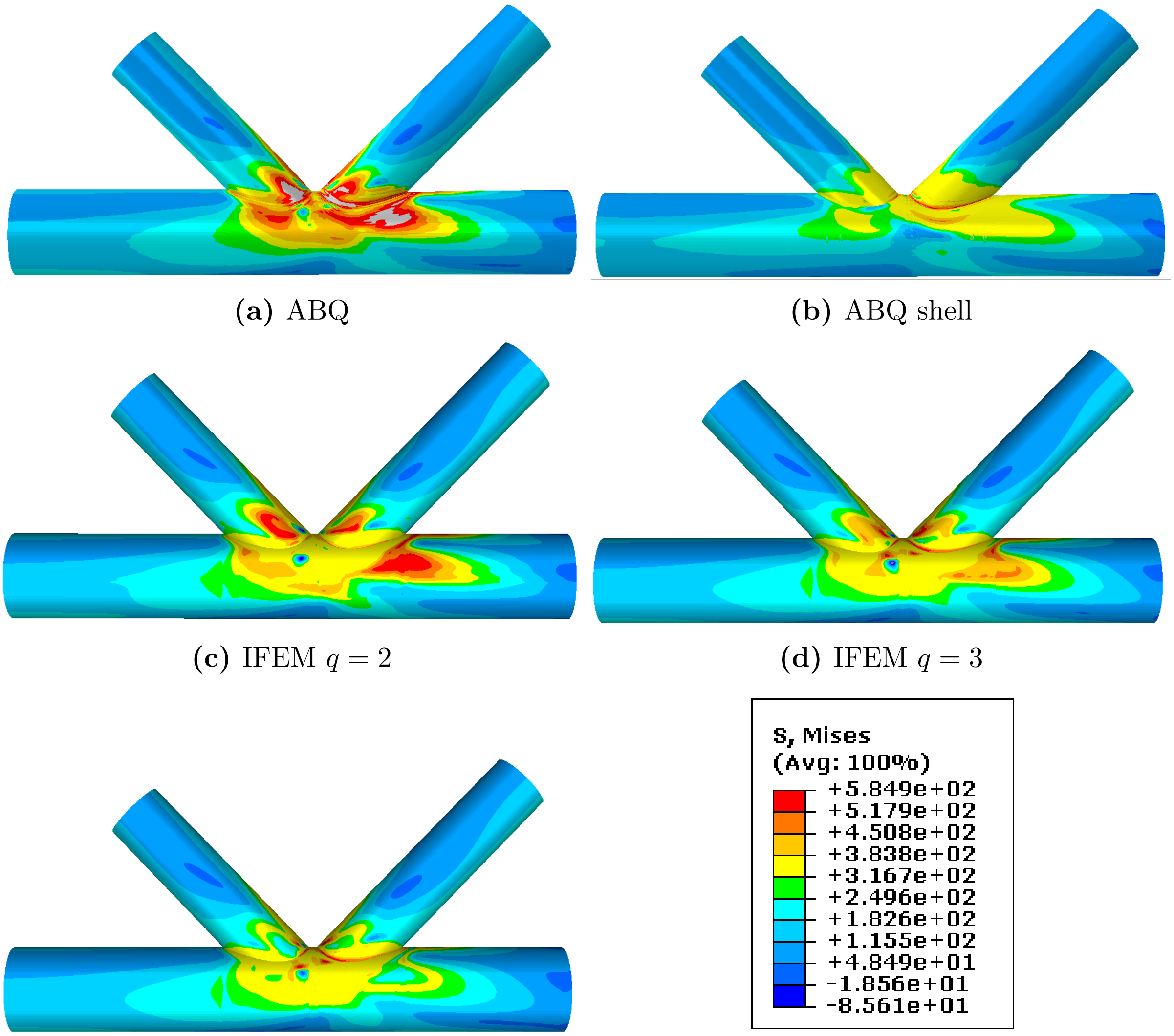
(b) IFEM $q = 2$

(c) IFEM $q = 3$



(d) IFEM $q = 4$

Figure 7.13: Stress in local z-direction, σ_{zz}



(a) ABQ

(b) ABQ shell

(c) IFEM $q = 2$

(d) IFEM $q = 3$

(e) IFEM $q = 4$

Figure 7.14: Von Mises stress, σ_M

Section 7.2 shows that the standard solid FE mesh in Abaqus/Standard can be assumed as an approximate exact solution in comparison with the results from IFEM, see Figure 7.1, and a desirable result for the other analyses would resemble this analysis. Figure 7.10 shows that the results from the IGAs vary in being closer to the standard FE mesh Abaqus/Standard results, indicating no certainty for this K-joint analyses as to the effect of the degree of the basis functions in the IGAs. Furthermore, before making conclusions regarding the degree of the basis functions, the computational time versus the accuracy profit should be taken into account. This is discussed in Section 7.1.

7.5.1 Errors in Values of Derived Stresses and Strains

The definition of the principal stresses given in Equation (4.25) gives limitation as to the magnitude of the principal stresses relative to each other, whereas the von Mises stress is defined to always be positive, cf. Equations (4.41). However, the values of the variables in Table C.1 show that the values of the principal stresses in result node 7 in IFEM for $q = 2$, node 8 for $q = 3$ and node 2 for $q = 4$ are non-descending. Furthermore, the values of the equivalent strain are negative in result nodes 7 and 8 in IFEM for $q = 3$ and nodes 4 through 8 for $q = 4$. In the calculations from Abaqus/Standard, all the values fulfill the mentioned definitions. The method of calculating the derived values in Abaqus/Standard ensures that these errors do not occur, because the mentioned equations are used on the extrapolated values of the components. This assurance is not maintained in IFEM as the derived values are calculated in the integration points and thereafter extrapolated.

Chapter 8

Conclusion

Throughout the discussion of the analyses results, two things related to the difference in shape functions have been shown to have an impact on the results. The major difference is the continuity of the results. The interpolation functions used in traditional solid FEA assures only C^0 continuity between the elements. Hence, the computed stresses and strains will in general be discontinuous between elements, especially in areas where the rate of change of displacement is large. These are the areas often most important to model correctly because they often exhibit the largest values of stresses and strains, as clearly shown in Figure 7.14. In this challenge, the IGA basis functions exhibit a distinct advantage in the higher interelement continuity, providing a better correspondence of the derivative of the variables between the elements. The higher degree of basis functions the higher continuity, and it is a known fact that if an error is present in the primary variable, it propagates and increases as the variable is differentiated. A disadvantage of the increased degree of the basis function is the increased computational time, and the advantage and disadvantage must be weighed against each other.

Another difference is the calculation of the derived variables such as von Mises stress and principal stresses. Because IFEM calculates the von Mises stress in the integration points negative values are possible to obtain in the nodes. This is an evident error and should be avoided, possibly by calculating the von Mises stress as in Abaqus/CAE: directly in the nodes with the extrapolated stress components.

An important feature relating to the geometry of the chord is the nature of the FE mesh refinement in the area surrounding the intersection between the braces and the chord. This is substantiated by the nonconverging discontinuous variables shown for the solid model in Abaqus/Standard.

As to which method of analysis that provides the most accurate and robust result, too few analyses have been performed as a part of this thesis to give a distinct answer. None of the models could be said to be more conservative than the other, and all models exhibited error of external energy slightly greater than preferred.

When it comes to using shell elements as representation of the gap K-joint, few decisive conclusions can be made because they have not been in focus in this thesis. However, the inability of the thin shell elements to represents three dimensional stresses, makes them unsuited to model intersections where the stresses in reality are three dimensional. The well-known solution is to apply submodelling where the intersection area is modelled with solid elements and the outer part of the pipes are modelled with shell elements. This will include the three dimensional stress state in the intersection area, and at the same time provide the advantage of the simplified thin shell theory in the outer parts, which will reduce the computational time.

Chapter 9

Further Work

To create a complete basis for comparison of traditional FEA and IGA more analyses must be performed. Extensions of the analyses could include:

- Optimization of the FE mesh refinements to be exactly equal for the FEA and the IGA.
- IGAs with more refined FE meshes.
- Impact of higher number of elements in the through thickness direction.
- Study of the impact of different patches in the intersection area between the braces and the chord. Fewer patches would eliminate some of the boundaries with only C^0 continuity.
- Study of higher continuity between the patches in the interface.
- Comparison between structures modelled with shell elements in the outer parts of the pipes and submodelled with solid elements in the intersection area to examine the profit of reduced computational time with areas modelled with shell elements.

Bibliography

[1] 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.

[2] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis Toward Integration of CAD and FEA.* Chichester: Wiley, 2009.

[3] "KMB-proposal: ICADA, Integrated Computer Aided Design and Analysis," March 2007.

