Universidade de Aveiro Departamento de Engenharia Mecânica
2014

João Filipe Moreira Caseiro

Ferramentas Numéricas para Análise Isogeométrica em Regime Não-Linear

Numerical Tools for Isogeometric Analysis in the Nonlinear Regime

Universidade de Aveiro Departamento de Engenharia Mecânica
2014

João Filipe Moreira Caseiro

Ferramentas Numéricas para Análise Isogeométrica em Regime Não-Linear

Numerical Tools for Isogeometric Analysis in the Nonlinear Regime

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Mecânica, realizada sob a orientação científica do Doutor Robertt Ângelo Fontes Valente e do Doutor Ricardo José Alves de Sousa, Professores Auxiliares do Departamento de Engenharia Mecânica da Universidade de Aveiro

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

o júri

presidente

Prof. Doutor Artur Manuel Soares da Silva
Professor Catedrático da Universidade de Aveiro

Prof. Doutor Renato Manuel Natal Jorge
Professor Associado com Agregação da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor Alessandro Reali
Professor Associado da Universidade de Pavia, Itália

Prof. Doutora Marta Cristina Cardoso de Oliveira
Professora Auxiliar da Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Prof. Doutor António Gil d'Orey de Andrade-Campos
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor Robertt Ângelo Fontes Valente
Professor Auxiliar da Universidade de Aveiro (orientador)

agradecimentos

Ao meu orientador Professor Robertt Valente, com quem tive o enorme prazer (e privilégio) de ter trabalhado, pelo desafio que me lançou e por todo o apoio prestado ao longo deste percurso. O seu constante acompanhamento, empenho e amizade contribuiram grandemente para o trabalho aqui apresentado e para o meu crescimento científico e pessoal. O meu sincero muito obrigado.

Ao Professor Ricardo Alves de Sousa, co-orientador e fonte constante de boa disposição, pela disponibilidade, apoio e amizade durante este percurso e por ter sempre a porta aberta para responder a todas as questões.

To Doctor Josef Kiendl, Professor Alessandro Reali and Professor Ferdinando Auricchio from the University of Pavia for all the productive scientific discussions, for sharing their expertise and for their friendship.

À Professora Marta Oliveira e ao Doutor Diogo Neto da Universidade de Coimbra pela disponibilidade e discussões científicas sobre Mecânica do Contacto.

To Professor Laura de Lorenzis from the Technische Universität Braunschweig and Professor Thomas Elguedj from the Université de Lyon for the valuable email exchanges and meetings.

Ao Professor João Oliveira pelas sugestões e apoio na correcção desta tese.

À Fundação para a Ciência e a Tecnologia (FCT), pelo apoio financeiro prestado no âmbito da Bolsa SFRH/BD/70815/2010.

A todos aqueles que fizeram parte do meu percurso no Departamento de Engenharia Mecânica da Universidade de Aveiro e em especial a todos aqueles com quem tive o prazer de partilhar o gabinete e as horas de almoço. Obrigado pela amizade, pela boa disposição, pelas discussões científicas (e pelas não científicas) e por todos os momentos de competição amigável.

À Filipa, parte tão importante da minha vida, por todo o apoio, carinho e incentivo. Ao pequeno Artur, a quem dedico esta tese.

palavras-chave

análise isogeométrica, elemento sólido-casca, método das deformações assumidas, retenção, mecânica do contacto, algoritmo ponto-para-segmento.

resumo

O presente trabalho foca-se no desenvolvimento de ferramentas numéricas robustas para problemas não-lineares de mecânica dos sólidos no contexto de Análises Isogeométricas. Com esse intuito, um novo elemento do tipo sólido-casca, baseado no método das Deformações Assumidas, é proposto para a análise de estruturas do tipo casca fina. A formulação proposta é validada recorrendo a um conjunto de problemas-tipo disponíveis na literatura, considerando tanto regimes lineares como não-lineares (geométrico e de material). É ainda apresentada uma formulação alternativa para aliviar o fenómeno de retenção volumétrica para problemas em regime linear elástico. Adicionalmente, é apresentado um estudo introdutório da mecânica do contacto no contexto de Análises Isogeométricas, dando especial ênfase ao algoritmo de Ponto-para-Segmento. As metodologias apresentadas no presente trabalho foram implementadas num código totalmente desenvolvido durante o decorrer do mesmo, juntamente com diversas ferramentas para pré- e pós processamento. Foram ainda implementadas rotinas do utilizador para o software comercial Abaqus.

Keywords isogeometric analysis, solid-shell element, assumed natural strain method, locking, contact mechanics, point-to-segment algorithm.

Abstract

The present work deals with the development of robust numerical tools for Isogeometric Analysis suitable for problems of solid mechanics in the nonlinear regime. To that end, a new solid-shell element, based on the Assumed Natural Strain method, is proposed for the analysis of thin shell-like structures. The formulation is extensively validated using a set of well-known benchmark problems available in the literature, in both linear and nonlinear (geometric and material) regimes. It is also proposed an alternative formulation which is focused on the alleviation of the volumetric locking pathology in linear elastic problems. In addition, an introductory study in the field of contact mechanics, in the context of Isogeometric Analysis, is also presented, with special focus on the implementation of a the Point-to-Segment algorithm. All the methodologies presented in the current work were implemented in a in-house code, together with several pre- and post-processing tools. In addition, user subroutines for the commercial software Abaqus were also implemented.

Contents

List of Figures

List of Tables

Nomenclature

List of Indices

$(\cdot)^1$	Master body
$(\cdot)^2$	Slave body
$(\cdot)^e$	Entity in a Finite Element domain
$(\cdot)^h$	Finite Element approximation of an entity
$(\cdot)^\alpha, (\cdot)^{\alpha\alpha}$	Enhanced-based entity
$(\cdot)^u, (\cdot)^{uu}$	Displacement-based entity
$(\cdot)^{u\alpha}, (\cdot)^{\alpha u}$	Coupled displacement-enhanced entity
$(\cdot)^{\text{int}}, (\cdot)_{\text{int}}$	Stands for internal
$(\cdot)^{\text{ext}}, (\cdot)_{\text{ext}}$	Stands for external
$(\cdot)^{\text{vol}}$	Stands for volumetric
$(\cdot)^{\text{dev}}$	Stands for deviatoric
$(\cdot)^{\text{el}}$	Stands for elastic
$(\cdot)^{\text{pl}}$	Stands for plastic
$(\cdot)_{\text{C}}$	Stands for contact contribution
$(\cdot)_{\text{NL}}$	Stands for geometric nonlinear
$(\cdot)^*$	Entity in an enriched basis
$\hat{(\cdot)}$	Entity evaluated at the local corotational coordinate system
$\tilde{(\cdot)}$	Entity evaluated at the local covariant coordinate system
$\bar{(\cdot)}$	Entity evaluated at the parent coordinate system/Closest Point Projection
$\breve{(\cdot)}$	Entity evaluated at the collocation point
$\dot{(\cdot)}$	Time derivative of $(\cdot)$
${}^{n+1}(\cdot)$	Entity evaluated at state $n+1$
${}^{n+1}_{\;\;n}(\cdot)$	Increment of the entity between states n and $n+1$

List of Mathematical Symbols

$\text{div}\,(\cdot)$	Divergence of $(\cdot)$
$\text{grad}\,(\cdot)$	Gradient of $(\cdot)$

$\mathrm{tr}\,(\cdot)$	Trace of $(\cdot)$
$\lvert\,(\cdot)\,\rvert$	Determinant of $(\cdot)$
$\lVert\,(\cdot)\,\rVert$	Norm of $(\cdot)$
$(\cdot)^{\mathrm{T}}$	Transpose of $(\cdot)$
$(\cdot)^{-1}$	Inverse of $(\cdot)$
$\delta\,(\cdot)$	Variation of $(\cdot)$/virtual quantity
$(\cdot)_{,i}\,,(\cdot)_{,ij}$	First/second derivative of $(\cdot)$ with respect to *i*/*i* and *j*
$\boldsymbol{\nabla}_{\mathrm{s}}$	Differential operator
δ_{ij}	Kronecker delta

List of Symbols

$\mathbf{a}$	Tangent vector
$\bar{a}_{11}$	Metric of a boundary
$\boldsymbol{\alpha}$	Internal variables field of the EAS method
$\mathbf{B}$	Strain-displacement matrix
$\mathbf{B}_i, \mathbf{B}_{ij}, \mathbf{B}_{ijk}$	Control polygon, net and lattice
$\mathbf{b}$	Body force
$\mathbf{b}^{-}$	Left Cauchy-Green strain tensor
$\bar{b}_{11}$	Curvature of a boundary
$\mathbf{C}$	Right Cauchy-Green strain tensor
$\mathbf{C}_4$	Constitutive tensor
$\mathbf{C}\,(\xi)$	B-Spline/NURBS curve
$\mathbf{D}$	Rate of deformation tensor
$\mathbf{d}, \mathbf{d}_i$	Nodal displacements
Δt	Time increment
$\mathbf{E}$	Green-Lagrange strain tensor
E	Elastic modulus
$\mathbf{e}^{\mathrm{A}}$	Almansi-Euler strain tensor
$\mathbf{e}, \boldsymbol{\eta}$	Linear and nonlinear components of the Green-Lagrange strain tensor
$\mathbf{e}_3$	Orthogonal unit vector
$\boldsymbol{\varepsilon}$	General strain tensor
$\mathbf{F}$	Deformation gradient
$\mathbf{f}$	General force vector
$\mathbf{g}_i$	Covariant basis vector
g_{N}	Normal gap
$\mathbf{H}$	Subspace matrix for transverse shear locking

H	Linear isotropic hardening coefficient
$\mathbf{I}$	Identity tensor
I	Incompressible deformation subspace
$\mathbf{J}, \boldsymbol{\phi}, \boldsymbol{\psi}$	Jacobian mapping matrices
J	Determinant of the deformation gradient
$\mathbf{K}$	Stiffness matrix
$\mathbf{L}$	Velocity gradient tensor
$L_{k,r}(\zeta)$	Univariate B-Spline basis functions of order r along the ζ-direction
λ, G	Lamé parameters
λ_N	Lagrange multiplier/contact pressure
$M_{j,q}(\eta)$	Univariate B-Spline basis functions of order q along the η-direction
m_i	Multiplicity of knot i
$\mathbf{N}, \mathbf{N}_i$	Shape functions matrix
N_i	Shape function of node i
$N_{i,p}(\xi)$	Univariate B-Spline basis functions of order p along the ξ-direction
n	Increment number
$n_\mathrm{c}, m_\mathrm{c}, l_\mathrm{c}$	Number of control points along ξ- η- and ζ-directions
n_e	Number of elements
n_f	Number of external concentrated loads
$n_\mathrm{m}, n_\mathrm{s}$	Number of control points defining the master and slave surfaces
n_n	Number of nodes of an element
n_ξ, n_η, n_ζ	Nodes along ξ- η- and ζ-directions
ν	Poisson's coefficient
$\boldsymbol{\Omega}$	Rotation rate tensor
$\mathbf{P}$	First Piola-Kirchhoff stress tensor
p, q, r	Polynomial degree of basis functions in ξ-, η- and ζ-directions
$p_\mathrm{m}, p_\mathrm{s}$	Polynomial degree of basis functions of the master and slave surfaces
p_N	Contact pressure
$\mathbf{Q}$	Subspace matrix for volumetric locking
$\mathbf{R}$	NURBS basis functions vector/Orthogonal rotation tensor
$\mathbf{r}_i$	Local orthonormal frame
$R_i^p(\xi)$	Univariate NURBS basis functions
$R_{i,j}^{p,q}(\xi, \eta)$	Bivariate NURBS basis functions
$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta)$	Trivariate NURBS basis functions
ρ	Density
$\mathbf{S}$	Second Piola-Kirchhoff stress tensor
$\mathbf{S}(\xi, \eta)$	B-Spline/NURBS surface