[4] Ceetron, "Glview inova." Online. http://ceetron.com/products/glview-inova.

[5] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, *Concepts and Applications of Finite Element Analysis.* Singapore ; New York: Wiley (Asia), 4th ed., 2003. 1st edition 1974.

[6] Dassault Systèmes Simulia Corp., Providence, RI, USA, *Abaqus Analysis User's Manual*, 2011.

[7] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method for Solid and Structural Mechanics.* 30 Corporate Drive, Burlington, MA 01803, USA: Elsevier Butterworth-Heinemann, 6.th ed., 2005.

[8] J. F. Kiendl, *Isogeometric Analysis and Shape Optimal Design of Shell Structures.* PhD thesis, Technische Universität München, 2010.

[9] H. Yamashita and H. Sugiyama, "Numerical Convergence of Finite Element Solutions of Nonrational B-Spline Element and Absolute Nodal Coordinate Formulation," *Nonlinear Dynamics*, vol. 67, no. 1, pp. 177–189, 2012.

[10] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, and V. Calo, "The Cost of Continuity: A Study of the Performance of Isogeometric Finite Elements Using Direct Solvers," *Computer Methods in Applied Mechanics and Engineering*, vol. 213–216, no. 0, pp. 353 – 361, 2012.

[11] L. E. Malvern, *Introduction to the Mechanics of a Continous Medium.* Prentice-Hall, 1969.

[12] G. Wempner, *Mechanics of Solids with Applications to Thin Bodies.* Alphen aan den Rijn, The Netherlands: Sjithoff & Noordhoff, 1981.

[13] K. M. Mathisen, "Lecture 12: Formulation of Geometrically Nonlinear FE." It's Learning, October 2012. Lecture Notes in TKT4197 at NTNU.

[14] R. L. Taylor, *FEAP - A Finite Element Analysis Program.* Department of Civil and Environmental Engineering, University of California at Berkeley, Berkeley, California 94720-1710, USA, March 2011.

[15] R. D. Cook and W. C. Young, *Advanced Mechanics of Materials.* Upper Sadler River,

New Jersey 07458, USA: Prentice Hall, 2nd ed., 1999.

[16] O. S. Hopperstad and T. Børvik, "Lecture Notes TKT4135 Mechanics of Materials." Department of Structural Engineering at NTNU, January 2012.

[17] G. R. Johnson and W. H. Cook, "A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rate, and Temperatures," in *International Symposium on Ballistics*, (The Hague, The Netherlands), pp. 1–7, 1983.

[18] Institutt for matematiske fag NTNU, *Calculus 2.* Pearson Education Limited, 2009.

[19] T. J. Hughes, A. Reali, and G. Sangalli, "Efficient Quadrature for NURBS-based Isogeometric Analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, pp. 301–313, December 2008.

[20] C. J. Zarowski, *An Introduction to Numerical Analysis for Electrical and Computer Engineers.* Hoboken, NJ, USA: Wiley, May 2004.

[21] K. M. Mathisen, "Lecture 5: Incremental Reltions for Time-Independent Elasto-Plastic Materials." It's Learning, August 2012. Lecture Notes in TKT4197 at NTNU.

[22] K. M. Mathisen, "Lecture 10: Imposing Constraints." It's Learning, October 2012. Lecture Notes in TKT4197 at NTNU.

[23] K. M. Mathisen, "Lecture 3: Adaptive Solution Algorithms." It's Learning, August 2012. Lecture Notes in TKT4197 at NTNU.

[24] "Additional Notes - Constrained Methods." It's Learning, September 2012.

[25] Dassault Systèmes Simulia Corp., Providence, RI, USA, *Abaqus Theory Manual*, 2011.

[26] O. H. Bjørnøy, "Static Strengt of Tubular Joints, Phase 2 Analyses - and Tests of Gap- and Overlap K-Joints," tech. rep., Det Norske Veritas Industri Norge AS, 1993.

[27] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method for Solid and Structural Mechanics.* 30 Corporate Drive, Burlington, MA 01803, USA: Elsevier Butterworth-Heinemann, 6.th ed., 2005.

[28] Dassault Systèmes Simulia Corp., Providence, RI, USA, *Abaqus/CAE User's Manual*, 2011.

Appendices

Appendix A

Meshes

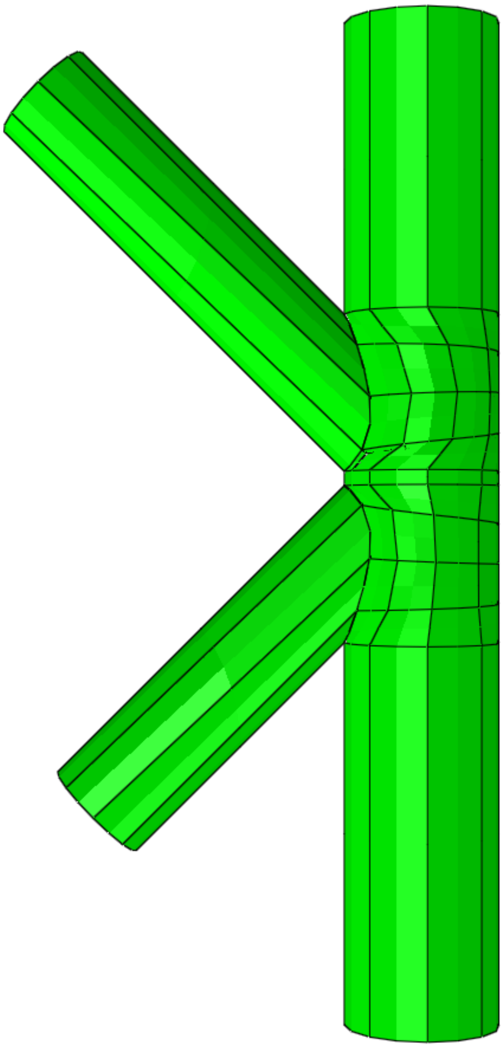


Figure A.1: Very coarse Abaqus/CAE solid

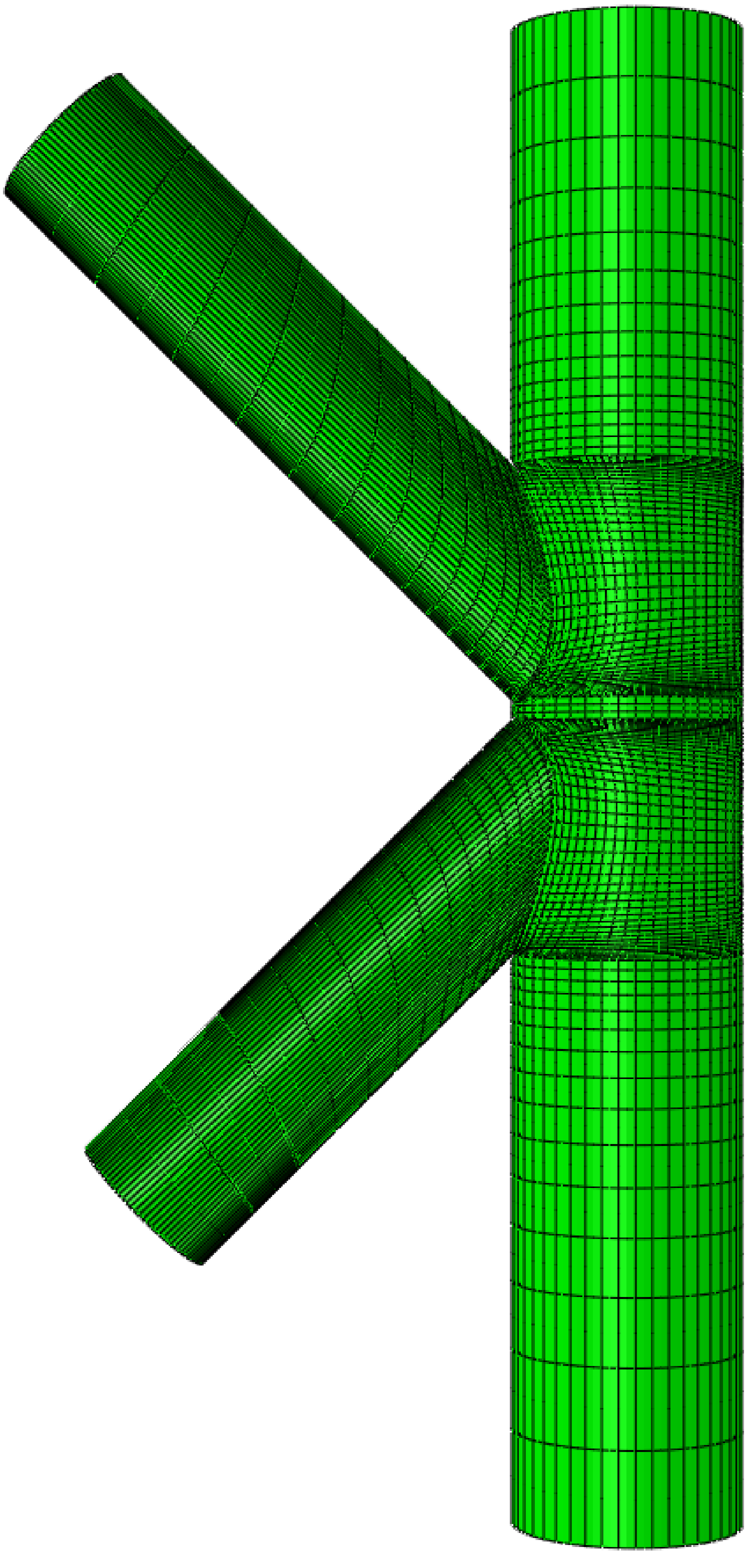


Figure A.2: Standard Abaqus/CAE solid

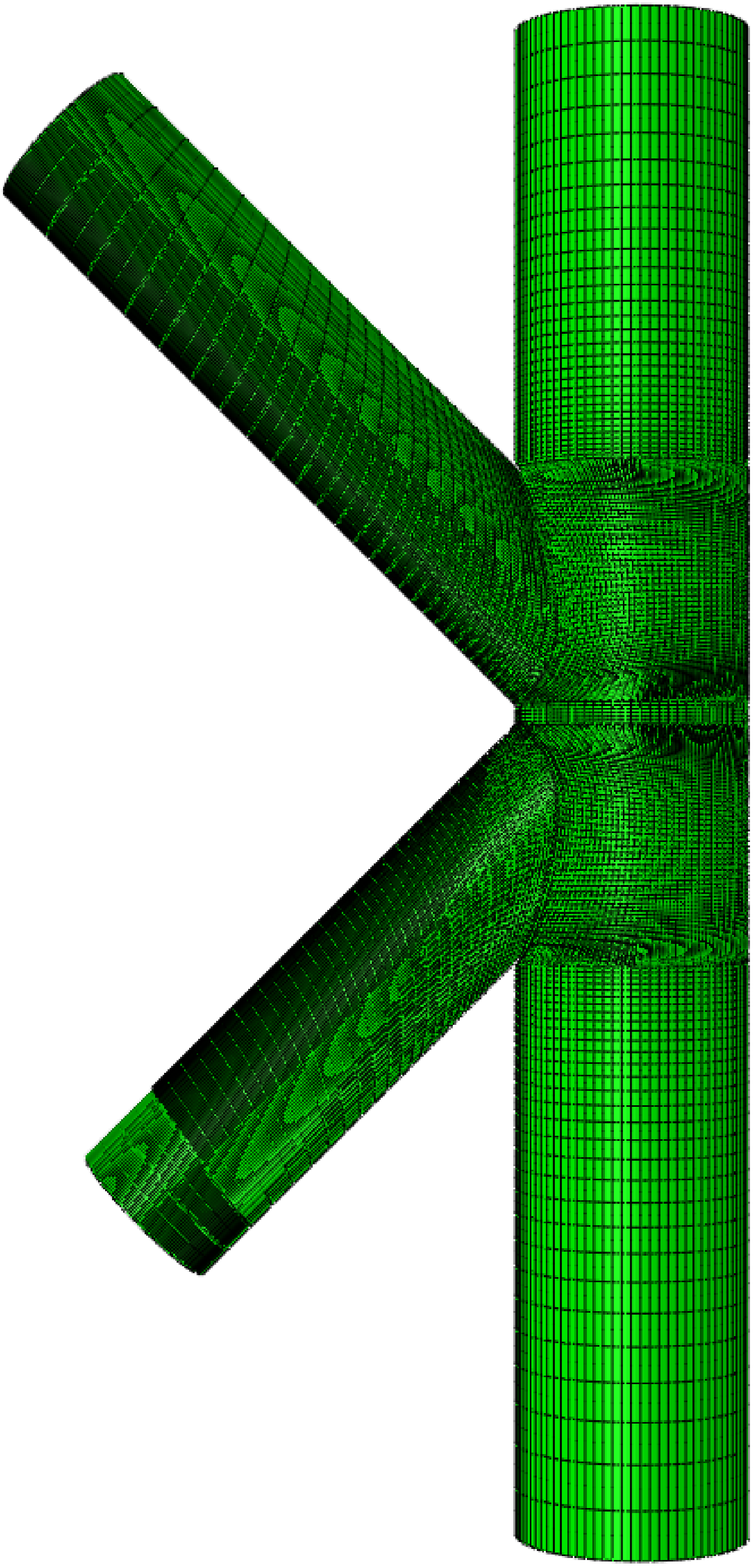


Figure A.3: Fine Abaqus/CAE solid

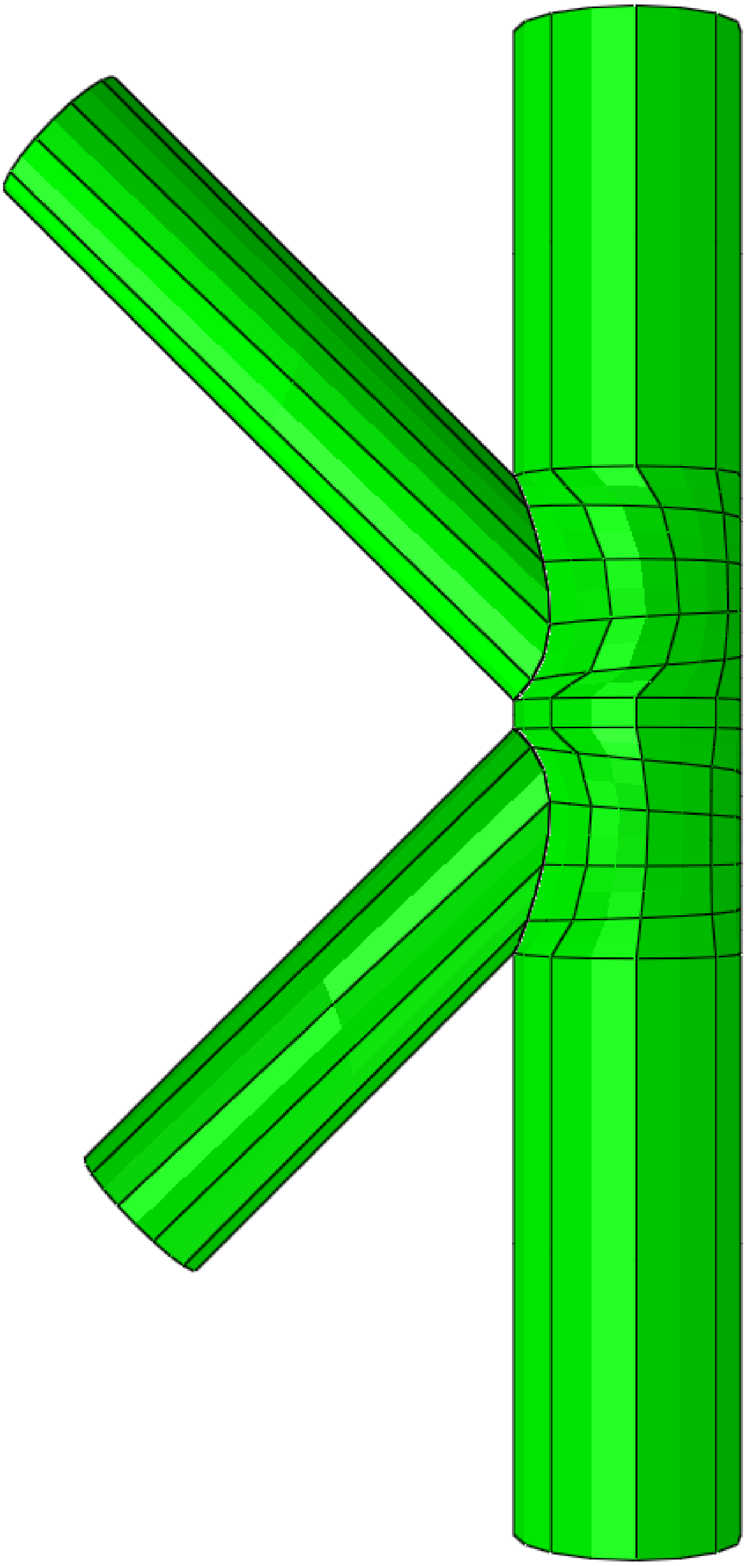


Figure A.4: Very coarse Abaqus/CAE shell

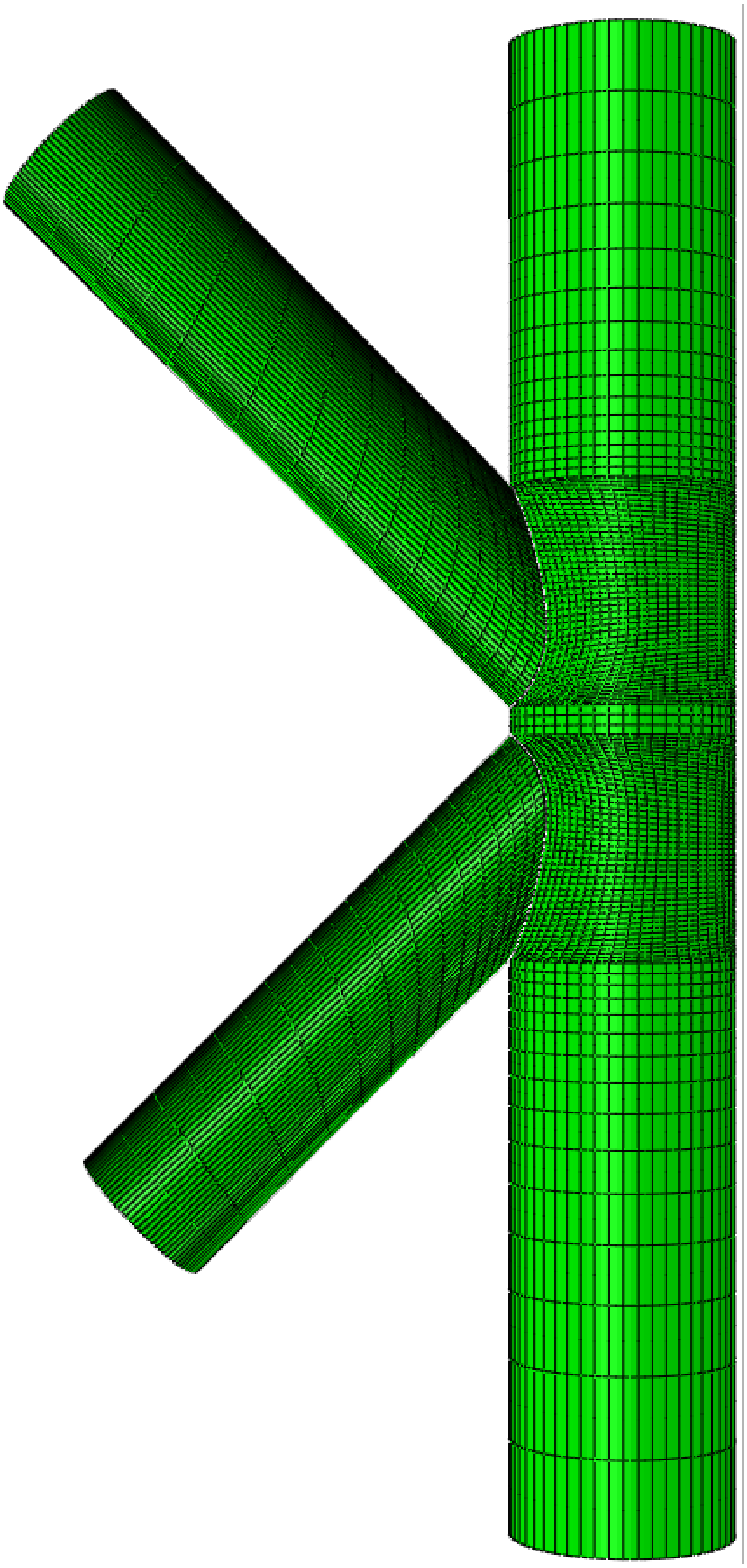


Figure A.5: Standard Abaqus/CAE shell

Appendix B

Local Results

Analysis	ABQ v. coa.	ABQ	ABQ fine	ABQ shell v. coa.	ABQ shell	IFEM $q = 2$	IFEM $q = 3$	IFEM $q = 4$
Max. x-displ.	7.826	11.04	11.14	11.44	12.12	11.10	11.45	11.57
Max. y-displ.	25.74	22.75	22.90	24.84	24.61	22.66	23.24	23.29
Max. z-displ.	8.55	10.33	10.28	8.777	10.35	11.11	10.50	10.42
Max. von Mises stress	1181	974	1294	900.5	755.5	590.6	625.5	582.5
Max. eq. plastic strain	0.08275	0.09431	0.1535	0.03572	0.1373	0.1319	0.1295	0.2541
Max. pressure	338.6	648.8	755.7	359.1	346.7	1173	657.1	958.1
Max. princ. stress	749.1	798.3	891.9	832.0	728.2	13634	949.4	1124
Min. princ. stress	-1085	-1132	-1238	-749.2	-762.3	-1815	-1833	-1956

Table B.1: Global results

Appendix C

Nodal Results

Node	v	w	σ_M	p	σ_h	σ_1	σ_2	σ_3
1ABQ v.c.	9.9152	7.1883	922.63	0.027626	147.39	412.95	-228.85	-626.27
1ABQ	11.308	9.3788	491.35	0.023344	102.4	172.83	-86.647	-393.38
1ABQ f	11.272	9.2447	632.28	0.046115	190.58	165.85	-174.51	-563.08
1ABQ sh. v.c.	-	-	900.46	0.035723	1.4393	509.43	0	-513.74
1ABQ sh.	-	-	541.98	0.038007	123.1	123.26	-21.354	-471.19
1Iq2	11.69	9.6109	423.96	0.053519	119.85	370.69	99.321	-110.46
1Iq3	11.26	9.6109	371.05	0.009921	109.68	255.18	202.13	-128.27
1Iq4	11.24	9.6109	395.25	0.035107	114.2	283.22	176.55	-117.16
2ABQ v.c	12.743	6.8714	872.66	0.030691	-121.48	569.03	208.72	-413.31
2ABQ	11.117	9.4506	444.25	0.013564	-202.49	489.73	113.87	3.8563
2ABQ f.	11.107	9.3596	484	0.020842	-195.93	497.4	143.27	-52.88
2ABQ sh. v.c	-	-	673.81	0.028519	81.388	254.14	0	-498.31
2ABQ sh.	-	-	504.57	0.044263	211.69	64.08	-187.77	-511.38
2Iq2	10.824	9.6751	377.12	0.017613	-456.91	-285.07	-361.18	-724.48
2Iq3	11.129	9.6751	355.11	0.0071443	248.72	474.91	158.9	112.34
2Iq4	11.153	9.6751	335.54	0.0086284	106.67	344.33	-39.488	15.168
3ABQ v.c	8.3617	6.0893	920.83	0.014483	182.43	277.69	-60.771	-764.21
3ABQ	11.864	9.7358	540.11	0.030416	271.43	49.886	-305.82	-558.36
3ABQ f.	11.759	9.5669	533.58	0.032793	343.67	9.6021	-488.71	-551.89
3ABQ sh. v.c.	-	-	674.17	0.044996	428.5	0	-532.42	-753.08
3ABQ sh.	-	-	770.23	0.038511	378.43	0	-270.19	-865.1
3Iq2	12.328	9.8683	472.52	0.13187	-527.73	-205.1	-645.93	-732.17