S	Surface
$S_{\mathrm{N}}, S_{\mathrm{D}}, S_{\mathrm{C}}$	Surfaces for Newman, Dirichelet and contact boundary conditions
$\boldsymbol{\sigma}$	Cauchy stress tensor
σ_{y}	Yield stress
σ^{C}	Contact stress
Π	Potential energy
$\mathbf{T}$	Second-order transformation tensor
T	Subspace of transverse shear deformations
$\mathbf{t}$	Prescribed tractions
$\mathbf{U}$	Right stretch tensor
U	Space of all displacement patterns
$\mathbf{u} = (u, v, w)$	Displacement field
$\mathbf{u}_{S_{\mathrm{D}}}$	Prescribed displacements
$\mathbf{V}(\xi, \eta, \zeta)$	B-Spline/NURBS volume
$\mathbf{v}$	Velocity field
V	Volume
$\mathbf{W}$	Spin rate tensor
$W(\xi)$	NURBS weighting function
w_i	NURBS weight
X, Y, Z	Global coordinate system (reference configuration)
x, y, z	Global coordinate system (current configuration)
$x_{\mathrm{s}}, y_{\mathrm{s}}, z_{\mathrm{s}}$	Global coordinates of the contact collocation point
ξ, η, ζ	Natural coordinate system
$\boldsymbol{\Xi}, \mathbf{H}, \mathbf{Z}$	Knot vectors
ξ_i, η_i, ζ_i	Knot values of $\boldsymbol{\Xi}$, $\mathbf{H}$ and $\mathbf{Z}$

List of Abbreviations

ANS	Assumed Natural Strain
BB	Babuška-Brezzi
EAS	Enhanced Assumed Strain
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CPP	Closest Point Projection
DSG	Discrete Strain Gap
FEAP	Finite Element Analysis Program
FEM	Finite Element Method

GPTS	Gauss-Point-to-Surface
HWV	Hu-Wazhizu-Veubeke
ICO	Isogeometric COde
IGA	Isogeometric Analysis
KTS	Knot-to-Surface
LM	Lagrange Multiplier
MITC	Mixed Interpolation of Tensorial Components
MPC	Multiple Point Constraint
NICE	Nodally Integrated Continuous Element
NTS	Node-to-Segment
NURBS	Non-Uniform Rational B-Splines
PTS	Point-to-Segment
PVW	Principle of Virtual Work
RI	Reduced Integration
SRI	Selective Reduced Integration
TL	Total Lagrangian
$\mathrm{u}-\mathrm{p}$	Mixed Displacement-Pressure
UEL	User Element
UELMAT	User Element with access to Abaqus Material Database
UL	Updated Lagrangian

Chapter 1

Introduction

In this chapter, the present research work is introduced. The motivation and main objectives of the Thesis are presented, followed by a brief description of the developed numerical tools. The general outline of the document is described.

The analysis of shell-like structures in the geometric and material nonlinear regimes still represents a challenge in the field of computational mechanics. The development of reliable and computationally effective numerical formulations for this kind of applications is, therefore, an important research topic in computational mechanics.

In the context of the Finite Element Method (FEM), the use of Lagrangian-based formulations to solve structural problems has been the subject of a significant research effort. When thickness values are relatively low, this kind of problems are often modelled using shell finite elements. However, shell finite element formulations present some drawbacks, specially when considering nonlinear regimes. For example, the use of rotational degrees-of-freedom leads to a more complex treatment of large deformations when compared with formulations based only on displacement degrees-of-freedom. As an alternative, solid finite elements can be used, but these are known to lead to solutions that are often polluted by spurious high stiffness values, leading to overly small displacement fields. This non-physical phenomenon is often encountered when modelling straight and curved structures with high length-to-thickness ratios.

In order to circumvent these difficulties, the so called *solid-shell* class of elements was proposed and received a great amount of attention in the last years. Solid-shell formulations combine the advantages of both solid and shell formulations, leading to an approach that relies uniquely in displacement degrees-of-freedom, but suitable for the numerical simulation of thick and thin structures. Since in this kind of formulations only displacement degrees-of--freedom are employed, they can automatically account for 3D constitutive relations, being able to more accurately model through the thickness gradients of stress and strains. It is worth mentioning that solid-shell formulations also present important advantages when considering double-sided contact situations.

Departing from conventional FEM formulations for engineering problems, the recently introduced Isogeometric Analysis (IGA) concept is a numerical method in which the basis functions employed to (exactly) define the geometries involved are also used to determine the unknown fields of the discretised system. This contrasts with FEM, in which the geometry is always approximated. Instead of the standard shape functions based on Lagrangian polynomials usually considered in FEM, IGA can employ as basis functions, for instance, B-Splines, Non-Uniform Rational B-Splines (NURBS) or even T-Splines, the latter developed in recent years by the Computed Aided Design community. This approach allows for a direct connection between the Computer Aided Design (CAD) and the Computer Aided Engineering (CAE) worlds. In the past few years, a significant amount of research effort has been devoted to IGA due to its advantages over classical Finite Element Method. However, recent studies also demonstrate that, although NURBS-based formulations present, in general, a superior performance over standard Lagrangian-based ones, they can still be affected by the same non-physical phenomena appearing in FEM, and leading to an overestimation of some components of the stiffness matrix. This will then result in small (spurious) displacement fields and, in these cases, the solution is said to be *locked*.

The present Thesis is related to the study, development and implementation of numerical models and formulations in the context of Isogeometric Analysis. The main goal of the current research work is then to develop and implement robust tools that can be applied to problems of solid mechanics in both linear and nonlinear regimes. In this context, an efficient NURBS-based solid-shell formulation suitable for the analysis of thin-shell structures is proposed. The methodology employs the Assumed Natural Strain (ANS) technique that has been widely used in the context of FEM to alleviate locking pathologies. In order to be validated in a generality of applications, the formulation is extensively tested using well-known benchmark problems encompassing both linear and nonlinear behaviours. It is also proposed in this work an extension of the Assumed Natural Strain methodology in order to account for a material-based locking pathology which occurs in near incompressible problems. Additionally, an introductory study of contact mechanics problems in the context of Isogeometric Analysis is also presented. Due to the inherent properties of B-Splines/NURBS basis functions (such as high inter-element continuity and superior approximations of the contact stress) the use of NURBS-based formulations to model contact mechanics problems can represent a very attractive alternative to classical Lagrangian-based methodologies.

In order to fulfil the objectives proposed, a set of numerical tools were developed and implemented and an *Isogeometric COde* (ICO) was developed and written using the programming language Fortran 90. The purpose of this code is twofold: (*i*) it serves as an invaluable tool to implement, test and validate new methodologies, and (*ii*) it was written with the aim of becoming a solid foundation for future researchers to build upon. In addition, several tools were developed for the pre- and post-processing steps of a typical engineering analysis. In order to solve problems containing a high number of degrees-of-freedom and/or requiring advanced solution techniques, a set of user subroutines for the commercial software *Abaqus* was also implemented.

The Thesis is composed of 8 chapters. A description of the content of each chapter is given in the following:

- **Chapter 1**: In this chapter, the present research work is introduced. The motivation and main objectives of the Thesis are presented, followed by a brief description of the developed numerical tools. The general outline of the document is described;

- **Chapter 2**: In this chapter, the fundamental concepts behind the Finite Element Method are reviewed. A detailed description about the implementation of classical displacement Lagrangian-based formulations is given. This chapter also serves the purpose of introducing the nomenclature that will be employed throughout the remainder of this work;

- **Chapter 3**: The concept of Isogeometric Analysis is presented and detailed. The first part of the chapter is concerned with the introduction of B-Spline basis functions and the definition of curves, surfaces and solids. This is followed by a description of Non--Uniform Rational B-Splines, as a general case of the B-Splines, and special attention is given to their integration with Finite Element Analysis. Finally, a description of the tools developed throughout this research work is provided;

- **Chapter 4**: A summary of nonlinear continuum mechanics is provided, with focus on the main topics that have been studied and implemented throughout the current work. The theoretical background of the adopted corotational approach is described, along with a detailed description concerning the implementation of numerical models for analysis including geometric nonlinearities, as well as the corresponding developed algorithms;

- **Chapter 5**: In this chapter, the locking phenomena that can pollute numerical analyses based on FEM and well as IGA are described. This is followed by a state-of-the-art review of the main methodologies used to alleviate these non-physical phenomena in the context of both approaches. A special focus is given to the Enhanced Assumed Strain (EAS) and Assumed Natural Strain (ANS) methods and their possible application in Isogeometric Analysis. An innovative extension of the Assumed Natural Strain method is proposed in the context of IGA, leading to the development of high-order NURBS-based solid-shell elements, suitable for the analysis of thin structures. Finally, some insight into volumetric locking in the context of IGA is also provided;

- **Chapter 6**: A brief state-of-the-art review of the main developments in the context of contact mechanics for Isogeometric Analysis is presented. The description of a general two-dimensional frictionless contact problem is given, followed by a detailed description of the Point-to-Segment algorithm where special attention is provided to the main aspects of the implementation procedure;

- **Chapter 7**: The performance of the NURBS-based formulations proposed in Chapter 5 are assessed using a set of well-known benchmark problems in both linear and nonlinear regimes. Additionally, in the context of contact mechanics, the validation of the implemented Point-to-Segment algorithm described in Chapter 6 in the linear elastic regime is performed, also by means of various benchmark problems;

- **Chapter 8**: The main conclusions of the work are presented, along with some suggestions for future developments.

Chapter 2

Formulation of the Finite Element Method for Linear Analysis

In this chapter, the fundamental concepts behind the Finite Element Method are reviewed. A detailed description about the implementation of classical displacement Lagrangian-based formulations is given. This chapter also serves the purpose of introducing the nomenclature that will be employed throughout the remainder of this work.

2.1 The Principle of Virtual Work

Consider the generic three-dimensional body in the global reference system *Oxyz* which is represented in Figure 2.1, involving a stress field $\boldsymbol{\sigma}$ and subject to a body force **b**. In multidimensional elasticity, the equilibrium of the system is given as [Timoshenko 51]

$$\operatorname{div}(\boldsymbol{\sigma}) + \mathbf{b} = 0, \tag{2.1}$$

in the entire volume V, where $\operatorname{div}(\cdot)$ is the divergence operator. Equation 2.1 is known as the strong formulation of the problem. The volume domain is bounded by a surface S, where the two subsets S_{N} and S_{D} can be identified. In the boundary S_{N}, tractions **t** can be prescribed defining the natural (Newman) boundary conditions. Prescribed displacements represented by essential (Dirichlet) boundary conditions can be defined as

$$\mathbf{u} = \mathbf{u}_{S_{\mathrm{D}}}, \tag{2.2}$$

on surface S_{D}. Since S_{D} and S_{N} are two subsets of the boundary surface S, it can be written that $S_{\mathrm{D}} \cup S_{\mathrm{N}} = S$ and $S_{\mathrm{D}} \cap S_{\mathrm{N}} = \emptyset$.