3Iq3	11.682	9.8683	484.04	0.12792	-309.31	3.3683	-398.83	-532.47
3Iq4	11.636	9.8683	512.07	0.2541	-461.63	-158.74	-483.71	-742.44
4ABQ v.c.	14.209	5.8994	838.16	0.0079399	-236.34	723.65	210.84	-225.46
4ABQ	10.77	9.6153	461.21	0.00403	-83.421	378.29	8.8449	-136.87
4ABQ f.	10.865	9.4727	458.44	0.017469	-131.05	415.94	74.399	-97.177
4ABQ sh. v.c.	-	-	633.16	0.032858	-410.65	687.13	544.83	0
4ABQ sh.	-	-	548.28	0.021786	-322.6	627.66	340.13	0
4Iq2	10.454	9.6761	368.78	0.011104	-173.59	48.467	-190.88	-378.35
4Iq3	11.017	9.6761	339.34	0.0081299	-9.6309	209.65	-84.783	-153.76
4Iq4	11.047	9.6761	346.55	0.0097696	220.77	431.5	206.13	24.681
5ABQ v.c	17.122	1.0666	407.53	0	-47.407	272.54	63.747	-194.07
5ABQ	19.929	0.94	130.08	0	-23.272	107.3	-0.15249	-37.329
5ABQ f.	20.053	0.92301	129.4	0	-24.296	108.02	0.44676	-35.578
5ABQ sh. v.c.	-	-	153.48	0	-23.453	120.97	0	-50.608
5ABQ sh.	-	-	121.46	0	-18.914	96.556	0	-39.813
5Iq2	19.871	0.89361	103.47	2.6231e-07	10.493	73.074	1.434	-43.028
5Iq3	20.275	0.89361	132.44	2.9779e-07	25.948	115.1	-3.7414	-33.512
5Iq4	20.315	0.89361	124.69	-1.2147e-06	20.558	92.693	8.2252	-39.245
6ABQ v.c.	16.776	1.0286	216.35	0	-38.111	172.23	16.575	-74.468
6ABQ	19.783	0.93104	129.92	0	-25.674	110.23	-0.35525	-32.852
6ABQ f.	19.907	0.91417	129.07	0	-26.848	110.99	0.33988	-30.791
6ABQ sh. v.c.	-	-	153.99	0	-42.452	141.81	15.109	-29.562
6ABQ sh.	-	-	120.58	0	-24.815	103.44	0	-28.989
6Iq2	19.73	0.88483	102.39	2.2258e-07	14.32	77.972	1.9723	-36.984
6Iq3	20.131	0.88483	134.01	1.3568e-07	29.122	118.87	-2.343	-29.16
6Iq4	20.172	0.88483	122.37	-5.8943e-07	22.134	94.142	7.0781	-34.817
7ABQ v.c.	17.307	0.88875	200.38	3.2399e-05	35.078	82.375	-39.406	-148.21
7ABQ	19.983	0.88835	102.88	0	67.041	1.5361	-100.32	-102.34
7ABQ f.	20.107	0.87112	102.62	0	67.58	0.81223	-100.37	-103.18
7ABQ sh. v.c.	-	-	157.84	0	67.1	3.8583	-40.872	-164.29
7ABQ sh.	-	-	101.11	0	60.505	0	-66.378	-115.14
7Iq2	19.922	0.84489	121.31	1.8424e-06	-74.728	7.5821	-123.76	-108
7Iq3	20.326	0.84489	104.28	-4.4031e-06	-66.424	4.4275	-101.27	-102.43
7Iq4	20.367	0.84489	98.455	-1.1926e-05	-65.237	-6.2822	-87.417	-102.01
8ABQ v.c.	16.623	0.96443	267.26	0	25.201	108.7	9.1665	-193.47
8ABQ	19.727	0.87251	99.945	0	64.528	1.8434	-92.637	-102.79
8ABQ f.	19.85	0.85547	99.731	0	65.309	0.89257	-93.082	-103.74
8ABQ sh. v.c.	-	-	132.4	0	64.627	0	-52.398	-141.48
8ABQ sh.	-	-	98.717	0	61.984	0	-74.419	-111.53
8Iq2	19.677	0.82822	131.91	1.3323e-06	-81.057	7.1385	-124.12	-126.19
8Iq3	20.077	0.82822	102.97	-3.1796e-06	-64.594	5.8667	-103.84	-95.812
8Iq4	20.118	0.82822	96.854	-8.7856e-06	-63.81	-8.3384	-86.271	-96.822