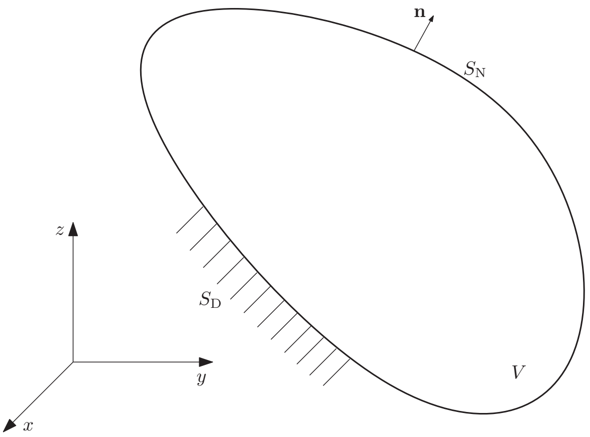


Figure 2.1: General three-dimensional body.

In order to obtain the weak (variational) form of Equation 2.1, it is necessary to multiply the previous equation by an arbitrary test function $\delta\mathbf{u}$ and integrate over the whole domain. This test function is also known as a virtual displacement field and must be consistent with the given boundary conditions. Therefore, Equation 2.1 can now be re-written as

$$\int_V \delta\mathbf{u}\cdot[\mathrm{div}(\boldsymbol{\sigma})+\mathbf{b}]\ \mathrm{d}V=0, \tag{2.3}$$

and, by means of the mathematical identity

$$\mathrm{div}(\delta\mathbf{u}\cdot\boldsymbol{\sigma})=\delta\mathbf{u}\cdot\mathrm{div}(\boldsymbol{\sigma})-\mathrm{grad}(\delta\mathbf{u}):\boldsymbol{\sigma}, \tag{2.4}$$

it is possible to obtain

$$-\int_V \mathrm{grad}(\delta\mathbf{u}):\boldsymbol{\sigma}\ \mathrm{d}V+\int_V \mathrm{div}(\delta\mathbf{u}\cdot\boldsymbol{\sigma})\ \mathrm{d}V+\int_V \delta\mathbf{u}\cdot\mathbf{b}\ \mathrm{d}V=0, \tag{2.5}$$

where $\mathrm{grad}(\cdot)$ is the gradient operator. By applying the divergence theorem to the second term of the previous equation, and taking into account that $\delta\mathbf{u}=0$ in S_D, the equation can now be re-written as

$$-\int_V \mathrm{grad}(\delta\mathbf{u}):\boldsymbol{\sigma}\ \mathrm{d}V+\int_{S_\mathrm{N}} \delta\mathbf{u}\cdot\mathbf{t}\ \mathrm{d}S+\int_V \delta\mathbf{u}\cdot\mathbf{b}\ \mathrm{d}V=0. \tag{2.6}$$

Recalling that, due to symmetry, the strain in the whole volume can be expressed as

$$\boldsymbol{\varepsilon}=\frac{1}{2}\left[\mathrm{grad}(\mathbf{u})+\mathrm{grad}(\mathbf{u})^\mathrm{T}\right]=\mathrm{grad}(\mathbf{u}), \tag{2.7}$$

and the corresponding stress field given by

$$\boldsymbol{\sigma}=\mathbf{C}_4:\boldsymbol{\varepsilon}, \tag{2.8}$$

where $\mathbf{C}_4$ is the fourth-order constitutive tensor, then Equation 2.6 can now be re-written, after rearranging, as

$$\int_V \boldsymbol{\sigma}:\delta\boldsymbol{\varepsilon}\ \mathrm{d}V=\int_V \delta\mathbf{u}\cdot\mathbf{b}\ \mathrm{d}V+\int_{S_\mathrm{N}} \delta\mathbf{u}\cdot\mathbf{t}\ \mathrm{d}S, \tag{2.9}$$

which is the weak form of Equation 2.1, also known as the Principle of Virtual Work (PVW). The PVW is the principle behind classical displacement-based Finite Element models.

Take now into account the body with volume V depicted in Figure 2.2, before (solid line) and after (dashed line) the application of infinitesimal virtual displacements $\delta\mathbf{u}$ and a set of forces $\mathbf{f}_i$ acting upon it. In the picture, it is assumed that the virtual displacements are small enough to maintain the forces $\mathbf{f}_i$ unaltered. The virtual strains $\delta\boldsymbol{\varepsilon}$ coming from the compatible virtual displacements can be then used to determine the internal virtual work as

$$\delta\Pi^{\text{int}} = \int_V \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} \; \mathrm{d}V. \tag{2.10}$$

Additionally, the total external virtual work can be expressed as

$$\delta\Pi^{\text{ext}} = \int_V \mathbf{b} \cdot \delta\mathbf{u} \; \mathrm{d}V + \int_{S_\mathrm{N}} \mathbf{t} \cdot \delta\mathbf{u} \; \mathrm{d}S. \tag{2.11}$$

The postulate of the PVW states that [Bathe 96], in the state of the equilibrium of the body, the total internal virtual work is equal to the total external virtual work, which is the condition given in Equation 2.9. The effect of concentrated forces $\mathbf{f}_i$ acting upon the body can also be included in the PVW as

$$\int_V \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon} \; \mathrm{d}V = \int_V \mathbf{b} \cdot \delta\mathbf{u} \; \mathrm{d}V + \int_{S_\mathrm{N}} \mathbf{t} \cdot \delta\mathbf{u} \; \mathrm{d}S + \sum_i^{n_\mathrm{f}} \delta\mathbf{u} \cdot \mathbf{f}_i, \tag{2.12}$$

where n_f is the total number of applied concentrated forces.

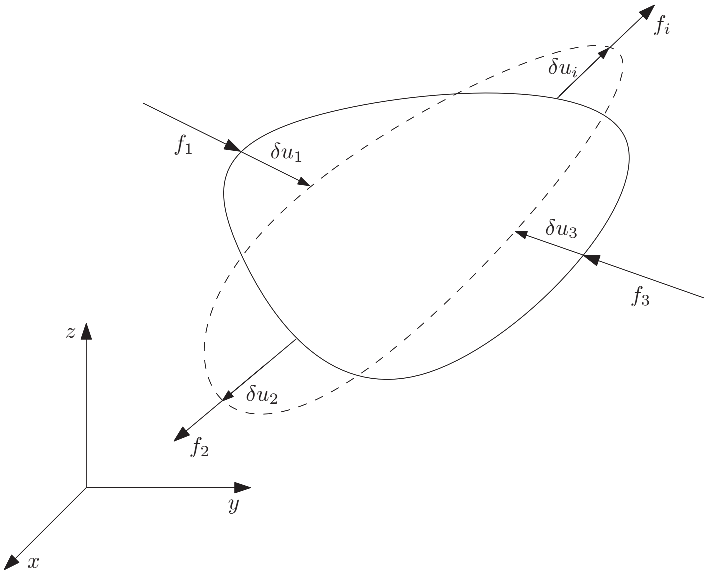


Figure 2.2: Body with volume V in equilibrium before and after the application of virtual displacements and forces.

2.2 Displacement-Based Finite Element Formulations

In order to obtain a Finite Element solution, a given continuous body must be subdivided in a (finite) number of elements, which are resolved individually and subsequently assembled, in order to obtain a global solution of the problem. On each of these elements, connected by nodes, the governing equations can be formulated using variational methods, where, in the case of displacement-based elements, the variational principle used is the PVW. By making use of the upper index $(\cdot)^{\mathrm{h}}$ to denote a Finite Element approximation, the displacement field ($\mathbf{u}$) on each individual element can be approximately interpolated as

$$\mathbf{u} \approx \mathbf{u}^{\mathrm{h}} = \mathbf{N}\mathbf{d}, \tag{2.13}$$

where $\mathbf{d}_i = [u_i \; v_i \; w_i]^{\mathrm{T}} \; (i = 1, ..., n_{\mathrm{n}})$ represents the nodal displacement vector for a given element with n_{n} nodes and $\mathbf{N}$ is a matrix that contains the interpolating shape functions as

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & ... & \mathbf{N}_{n_{\mathrm{n}}} \end{bmatrix}, \tag{2.14}$$

in which

$$\mathbf{N}_i = \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix}, \; i = 1, \ldots, n_{\mathrm{n}}. \tag{2.15}$$

In a general three-dimensional continuum analysis, the strain field can be expressed as

$$\boldsymbol{\varepsilon} = \left\{ \begin{array}{c} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \end{array} \right\} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \left\{ \begin{array}{c} u \\ v \\ w \end{array} \right\}, \tag{2.16}$$

or in a more compact way as

$$\boldsymbol{\varepsilon} = \boldsymbol{\nabla}_{\mathrm{s}}\mathbf{u}. \tag{2.17}$$

By combining Equations 2.13 and 2.16, the strain field can then be obtained in a discretised form as

$$\boldsymbol{\varepsilon} = \boldsymbol{\nabla}_{\mathrm{s}}\mathbf{u} \approx \boldsymbol{\nabla}_{\mathrm{s}}(\mathbf{N}\mathbf{d}) = \mathbf{B}\mathbf{d}, \tag{2.18}$$

where $\mathbf{B}$ is known as the strain-displacement operator which contains the derivatives of the element shape functions as

$$\mathbf{B} = [\mathbf{B}_1 \; \mathbf{B}_2 \; \ldots \; \mathbf{B}_{n_{\mathrm{n}}}], \tag{2.19}$$

with

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \end{bmatrix}. \tag{2.20}$$

By employing the constitutive tensor $\mathbf{C}_4$ it is then possible to obtain the elemental stress field as

$$\boldsymbol{\sigma} = \mathbf{C}_4 : \boldsymbol{\varepsilon}, \tag{2.21}$$

where, in the case of an isotropic linear elastic material, the matrix form of the $\mathbf{C}_4$ tensor can be defined as

$$\mathbf{C}_4 = \frac{E\,(1-\nu)}{(1+\nu)\,(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}, \tag{2.22}$$

in which E is the elastic modulus and ν the Poisson's coefficient.

Introducing the Equations 2.18 and 2.21 into Equation 2.12, for each finite element it is possible to state that

$$\int_{V^e} (\delta\mathbf{d})^{\mathrm{T}}\,\mathbf{B}^{\mathrm{T}}\mathbf{C}_4\mathbf{B}\mathbf{d}\;\mathrm{d}V^e - \int_{V^e} (\delta\mathbf{d})^{\mathrm{T}}\,\mathbf{N}^{\mathrm{T}}\mathbf{b}^{\mathrm{T}}\;\mathrm{d}V^e = (\delta\mathbf{d})^{\mathrm{T}}\,\mathbf{f}^e. \tag{2.23}$$

Taking into account that the virtual nodal displacements $\delta\mathbf{d}$ are constant and always non-zero and the nodal displacements $\mathbf{d}$ are also constant, Equation 2.23 can now be expressed as

$$\left(\int_{V^e} \mathbf{B}^{\mathrm{T}}\mathbf{C}_4\mathbf{B}\;\mathrm{d}V^e\right)\mathbf{d} - \int_{V^e} \mathbf{N}^{\mathrm{T}}\mathbf{b}^{\mathrm{T}}\;\mathrm{d}V^e = \mathbf{f}^e. \tag{2.24}$$

From this equation, the elemental stiffness matrix can be defined as

$$\mathbf{K}^e = \int_{V^e} \mathbf{B}^{\mathrm{T}}\mathbf{C}_4\mathbf{B}\;\mathrm{d}V^e, \tag{2.25}$$

as well as the elemental body force vector

$$\mathbf{b}^e = \int_{V^e} \mathbf{N}^{\mathrm{T}}\mathbf{b}^{\mathrm{T}}\;\mathrm{d}V^e. \tag{2.26}$$

The elements' stiffness matrices must then be assembled (element by element) in order to obtain the global stiffness matrix $\mathbf{K}$, leading to the global system of equations defined as

$$\mathbf{K}\mathbf{d} = \mathbf{f}, \tag{2.27}$$

which must be solved to obtain the unknown nodal displacements.

2.3 The Classical Displacement-Based Hexahedral Element

In the field of computational mechanics, very often a problem must be modelled using a three-dimensional geometry. In the current section, the classical displacement-based 3D hexahedral (brick) Finite Element formulation is described is detail, in its simplest trilinear form.

2.3.1 Shape Functions

When using finite elements, it is effective to employ the isoparametric concept. This concept states that the interpolating functions adopted in the approximation of the degrees-of-freedom are also used in the description of the geometry. To that end, a normalized natural coordinate system $O\xi\eta\zeta$ is defined. All points within a finite element are contained in the domain $[-1,+1]\times[-1,+1]\times[-1,+1]$, and the ξ, η and ζ axis are assumed to have their origin at the centre of the element, passing through the centre of opposite surfaces. The use of a natural coordinate system is convenient for constructing the shape functions, as well as to perform the numerical integrations by a Gauss-Legendre quadrature scheme. The representation of the global and natural reference systems for a general linear brick element can be seen in Figure 2.3.

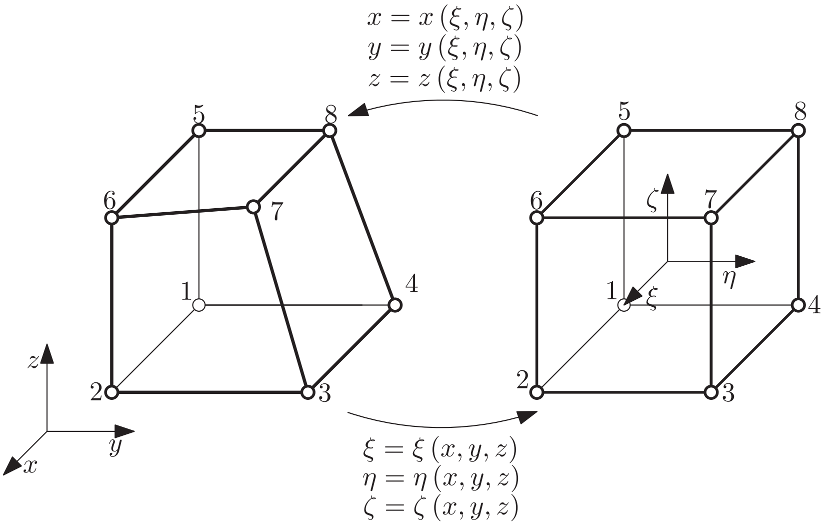


Figure 2.3: Schematic representation of a linear hexahedral finite element in the global (left) and natural (right) reference systems.

The first step in developing an isoparametric hexahedral finite element is to define the shape functions that will be used for the discretisation. The shape function for a given node i, with $i = 1,...,n_n$, can be obtained by the Lagrange interpolation functions as

$$N_i(\xi,\eta,\zeta) = N_i(\xi)\,N_i(\eta)\,N_i(\zeta), \tag{2.28}$$

where

$$N_i(\xi) = \prod_{j=1, j\neq i}^{n_\xi} \frac{\xi - \xi_j}{\xi_i - \xi_j}, \tag{2.29}$$

$$N_i(\eta) = \prod_{j=1, j\neq i}^{n_\eta} \frac{\eta - \eta_j}{\eta_i - \eta_j}, \tag{2.30}$$

and

$$N_i(\zeta) = \prod_{j=1, j\neq i}^{n_\zeta} \frac{\zeta - \zeta_j}{\zeta_i - \zeta_j}, \tag{2.31}$$

in which n_ξ, n_η and n_ζ represent the number of nodes along the ξ-, η- and ζ-directions, respectively. For the case of the trilinear brick element (eight nodes) depicted in Figure 2.3, the shape functions defined in the natural domain are given as

$$N_i(\xi, \eta, \zeta) = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i), \tag{2.32}$$

where ξ_i, η_i and ζ_i are the components of the vectors $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$, respectively, defined as

$$\boldsymbol{\xi} = \begin{Bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{Bmatrix}, \ \boldsymbol{\eta} = \begin{Bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{Bmatrix}, \text{ and } \boldsymbol{\zeta} = \begin{Bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix}. \tag{2.33}$$

The shape functions derived in the previous equations are referred to the natural coordinate system $O\xi\eta\zeta$. In order to compute the stress and strain fields, it is required nevertheless to write the interpolatory functions in the global coordinate system $Oxyz$. The mapping between the global and natural coordinate systems can be obtained by the relation between the shape functions' derivatives in the global and natural spaces, obtained by the chain rule as

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix}, \tag{2.34}$$

where $\mathbf{J}^{-1}$ is the inverse of the Jacobian matrix, defined as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}. \tag{2.35}$$

The Jacobian matrix can be obtained by making use of the derivatives of the shape functions in the natural reference system and the coordinates of each node $\mathbf{x}_i = (x_i, y_i, z_i)$, as

$$\mathbf{J} = \sum_{i=1}^{n_\mathrm{n}} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} x_i & \frac{\partial N_i}{\partial \xi} y_i & \frac{\partial N_i}{\partial \xi} z_i \\ \frac{\partial N_i}{\partial \eta} x_i & \frac{\partial N_i}{\partial \eta} y_i & \frac{\partial N_i}{\partial \eta} z_i \\ \frac{\partial N_i}{\partial \zeta} x_i & \frac{\partial N_i}{\partial \zeta} y_i & \frac{\partial N_i}{\partial \zeta} z_i \end{bmatrix}. \tag{2.36}$$

Using the Jacobian operator, it is then possible to obtain the derivatives of the shape functions with respect to the global coordinates, which can be promptly used to build the strain-displacement operator **B** using Equations 2.19 and 2.20.

2.3.2 Elemental Stiffness Matrix and Load Vector

From the developments in Section 2.2, the elemental stiffness matrix can be calculated as

$$\mathbf{K}^e = \int_{V^e} \mathbf{B}^\mathrm{T} \mathbf{C}_4 \mathbf{B} \mathrm{d}V^e,$$

or, alternatively, in the natural domain as

$$\mathbf{K}^e = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{B}^\mathrm{T} \mathbf{C}_4 \mathbf{B} |\mathbf{J}| \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta, \tag{2.37}$$

where $|\cdot|$ is the determinant operator. An approximation of the stiffness matrix can be obtained by numerical integration using the Gauss-Legendre quadrature as

$$\mathbf{K}^e \approx \sum_{i=1}^{n_\mathrm{r}} \sum_{j=1}^{n_\mathrm{s}} \sum_{j=1}^{n_\mathrm{t}} \left(\mathbf{B}^\mathrm{T} \mathbf{C}_4 \mathbf{B} |\mathbf{J}| \right)_{r,s,t} w_\mathrm{r} w_\mathrm{s} w_\mathrm{t}, \tag{2.38}$$

where r, s and t are the number of integration points along ξ, η and ζ, respectively, and w_r, w_s and w_t the corresponding weights.

Similarly, the contribution of the volumetric load **b** to the load vector is given as

$$\mathbf{b}^e = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{N}_i^\mathrm{T} \mathbf{b} |\mathbf{J}| \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta.$$

When considering traction loads, a different approach must be taken. Since this type of load is applied to a face of the element, the normal vector of the face must be first determined. To do so, the tangential directions of the natural axis ξ, η and ζ are required, as

$$\mathbf{g}_1 = \begin{Bmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \\ \frac{\partial z}{\partial \xi} \end{Bmatrix}, \ \mathbf{g}_2 = \begin{Bmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \\ \frac{\partial z}{\partial \eta} \end{Bmatrix}, \text{ and } \mathbf{g}_3 = \begin{Bmatrix} \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \zeta} \end{Bmatrix}. \tag{2.39}$$

Comparing the tangential vectors with Equation 2.36, it can be seen that they correspond to the columns of the transposed Jacobian operator. The normal directions of the faces can now be calculated as

$$\mathbf{n}_1 = \frac{\mathbf{g}_2 \times \mathbf{g}_3}{||\mathbf{v}_2 \times \mathbf{g}_3||}, \tag{2.40}$$

$$\mathbf{n}_2 = \frac{\mathbf{g}_3 \times \mathbf{g}_1}{||\mathbf{g}_3 \times \mathbf{g}_1||}, \tag{2.41}$$

$$\mathbf{n}_3 = \mathbf{n}_1 \times \mathbf{n}_2. \tag{2.42}$$

The applied traction load $\mathbf{t}$ can then be determined as

$$\mathbf{t} = t\mathbf{n}, \tag{2.43}$$

where $\mathbf{n}$ is the normal to the face where the traction with magnitude t is applied. Afterwards, the equivalent nodal forces can be calculated as

$$\mathbf{f}^e_{t,i} = \int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N}_i^{\mathrm{T}}(\pm 1, \eta, \zeta)\, t_1\mathbf{n}_1 d\eta d\zeta, \text{ for direction } O\xi, \tag{2.44}$$

$$\mathbf{f}^e_{t,i} = \int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N}_i^{\mathrm{T}}(\xi, \pm 1, \zeta)\, t_2\mathbf{n}_2 d\xi d\zeta, \text{ for direction } O\eta, \tag{2.45}$$

$$\mathbf{f}^e_{t,i} = \int_{-1}^{+1}\int_{-1}^{+1} \mathbf{N}_i^{\mathrm{T}}(\xi, \eta, \pm 1)\, t_3\mathbf{n}_3 d\xi d\eta, \text{ for direction } O\zeta. \tag{2.46}$$

In Box 2.1, the general algorithm for the implementation of a Lagrangian displacement-based 3D isoparametric finite element is presented.

Box 2.1: Algorithm for the implementation of a general Lagrangian displacement-based 3D isoparametric element.