Table C.1: Local results in node 1 through 8 according to Figure 6.9 for the analyses in Abaqus/Standard with very coarse, standard and fine mesh and the analyses in IFEM with $q = 2$, $q = 3$ and $q = 4$. All displacements in mm, all stresses in MPa. Nomination: ABQ = Abaqus/Standard, v.c. = very coarse mesh, f. = fine mesh, sh. = shell, I = IFEM, q = degree of basis functions

Appendix D

Incremental-Iterative Solution Algorithm for Normal Plane Arc-Length Methods

INITIALIZATION

- NP: Normal Plan method, UNP: Updated Normal Plane method, MNR: Modified Newton-Raphson
- Prescribe arc-length l, minimum and maximum load proportionality factor (LPF) λ_{min} and λ_{max}, convergence tolerance ϵ_{tol}, number of load steps n_{nstep}, and maximum number of iterations per step n_{iter}
- Initialize load step counter $n := 0$, displacements $\{\boldsymbol{D}_0\} := \{\boldsymbol{0}\}$, $\{\Delta\boldsymbol{D}_0\} := \{\boldsymbol{0}\}$ and LPF $\lambda_0 := 0$
- Assemble external load gradient $\left\{\bar{\boldsymbol{f}}^{ext}\right\}$ (needs to be recalculated at every iteration if the external loading is non-proportional or if prescribed displacements are present)

PREDICTOR STEP

while $\lambda_{min} \leq \lambda_n \leq \lambda_{max}$ **and** $n \leq n_{nstep}$ **do**

- ○ Set $n := n + 1$
- ○ Assemble stiffness and residual forces:

$$\begin{cases} \left[\boldsymbol{K}^0_{T,n}\right] := [\boldsymbol{K}_T(\boldsymbol{D}_{n-1})] \\ \left\{\boldsymbol{R}^0_n\right\} := \lambda_{n-1}\left\{\bar{\boldsymbol{f}}^{ext}\right\} - \{\boldsymbol{f}^{int}(\boldsymbol{D}_{n-1})\} \end{cases}$$

- ○ Solve for incremental changes of residual and linearized displacements:

$$\begin{cases} \left\{\Delta\tilde{\boldsymbol{D}}^0_n\right\} = \left[\boldsymbol{K}^0_{T,n}\right]^{-1}\left\{\boldsymbol{R}^0_n\right\} \\ \left\{\Delta\bar{\boldsymbol{D}}^0_n\right\} = \left[\boldsymbol{K}^0_{T,n}\right]^{-1}\left\{\bar{\boldsymbol{f}}^{ext}\right\} \end{cases}$$

- ○ Computer LPF and displacement:

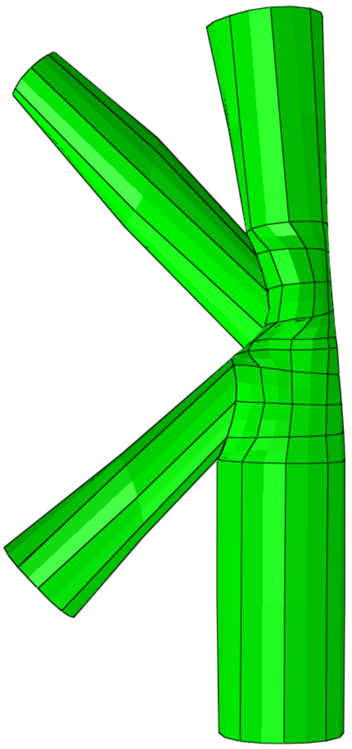
$$\begin{cases} \Delta\lambda^0_n := sign\left(\{\Delta\boldsymbol{D}_{n-1}\}^T\left\{\Delta\bar{\boldsymbol{D}}^0_n\right\}\right)\frac{l_n}{\sqrt{1+\left\{\Delta\bar{\boldsymbol{D}}^0_n\right\}^T\left\{\Delta\bar{\boldsymbol{D}}^0_n\right\}}} \\ \left\{\Delta\boldsymbol{D}^0_n\right\} := \left\{\Delta\tilde{\boldsymbol{D}}^0_n\right\} + \Delta\lambda^0_n\left\{\Delta\bar{\boldsymbol{D}}^0_n\right\} \end{cases}$$

- ○ Update LPF, displacement and residual forces:

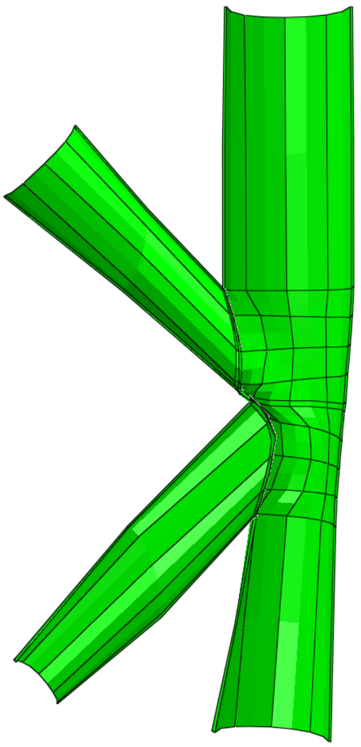
```
                $\begin{cases} \lambda_n^0 := \lambda_{n-1} + \Delta\lambda_n^0 \\ \left\{\boldsymbol{D}_n^0\right\} := \{\boldsymbol{D}_{n-1}\} + \left\{\Delta\boldsymbol{D}_n^0\right\} \\ \left\{\boldsymbol{R}_n^0\right\} := \lambda_n^0 \left\{\bar{\boldsymbol{f}}^{ext}\right\} + \left\{\boldsymbol{f}^{int}\left(\boldsymbol{D}_n^0\right)\right\} \end{cases}$
    ○ Initialize iteration counter $i := 0$
  CORRECTOR STEP
    while $\| \left\{\boldsymbol{R}_n^i\right\} \| > \epsilon_{tol} \| \lambda_n^i \left\{\bar{\boldsymbol{f}}^{ext}\right\} \|$ and $i \leq n_{iter}$ do
      • Set $i := i + 1$
      • Assemble stiffness:
      if MNR then
          $\left[\boldsymbol{K}_{T,n}^i\right] := [\boldsymbol{K}_T(\boldsymbol{D}_{n-1})]$
      else
          $\left[\boldsymbol{K}_{T,n}^i\right] := \left[\boldsymbol{K}_T(\boldsymbol{D}_n^{i-1})\right]$
      end if
      • Solve for iterative changes of the residual and linearized displacements:
                $\begin{cases} \left\{\delta\tilde{\boldsymbol{D}}_n^i\right\} = \left[\boldsymbol{K}_{T,n}^i\right]^{-1} \left\{\boldsymbol{R}_n^{i-1}\right\} \\ \left\{\delta\bar{\boldsymbol{D}}_n^i\right\} = \left[\boldsymbol{K}_{T,n}^i\right]^{-1} \left\{\bar{\boldsymbol{f}}^{ext}\right\} \end{cases}$
      • Compute iteratice LPF:
      if UNP then
          $\delta\lambda_n^i := -\frac{\left\{\Delta\boldsymbol{D}_n^{i-1}\right\}^T \left\{\delta\tilde{\boldsymbol{D}}_n^i\right\}}{\left\{\Delta\boldsymbol{D}_n^{i-1}\right\}^T \left\{\delta\bar{\boldsymbol{D}}_n^i\right\} + \Delta\lambda_n^{i-1}}$
      else
          $\delta\lambda_n^i := -\frac{\left\{\Delta\boldsymbol{D}_n^0\right\}^T \left\{\delta\tilde{\boldsymbol{D}}_n^i\right\}}{\left\{\Delta\boldsymbol{D}_n^0\right\}^T \left\{\delta\bar{\boldsymbol{D}}_n^i\right\} + \Delta\lambda_n^0}$
      end if
      • Update iterative displacements: $\left\{\delta\boldsymbol{D}_n^i\right\} := \left\{\delta\tilde{\boldsymbol{D}}_n^i\right\} + \delta\lambda_n^i \left\{\delta\bar{\boldsymbol{D}}_n^i\right\}$
      • Update incremental and total LPF and displacement, and residual forces:
                $\begin{cases} \Delta\lambda_n^i := \Delta\lambda_n^{i-1} + \delta\lambda_n^i \\ \lambda_n^i := \lambda_n^{i-1} + \delta\lambda_n^i \\ \left\{\Delta\boldsymbol{D}_n^i\right\} := \left\{\Delta\boldsymbol{D}_n^{i-1}\right\} + \left\{\delta\boldsymbol{D}_n^i\right\} \\ \left\{\boldsymbol{D}_n^i\right\} := \left\{\boldsymbol{D}_n^{i-1}\right\} + \left\{\delta\boldsymbol{D}_n^i\right\} \\ \left\{\boldsymbol{R}_n^i\right\} := \lambda_n^i \left\{\bar{\boldsymbol{f}}^{ext}\right\} + \left\{\bar{\boldsymbol{f}}^{int}\left(\boldsymbol{D}_n^i\right)\right\} \end{cases}$
    end while
    ○ Set: $\begin{cases} \lambda_n := \lambda_n^i \\ \{\Delta\boldsymbol{D}_n\} := \{\Delta\boldsymbol{D}_n^i\} \\ \{\boldsymbol{D}_n\} := \{\boldsymbol{D}_n^i\} \end{cases}$
  end while
```