1. Initialise elemental stiffness $\mathbf{K}^e$ matrix and load vector $\mathbf{f}^e$

2. **DO** integration points' cycle

   (a) Compute shape functions N_i in the natural frame (Equations 2.32 and 2.33) and its derivatives $\frac{\partial N_i}{\partial \xi}$, $\frac{\partial N_i}{\partial \eta}$ and $\frac{\partial N_i}{\partial \zeta}$

   (b) Calculate the Jacobian matrix $\mathbf{J}$ as

   $$\mathbf{J} = \sum_{i=1}^{n_\mathrm{n}} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} x_i & \frac{\partial N_i}{\partial \xi} y_i & \frac{\partial N_i}{\partial \xi} z_i \\ \frac{\partial N_i}{\partial \eta} x_i & \frac{\partial N_i}{\partial \eta} y_i & \frac{\partial N_i}{\partial \eta} z_i \\ \frac{\partial N_i}{\partial \zeta} x_i & \frac{\partial N_i}{\partial \zeta} y_i & \frac{\partial N_i}{\partial \zeta} z_i \end{bmatrix}$$

   and its determinant $|\mathbf{J}|$ and inverse $\mathbf{J}^{-1}$

   (c) Map the derivatives of the shape functions into the global space

   $$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix}$$

   and assemble the strain-displacement operator $\mathbf{B}$

   (d) Perform the numerical integration of the stiffness matrix for the current integration point and add it to $\mathbf{K}^e$

   $$\mathbf{K}^e = \mathbf{K}^e + \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{B}^\mathrm{T} \mathbf{C}_4 \mathbf{B} |\mathbf{J}| \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta$$

   (e) If the nodal displacements are available, compute the strain and stress fields

   $$\boldsymbol{\varepsilon} = \mathbf{Bd}$$

   $$\boldsymbol{\sigma} = \mathbf{C}_4 \boldsymbol{\varepsilon}$$

3. **END DO**

Chapter 3

B-Splines, NURBS and Isogeometric Analysis

The concept of Isogeometric Analysis is presented and detailed. The first part of the chapter is concerned with the introduction of B-Spline basis functions and the definition of curves, surfaces and solids. This is followed by a description of Non-Uniform Rational B-Splines, as a general case of the B-Splines, and special attention is given to their integration with Finite Element Analysis. Finally, a description of the tools developed throughout this research work is provided.

In the field of Computer Aided Design (CAD), the use of Non-Uniform Rational B--Splines (NURBS) is very popular. This is due to the fact that NURBS are very flexible and accurate, allowing the exact representation of conic curves and surfaces, as well as free-form entities. As a result, NURBS are the standard tools for geometric design and are used in many graphic formats, such as IGES and STEP. However, during the pre-processing stage of an analysis based on the Finite Element Method (FEM), the geometry must be discretised into elements, inevitably leading to a change in the geometry, particularly when considering curved structures discretised with low-order finite elements. This issue is the same even if higher-order finite elements are chosen and, therefore, in a general sense, classical Finite Element discretisations cannot exactly represent the geometry of the problem.

The concept of Isogeometric Analysis (IGA) was firstly introduced by Hughes *et al.* [Hughes 05]. In IGA, B-Spline and NURBS basis functions primarily used to describe the geometry are directly employed in the computation of the unknown fields. As a consequence, it is then possible to perform a numerical simulation in a geometry that is exactly represented, rather than in an approximate fashion as in the FEM. When compared to standard Lagrange elements, NURBS-based formulations have shown to present a better accuracy in structural applications when considering coarse meshes due to their higher-order inter-element continuity [Echter 13].

More recently, and in order to overcome some restrictions of the NURBS basis, T-splines have been used in the context of IGA. T-splines, which represent a generalisation of NURBS, are specially attractive due to their ability to perform local refinement and can generate models with complex geometry suitable for numerical analysis [Sederberg 03, Bazilevs 10, Dörfel 10, Scott 11]. The use of T-splines for Isogeometric Analysis is not considered in the current work.

3.1 B-Splines

In computational geometry, in order to create a curve, surface or volume, it is necessary to have a correct mathematical description of such entity. In order to create free-form curves, the Bézier curves were developed, which are a form of parametric functions that employ Bernstein polynomials and a set of control points to define the desired curve. However, the use of the Bernstein basis leads to limitations in the flexibility of the resulting curve with, for example, high-order curves leading to numerical instabilities. In addition, due to the global nature of the Bernstein basis, it is not possible to have a local control within the curve, which turns to an inability to reproduce local changes [Rogers 01].

In order to overcome these limitations, B-Spline basis (which contains the Bernstein basis as a special case) was introduced by Schoenberg [Schoenberg 46]. This basis presents a non-global behaviour, meaning that each control point only affects the shape of the curve in the range in which the associated basis function is non-zero. The description of B-Spline curves and surfaces is given in more detail in the following.

Consider the representation of a B-Spline curve given by

$$\mathbf{C}(\xi) = \sum_{i=1}^{n_\mathrm{c}} N_{i,p}(\xi)\mathbf{B}_i, \tag{3.1}$$

where $\mathbf{B}_i$ (with $i = 1, 2, ..., n_\mathrm{c}$) represents the coordinates of *control point* i and $N_{i,p}$ are piecewise polynomial functions, known as B-Splines basis functions of order p. As a particular case, piecewise linear ($p = 1$) interpolations of the control points lead to the so-called *control polygon*. Although there are different ways of defining the above basis functions, the Cox-de Boor recursion formula (described in the following) is usually employed since it is the most useful for computer implementations.

Let $\boldsymbol{\Xi} = \left[\xi_1, \xi_2, ..., \xi_{n_c+p+1}\right]$ be a non-decreasing sequence of real numbers known as *knot vector*, where ξ_i is the i^{th} *knot*. The interval defined by two subsequent knots is then known as a *knot span*, and therefore the knot vector divides the parameter space into knot spans. A given knot is said to have a *multiplicity* of m if it is repeated m times inside the knot vector, while a knot vector is considered to be *open* if the first and last knots have multiplicity $m = p+1$. An interesting property of open knot vectors is that their basis functions are interpolatory at the ends of the parametric space. Finally, a knot vector is considered to be *uniform* if, in the parameter space, the knots are equally spaced and *non-uniform* otherwise.

3.1.1 Basis Functions

Using the Cox-de Boor recursion formula, the i^{th} B-Spline basis function can be defined as

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{3.2}$$

for a polynomial function of order zero, and

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi), \tag{3.3}$$

otherwise (*i.e.* for $p \geq 1$). The convention $\frac{0}{0} = 0$ is adopted herein.

These basis functions have some important properties. First of all, they are non-negative over the entire domain and constitute a partition of the unity, *i.e.*,

$$\sum_{i=1}^{n_c} N_{i,p}(\xi) = 1, \; \forall \, \xi. \tag{3.4}$$

Each p^{th}-order basis function has $p - m_i$ continuous derivatives across ξ_i, where m_i is the multiplicity of knot ξ_i, and the support of a p^{th}-order basis function is $p+1$ knot spans. When the multiplicity of a given knot is equal to the order p, then the basis functions are interpolatory at that knot. Accordingly, when the multiplicity is $p+1$, the basis becomes discontinuous.

In Figure 3.1-(top) a quadratic basis function for an open, uniform knot vector $\boldsymbol{\Xi} = [0,0,0,1/3,2/3,1,1,1]$ is depicted. At both ends of the interval, the multiplicity is $p+1 = 3$, which means that only at these points the basis are interpolatory and discontinuous. Elsewhere, the functions are C^1-continuous. Consider now that a new knot $\xi = 2/3$ is inserted into the knot vector, as shown in Figure 3.1-(bottom). The knot vector will now be considered as non-uniform and the basis will be interpolatory at $\xi = 2/3$, since in this point the multiplicity is $m = p = 2$. Also, it can be seen that the continuity of the basis has now been decreased, at the repeated knot, to C^0.

Another interesting property of B-Spline curves is that they lie within the convex hull of its control polygon and exhibit a variation diminishing property, guaranteeing that the curve will not oscillate about any straight line more often than its control polygon does. Moreover,

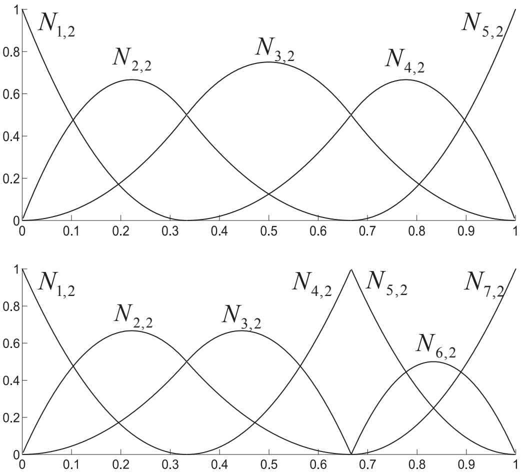


Figure 3.1: Basis functions for (top) open, uniform knot vector $\Xi = [0,0,0,1/3,2/3,1,1,1]$ and (bottom) open, non-uniform knot vector $\Xi = [0,0,0,1/3,2/3,2/3,1,1,1]$.

any affine transformation applied to the curve can be directly applied to the control points. This property is essential for satisfying the patch tests [Cottrell 09].

Since the basis functions are recursively obtained using the Cox-de Boor formula, their first derivatives can be represented in terms of lower order basis, as

$$\frac{\partial N_{i,p}(\xi)}{\partial \xi} = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) . \tag{3.5}$$

3.1.2 B-Spline Surfaces

A tensor product B-Spline surface can be defined as

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n_c} \sum_{j=1}^{m_c} N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{B}_{i,j}, \tag{3.6}$$

where $\mathbf{B}_{i,j}$ is the position of the control points in the (x,y) space, defining the so-called *control net*. In the previous equation, $N_{i,p}(\xi)$ and $M_{j,q}(\eta)$ are the univariate B-Spline basis functions of order p and q, corresponding to the knot vectors $\Xi = \left[\xi_1, \xi_2, ..., \xi_{n_c+p+1}\right]$ and $\mathbf{H} = \left[\eta_1, \eta_2, ..., \eta_{m_c+q+1}\right]$, respectively. The properties of the B-Spline curves follow the corresponding properties defined for the univariate basis functions described in the previous section, as a result of its tensor product nature.

In the following, a simple example will be employed to introduce important concepts related to Isogeometric Analysis. In addition, the example will also serve the purpose of demonstrating some of the differences between B-Splines/NURBS-based and Lagrangian--based formulations. Consider a B-Spline surface defined by two uniform, open knot vectors defined by $\Xi = [0,0,0,1/2,1,1,1]$ and $\mathbf{H} = [0,0,1,1]$, with control points forming the control net given in Figure 3.2. Also, in the same figure the *mesh* of the structure in the physical space can be seen. In the present context, it is worth mentioning that the concept of mesh refers to the non-zero knot spans, defined by the corresponding intervals of each knot vector. Accordingly, the control points can be interpreted as the IGA equivalent to nodes in FEM. However, due to the nature of the B-Splines basis functions, the control points are only interpolatory in the corners, where the multiplicity of the univariate basis functions are $p+1$ and $q+1$. Another key difference is that the elements defined using the B-Spline basis are able to exactly describe the geometry (which can be seen in Figure 3.2), as opposed to Lagrangian-based elements in which the geometry is only approximated.

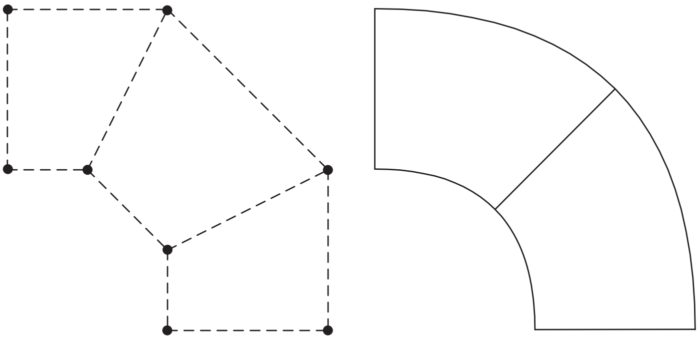


Figure 3.2: Control net (left) and the mesh composed of two elements (right) for a B-Spline surface.

Consider now the concept (exclusive to IGA) of *index space* which can be interpreted as a space in which the axis are defined by all the knots of the knot vector, independently of their value. In a two-dimensional parametric case (*i.e.* in a surface) this leads to a grid as shown in Figure 3.3 for the current example. Each non-zero knot span in a knot vector will then define one *element* along a coordinate direction. Analysing the given knot vectors, it can be seen that $\boldsymbol{\Xi}$ contains two non-zero knot spans while $\mathbf{H}$ contains only one, leading to a total of $2 \times 1 = 2$ elements. Starting from the index space, it is now possible to define the *parameter space* which contains only the non-zero knot spans (or elements). This set of elements is known as a *patch*. The parameter space is also depicted in Figure 3.3, along with the univariate basis functions along the ξ and η directions. In a B-Spline surface, the support of a given bivariate basis function $N_{i,j;p,q}(\xi,\eta)$, is $\left[\xi_i, \xi_{i+p+1}\right] \times \left[\eta_j, \eta_{j+q+1}\right]$. In practical terms, this means that a given basis function will affect a set of knot spans and, consequently, it is possible to obtain high-order inter-element continuity. This differs from standard Lagrangian-based formulation, in which the shape functions are only C^0-continuous between elements. In the example given, the basis will have C^1-continuity at the knot ξ_4, *i.e.*, there exists C^1-continuity between the two elements of the mesh.

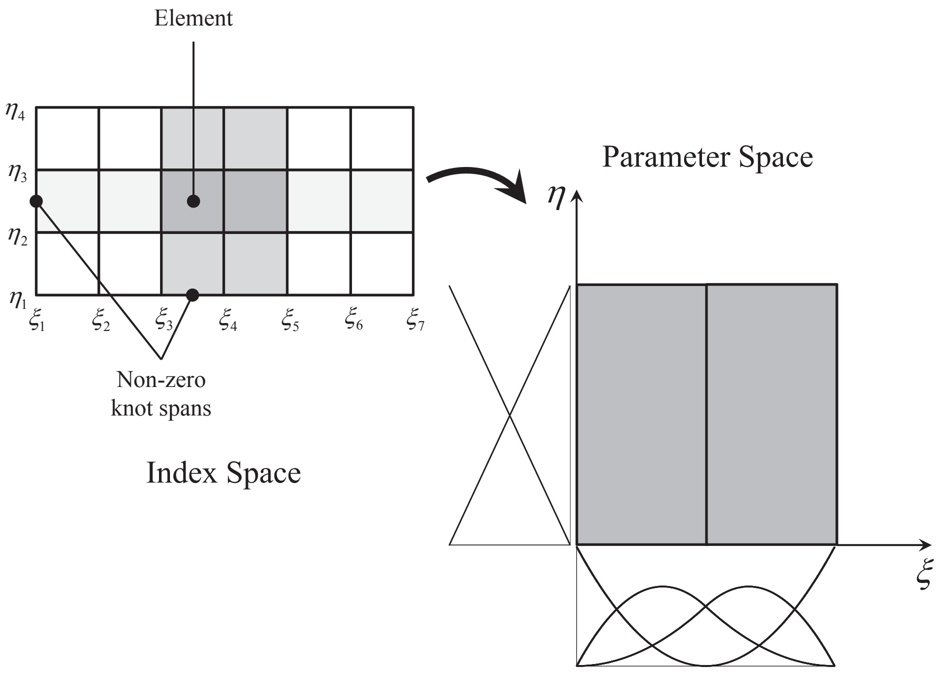


Figure 3.3: B-Spline surface represented in the index and parameter spaces, along with the corresponding basis functions.

3.1.3 B-Spline Solids

Analogously to B-Spline surfaces, it is possible to define a tensor product B-Spline solid. Given a *control lattice* $\mathbf{B}_{i,j,k}$ (the three-dimensional equivalent of a control net) and knot vectors $\mathbf{\Xi} = \left[\xi_1, \xi_2, ..., \xi_{n_c+p+1}\right]$, $\mathbf{H} = \left[\eta_1, \eta_2, ..., \eta_{m_c+q+1}\right]$ and $\mathbf{Z} = [\zeta_1, \zeta_2, ..., \zeta_{k_c+r+1}]$, a B-Spline solid can be expressed as

$$\mathbf{V}(\xi,\eta,\zeta) = \sum_{i=1}^{n_c}\sum_{j=1}^{m_c}\sum_{k=1}^{l_c} N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)\mathbf{B}_{i,j,k}. \tag{3.7}$$

The properties of B-Spline volumes can be obtained from generalizations of the properties of B-Spline curves and surfaces [Piegl 97, Rogers 01, Hughes 05].

3.1.4 Refinement

B-Spline basis can be enriched without changing the studied geometry and its parametrisation, which is an interesting feature when compared to conventional FEM. In CAD, the refinement can be typically performed by the so-called *knot insertion* and *order elevation* techniques. These two methods are closely related to the concepts of h- and p- refinements, respectively, in traditional Finite Element analysis. However, the use of B-Spline basis allows for a new type of refinement known as *k-refinement*. These refinement techniques will be detailed in the following. Efficient algorithms for knot insertion and order elevation procedures, among many others, can be found in [Piegl 97].

Knot Insertion

The knot insertion procedure consists in enriching the basis functions by including additional knot values into the knot vector. Considering, for the sake of simplicity a curve, when using this procedure the curve is not changed, neither geometrically nor parametrically. Taking into account a given knot vector $\boldsymbol{\Xi} = \left[\xi_1, \xi_2, ..., \xi_{n_\mathrm{c}+p+1}\right]$, with control points $\mathbf{B}_i$, and inserting a single knot into it will lead to an extended (refined) knot vector $\boldsymbol{\Xi}^* = \left[\xi_1^* = \xi_1, \xi_2^*, ..., \xi_{n_\mathrm{c}+p+2}^* = \xi_{n_\mathrm{c}+p+1}\right]$. The representation of the B-Spline curve on $\boldsymbol{\Xi}^*$ can be expressed as

$$\mathbf{C}(\xi) = \sum_{i=1}^{n_\mathrm{c}+1} N_{i,p}^* \mathbf{B}_i^*, \tag{3.8}$$

in which $N_{i,p}^*$ is the enriched basis function. Considering that $\xi^* \in [\xi_k, \xi_{k+1}]$, the new control points $\mathbf{B}_i^*$ can be obtained from a linear combination of $\mathbf{B}_i$ as

$$\mathbf{B}_i^* = \alpha_i \mathbf{B}_i + (1 - \alpha_i)\mathbf{B}_{i-1}, \tag{3.9}$$

where

$$\alpha_i = \begin{cases} 1 & i \leq i \leq k-p \\ \frac{\tilde{\xi} - \xi_i}{\xi_{i+p} - \xi_i} & k-p+1 \leq i \leq k \\ 0 & k+1 \leq i \leq n_c + p + 2 \end{cases} . \tag{3.10}$$

Inserting knot values that are already present in the original knot vector will increase their multiplicity and, consequently, the continuity of the basis will be decreased. An example of knot insertion can be seen in Figure 3.4 for an initial knot vector $\boldsymbol{\Xi} = [0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1]$. It can be seen that, after inserting the knots $\xi^* = 0.325$ and $\xi^* = 0.75$, the obtained curve is geometrically and parametrically identical to the original one. It can also be seen that at knot $\xi^* = 0.75$ the basis is now C^0, since the multiplicity at this location was increased. Since the knot $\xi^* = 0.75$ was already present in the original knot vector, a new element was not generated. The process of knot insertion can then be compared with the standard h-refinement in FEM, in which a given mesh is divided into smaller elements.

Order Elevation

The order elevation procedure consists in raising the polynomial order of the basis functions without changing the geometry and parametrisation of the original curve. In this process, the multiplicity of each knot is increased, but no new knots are added. The order elevation procedure can be seen as an extraction of Bézier segments from the curve by replicating

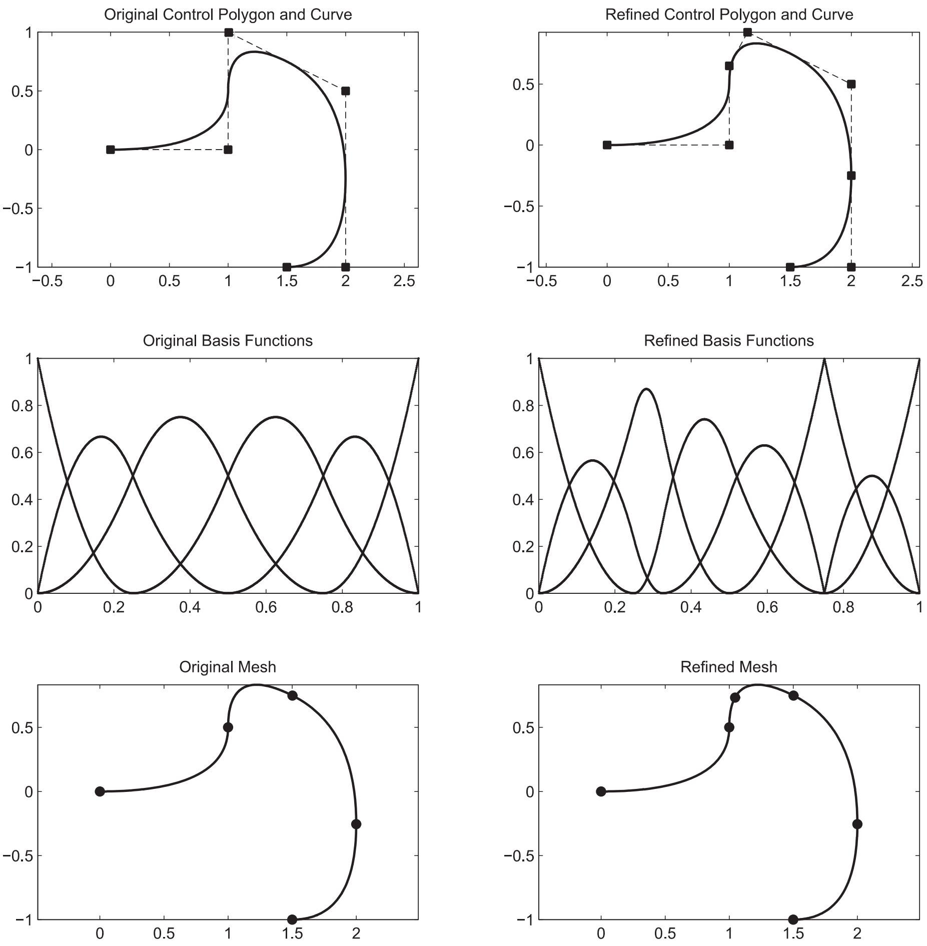


Figure 3.4: Knot insertion example: (left) original curve and (right) the curve after insertion of additional knots.

existing knots, order elevating this segment and, finally, removing unnecessary knots in order to obtain a final B-Spline curve of higher order. An advantage of this method is that the differentiability of the curve at the knots is not reduced, as in the case of the knot insertion procedure.