Appendix E

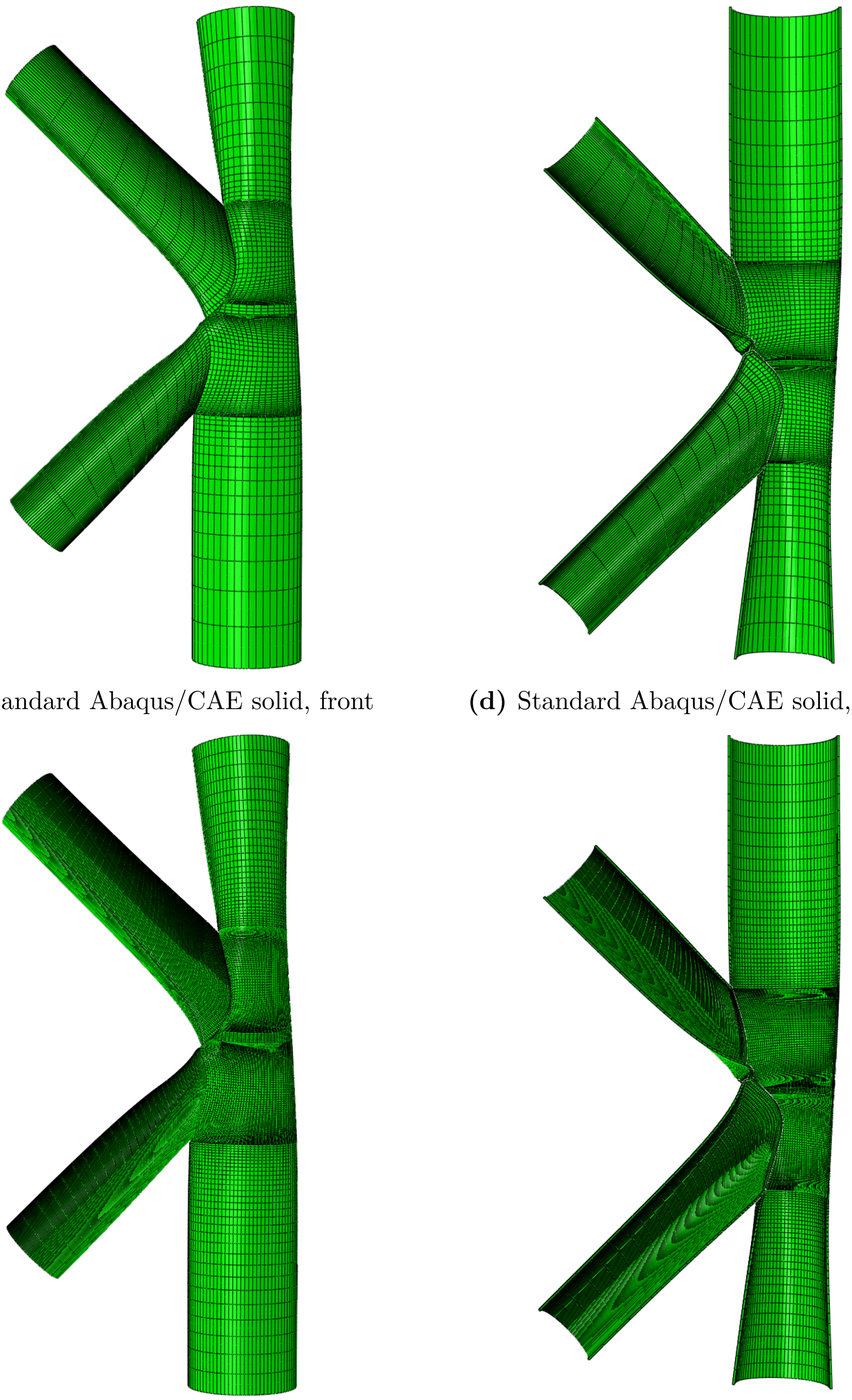
Deformation patterns



(a) Very coarse Abaqus/CAE solid, front



(b) Very coarse Abaqus/CAE solid, back

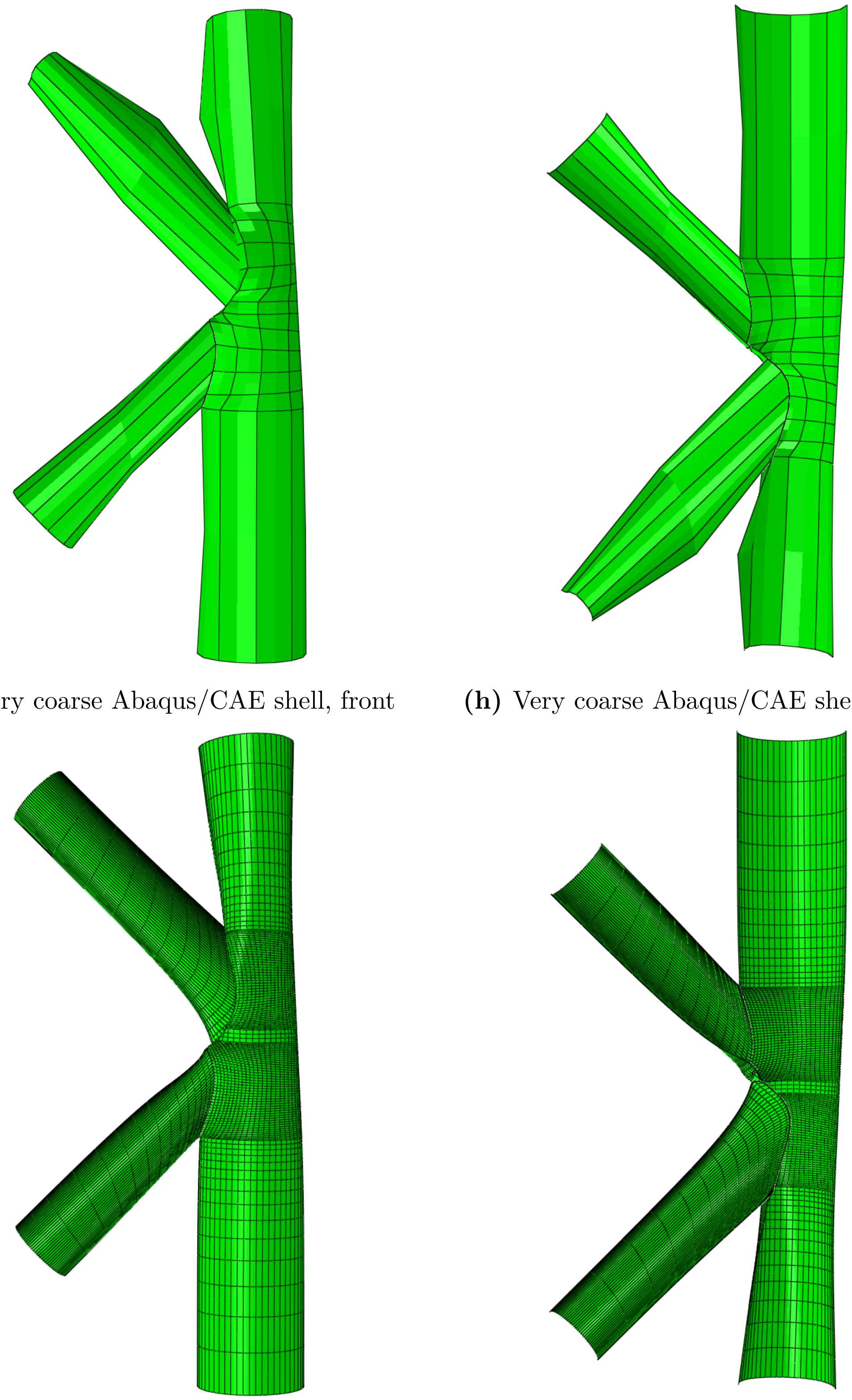


(c) Standard Abaqus/CAE solid, front

(d) Standard Abaqus/CAE solid, back

(e) Fine Abaqus/CAE solid, front

(f) Fine Abaqus/CAE solid, back

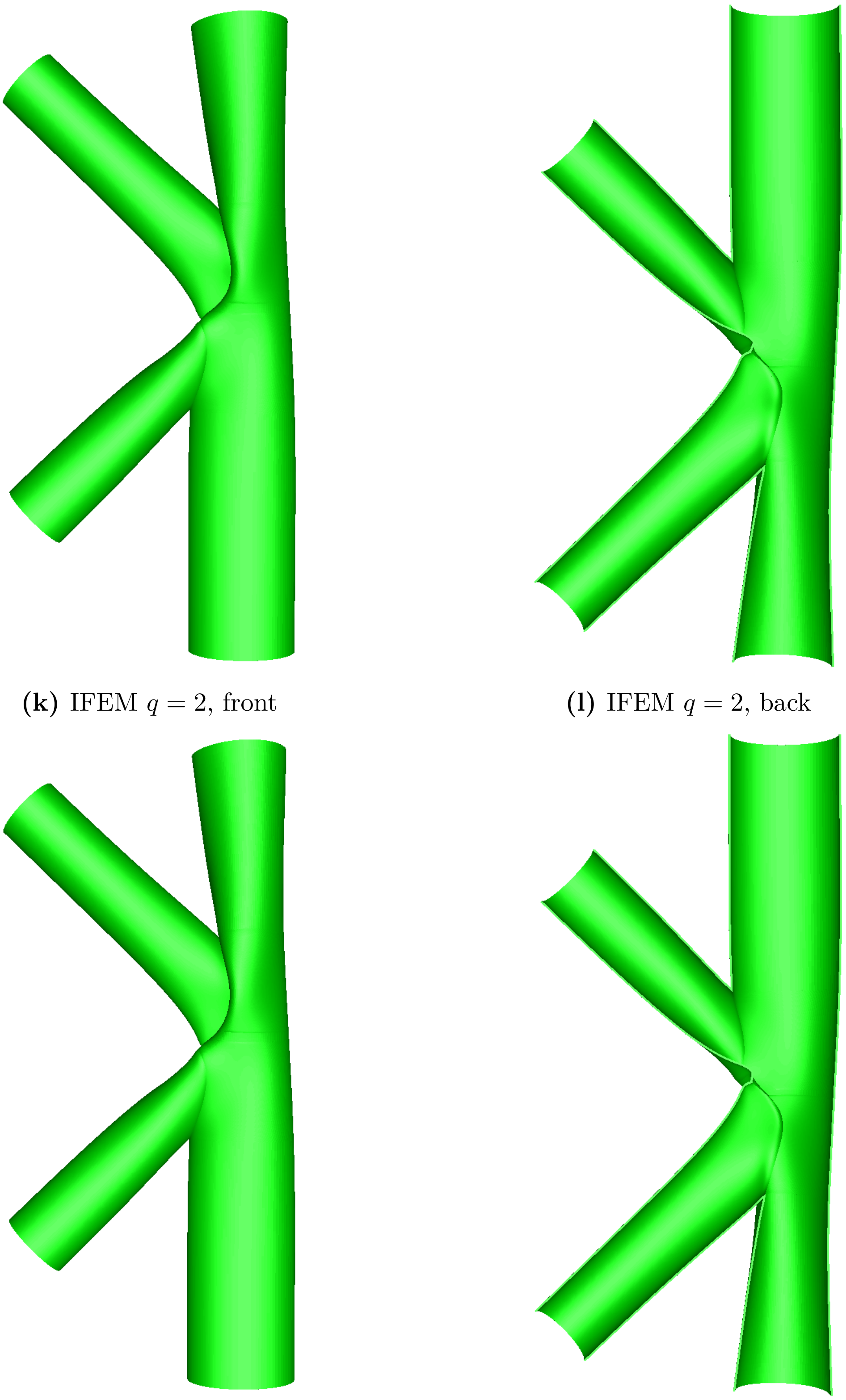


(g) Very coarse Abaqus/CAE shell, front

(h) Very coarse Abaqus/CAE shell, back

(i) Standard Abaqus/CAE shell, front

(j) Standard Abaqus/CAE shell, back

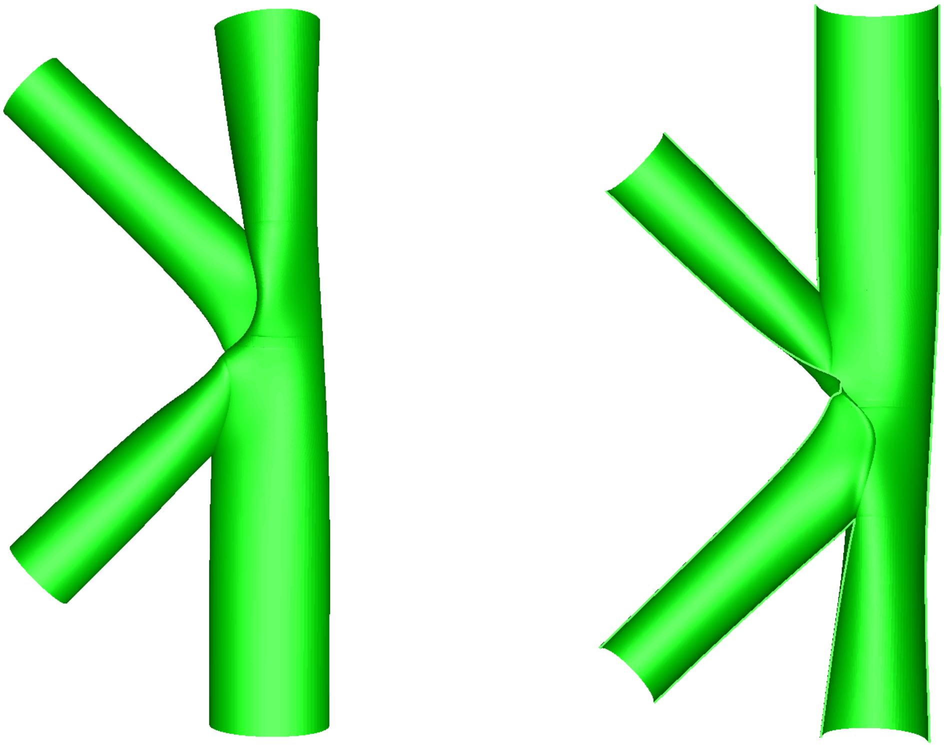


(k) IFEM $q = 2$, front

(l) IFEM $q = 2$, back

(m) IFEM $q = 3$, front

(n) IFEM $q = 3$, back