The mathematical details of the order elevation process are complex and will not be reproduced for the sake of simplicity. With this approach, when elevating the order of a B-Spline curve, the new curve must remain identical to the original. Thus, the order elevation of a B-Spline curve from order p to $p+1$ can be written as

$$\mathbf{C}(\xi) = \sum_{i=1}^{p+1} \mathbf{B}_i N_{i,p}(\xi) = \sum_{i=1}^{p+2} \mathbf{B}_i^* N_{i,p+1}^*(\xi), \tag{3.11}$$

where $\mathbf{B}_i^*$ are the control points defining the new (order elevated) curve. The original knot vector

$$\boldsymbol{\Xi} = [\underbrace{0,\ \ldots,\ 0}_{p+1},\ \underbrace{\xi_1,\ \ldots,\ \xi_1}_{m_1},\ \ldots,\ \underbrace{\xi_s,\ \ldots,\ \xi_s}_{m_s},\ \ldots,\ \underbrace{\xi_{n_c+p+1},\ \ldots,\ \xi_{n_c+p+1}}_{p+1}], \tag{3.12}$$

will now take the form

$$\boldsymbol{\Xi}^{*} = [\underbrace{0,\ ...,\ 0}_{p+2},\ \underbrace{\xi_1,\ ...,\ \xi_1}_{m_1+1},\ ...,\ \underbrace{\xi_s,\ ...,\ \xi_s}_{m_s+1},\ ...,\ \underbrace{\xi_{n_c+p+1},\ ...,\ \xi_{n_c+p+1}}_{p+2}], \tag{3.13}$$

where m_i represents the multiplicity of any of the s internal knots in the original basis. It can be seen from the previous equations that, when a B-Spline curve is order elevated, the curve at a knot of multiplicity m_i remains C^{p-m_i} continuous, *i.e.*, both the original and the order elevated curves have the same continuity at that knot.

A simple example demonstrating the order elevation procedure can be seen in Figure 3.5. The original curve of order $p = 2$ with a knot vector

$$\boldsymbol{\Xi} = [0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1],$$

is order elevated to $p = 3$, leading to the enriched knot vector

$$\boldsymbol{\Xi}^{*} = [0, 0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1, 1],$$

showing that the multiplicity of each knot was increased by one, but no new knots were added. The order elevated curve is geometrically and parametrically identical to the original one. The order elevation procedure can be seen as the IGA equivalent of p-refinement in FEM, where the order of the polynomial basis is increased.

K-refinement

In the context of IGA, there is one very important characteristic in the refinement procedures: the processes of order elevation and knot insertion do not commute. This property gives rise to an alternative refinement technique with no analogous in FEM.

Consider, for instance, the example given in Figure 3.6. Starting from a basis of order $p = 1$ and with a knot vector $\boldsymbol{\Xi} = [0, 0, 1, 1]$, a single knot $\xi^{*} = 0.5$ is inserted. Therefore, a refined two element mesh with a new knot vector $\boldsymbol{\Xi}^{*} = [0, 0, 0.5, 1, 1]$ is obtained. Afterwards, an order elevation of one is performed, leading to an increase in the multiplicity of each knot. Thus, the final knot vector will be given as $\boldsymbol{\Xi}^{*} = [0, 0, 0, 0.5, 0.5, 1, 1, 1]$. As can be seen, at $\xi^{*} = 0.5$ the basis still has C^0 continuity due to the multiplicity of 2, although the polynomial order is now $p = 2$.

Consider now the alternative case represented in Figure 3.7. In this example, and starting from the same knot vector $\boldsymbol{\Xi} = [0, 0, 1, 1]$ as before, the basis of order $p = 1$ is firstly order elevated to an order $p = 2$, leading to the refined knot vector $\boldsymbol{\Xi}^{*} = [0, 0, 0, 1, 1, 1]$. In the second step, a knot $\xi^{*} = 0.5$ is inserted, leading to the knot vector $\boldsymbol{\Xi}^{*} = [0, 0, 0, 0.5, 1, 1, 1]$. The basis (with the same polynomial order as in the previous example) will now present C^1 continuity at $\xi^{*} = 0.5$, since the multiplicity of the knot is only one. This procedure

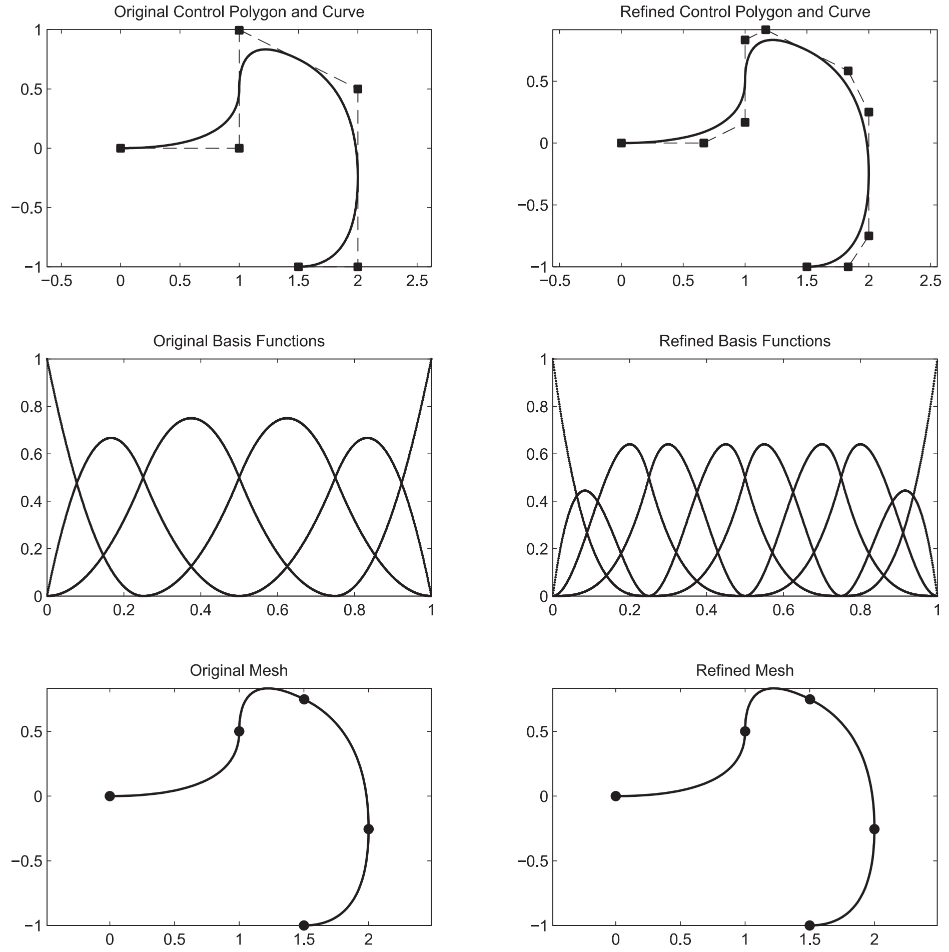


Figure 3.5: Order elevation example: (left) original curve of order $p = 2$ and (right) after order elevation to $p = 3$.

is known as k-refinement. Comparing Figures 3.6 and 3.7, it can be clearly seen that the k-refinement procedure leads to a basis in which the continuity at the location of the inserted knot is superior. Moreover, the control polygon resulting from this procedure contains one less control point and is, therefore, computationally more efficient.

3.2 Non-Uniform Rational B-Spline

Despite being a powerful tool, B-Spline are not able to represent some geometries, such as circles and ellipsoids. However, this problem can be circumvented by employing a generalised form of B-Spline known as Non-Uniform Rational B-Spline (NURBS). NURBS provide a single precise mathematical form capable of representing common analytical shapes such as lines, planes, conic curves, free-form curves and quadric surfaces that are used in computer graphics and CAD [Rogers 01].

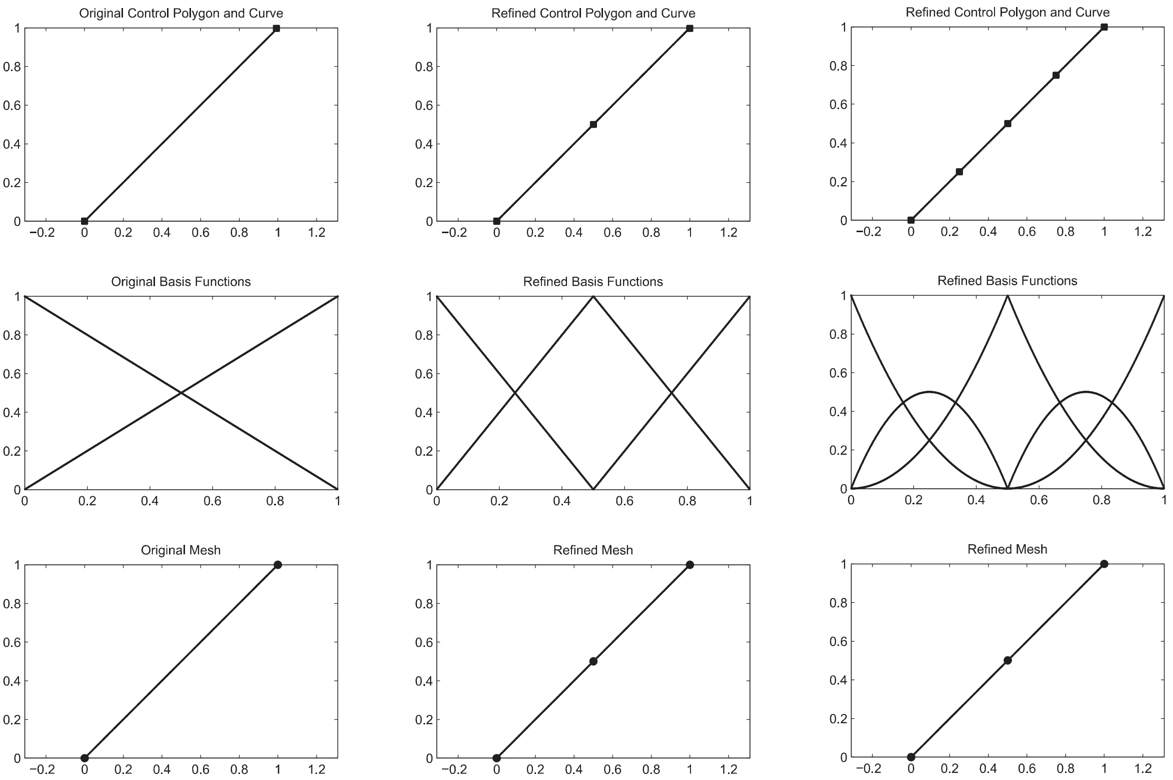


Figure 3.6: Example of successive refinements: (left) original curve, control polygon and basis, (centre) after knot insertion and (right) followed by order elevation.