(o) IFEM $q = 4$, front

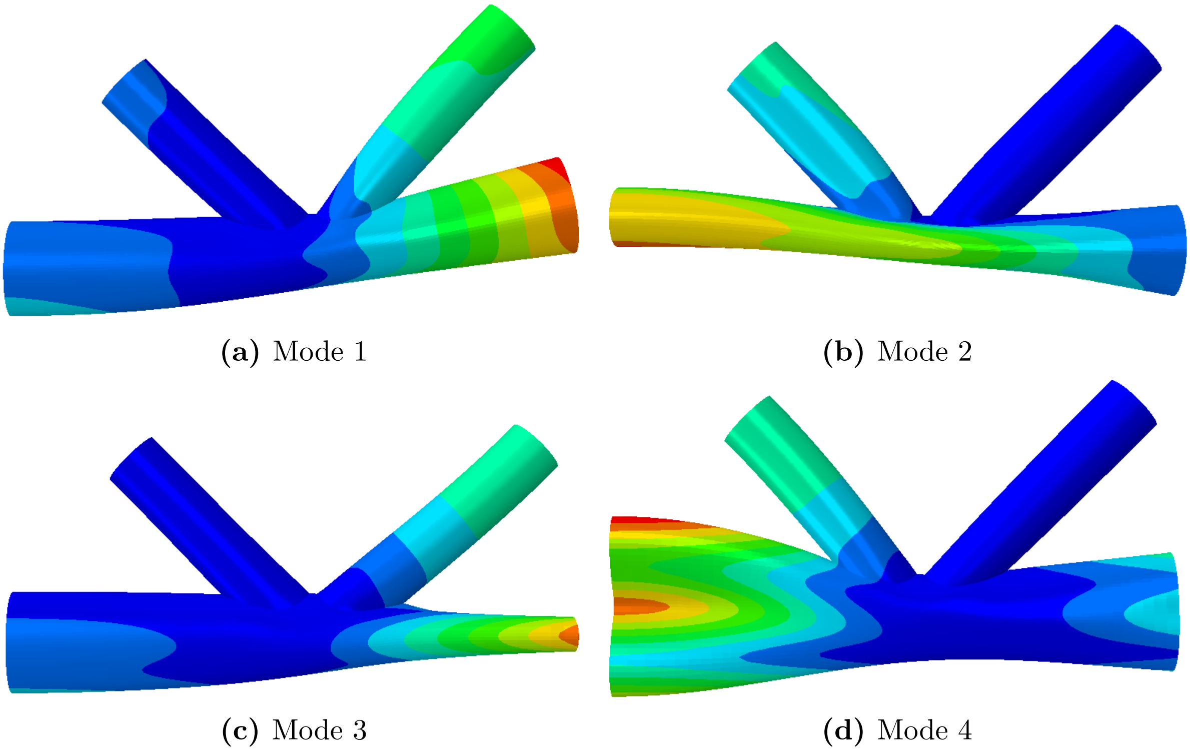
(p) IFEM $q = 4$, back

Figure E.1: Deformation patterns from the solid analyses in Abaqus/Standard with very coarse, standard and fine mesh, the Abaqus/Standard analyses with shell elements and very coarse and standard mesh and the analyses from IFEM with $q = 2$, $q = 3$ and $q = 4$

Appendix F

Mode Shapes

In this appendix the mode shapes for the standard model in Abaqus/Standard are shown.



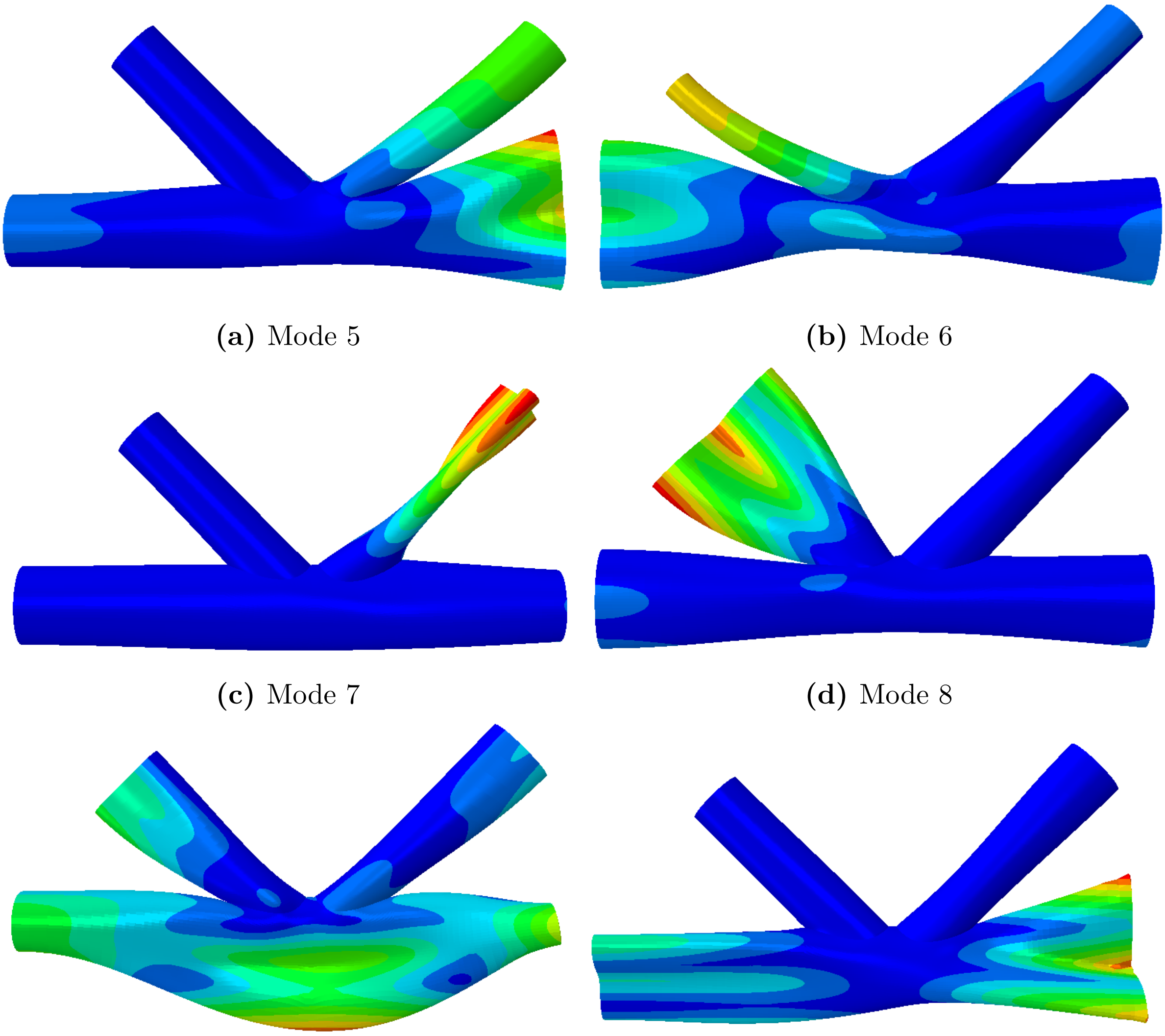
(a) Mode 1

(b) Mode 2

(c) Mode 3

(d) Mode 4

Figure F.1: Mode shapes 1 through 4



(a) Mode 5

(b) Mode 6

(c) Mode 7

(d) Mode 8

(e) Mode 9

(f) Mode 10

Figure F.2: Mode shapes 5 through 10