In an analogous way to Equation 3.1, a NURBS curve of order p can be defined as

$$\mathbf{C}(\xi) = \sum_{i=1}^{n_c} R_i^p(\xi)\,\mathbf{B}_i, \tag{3.14}$$

where $R_i^p(\xi)$ are rational basis functions. These functions are defined as

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)\,w_i}{W(\xi)} = \frac{N_{i,p}(\xi)\,w_i}{\sum_{\hat{j}=1}^{n_c} N_{\hat{j},p}(\xi)\,w_{\hat{j}}}, \tag{3.15}$$

where, as seen before, $N_{i,p}(\xi)$ represents the i^{th} basis function of order p and w_i are selected weights. The choice of appropriate values of the weights w_i allows for a proper representation of different types of curves, such as circular arcs. A simple example to illustrate the influence of the weight in a NURBS curve is presented in Figure 3.8. In this example, the weight w_2 of the middle control point $\mathbf{B}_2$ is varied within the range $[0.25, 2.0]$. As can be seen, as the weight's value increases, the obtained NURBS curve tends to come closer to the control point.

As seen before for B-Splines, it is possible to define the NURBS basis functions for surfaces and volumes by means of the tensorial product feature, as

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi)\,M_{j,q}(\xi)\,w_{i,j}}{\sum_{\hat{i}=1}^{n_c}\sum_{\hat{j}=1}^{m_c} N_{\hat{i},p}(\xi)\,M_{\hat{j},q}(\eta)\,w_{\hat{i},\hat{j}}}, \tag{3.16}$$

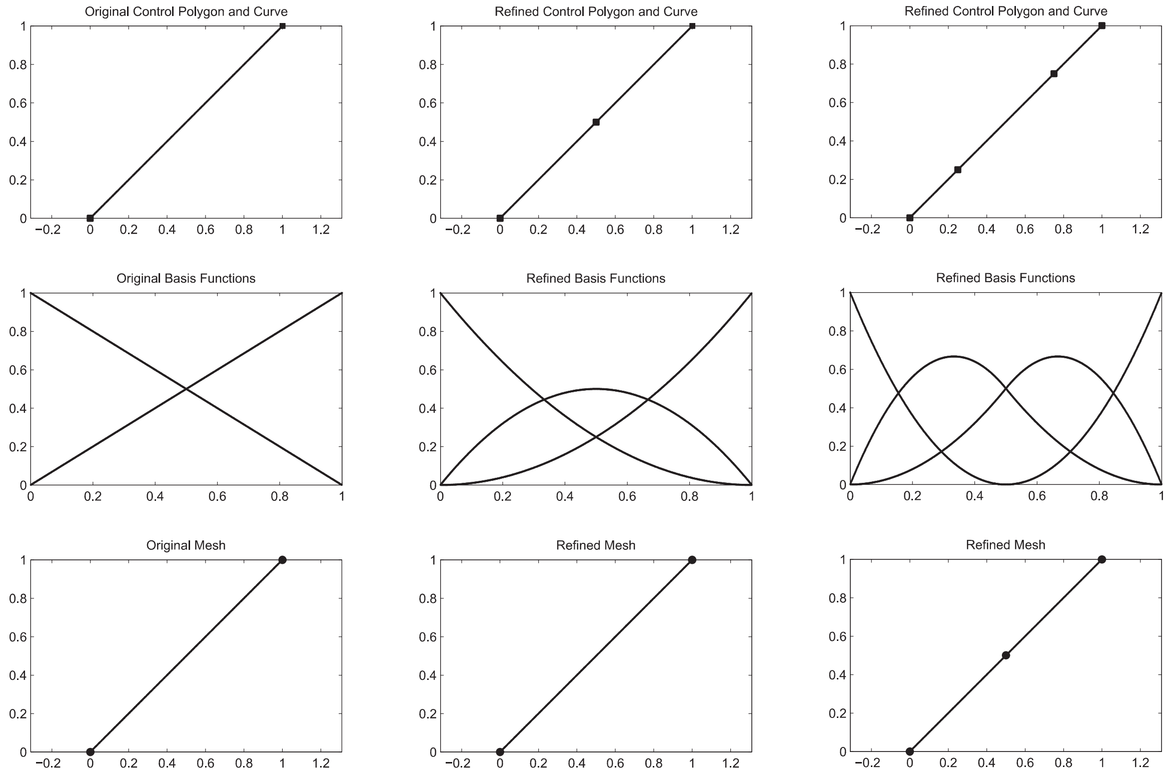


Figure 3.7: Example of k-refinement: (left) original curve, control polygon and basis, (centre) after order elevation and (right) followed by knot insertion.

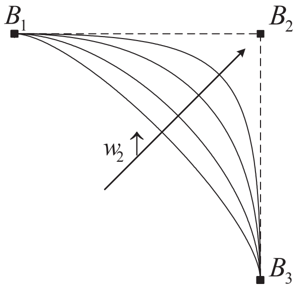


Figure 3.8: Influence of the weight of the middle control point in the NURBS curve (control polygon represented by dashed line).

and

$$R_{i,j,k}^{p,q,r}\left(\xi,\eta,\zeta\right)=\frac{N_{i,p}\left(\xi\right)M_{j,q}\left(\xi\right)L_{k,r}\left(\xi\right)w_{i,j,k}}{\sum_{\hat{i}=1}^{n_{\mathrm{c}}}\sum_{\hat{j}=1}^{m_{\mathrm{c}}}\sum_{\hat{k}=1}^{l_{\mathrm{c}}}N_{\hat{i},p}\left(\xi\right)M_{\hat{j},q}\left(\eta\right)L_{\hat{k},r}\left(\zeta\right)w_{\hat{i},\hat{j},\hat{k}}},\tag{3.17}$$

respectively.

By applying the quotient rule to Equation 3.15, one can obtain the first derivative of the NURBS basis function as

$$\frac{\partial R_i^p\left(\xi\right)}{\partial\xi}=R_{i,\xi}^p=w_i\frac{N_{i,p}'\left(\xi\right)W\left(\xi\right)-N_{i,p}\left(\xi\right)W'\left(\xi\right)}{\left(W\left(\xi\right)\right)^2},\tag{3.18}$$

where $N'_{i,p}(\xi) = \frac{\partial N_{i,p}(\xi)}{\partial \xi}$ and

$$W'(\xi) = \sum_{i=1}^{n_c} N'_{i,p}(\xi) \, w_i. \tag{3.19}$$

The NURBS basis functions inherit many of the properties of the B-Splines basis functions, such as the partition of unity and pointwise non-negativity. The algorithms described for order elevation and knot insertion can also be employed for the case of NURBS. It can be seen that if all points have the same weight, then $R_i^p(\xi) = N_{i,p}(\xi)$ and, therefore, B-Splines can be interpreted as a special case of NURBS, with Equation 3.14 being simplified to 3.1.

3.3 NURBS as Basis for Finite Element Analysis

As in classical FEM, Isogeometric Analysis also employs the isoparametric concept in the sense that the same parametrisation is used for the discrete solution variables and the geometry. However, one major difference can be identified between both approaches. In the case of FEM, the basis used to interpolate the unknown solution fields are also used to approximate the geometry. On the other hand, in IGA, the B-Splines/NURBS basis used to exactly interpolate the geometry are also employed to approximate the unknown solution variables.

In an Isogeometric Analysis, as introduced before, one can distinguish three domains: the physical space, the parametric space and the parent element space. In the single patch case presented in Figure 3.9, elements in these domains are represented by the volumes V, $\tilde{V}$ and $\bar{V}$, respectively. For the sake of simplicity, a two-dimensional representation is adopted in Figure 3.9, where the extension to 3D volumes is straightforward.

The physical space represents the actual geometry under analysis. This geometry is defined by the basis functions and the control points. The physical model can be divided into multiple patches, which can be seen as macro elements. Some geometries can be modelled using a single patch. In the parametric space, each patch is represented as a rectangle (or cuboid in a 3D case). The construction of the parametric space was detailed in Section 3.1.2. Finally, the numerical quadrature is performed at each parent element (represented by non-zero knot spans along each direction in the parent element domain) exactly as happens with FEM. The structure of an Isogeometric code is therefore very similar to the structure of a FEM code. Aside from the data input and results output, the major change resides in the computation of the basis functions (and their derivatives), which will replace the classical Finite Element shape functions. A detailed procedure on how to obtain the basis function for a general three-dimensional Isogeometric Analysis is given in the following.

Consider that the physical domain V is subdivided into n_e elements V^e (Figure 3.9).

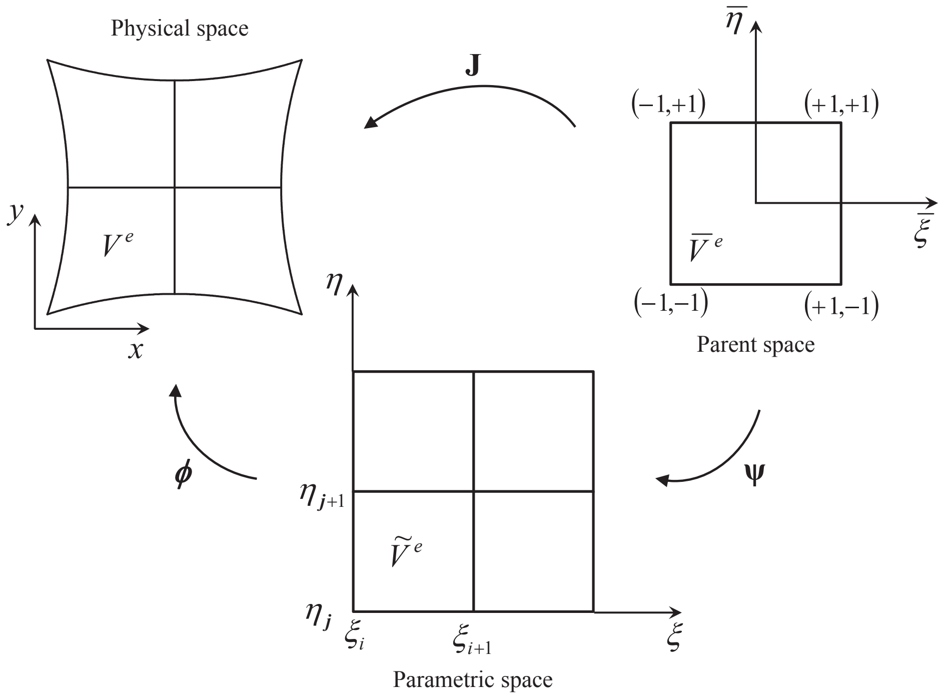


Figure 3.9: Representation of the different domains in Isogeometric Analysis.

Using a geometric mapping, the integrals in the physical domains can be pulled back to the parametric space $\tilde{V}^e$, which can then be pulled back to the parent element domain $\bar{V}^e$. Mathematically, this can be expressed by successive mappings as

$$\begin{aligned}\int_V f(x,y,z)\,\mathrm{d}V = \sum_{e=1}^{n_\mathrm{e}} \int_{V^e} f(x,y,z)\,\mathrm{d}V^e =\\ = \sum_{e=1}^{n_\mathrm{e}} \int_{\tilde{V}^e} f(\xi,\eta,\zeta)\,|\boldsymbol{\phi}|\mathrm{d}\tilde{V}^e =\\ = \sum_{e=1}^{n_\mathrm{e}} \int_{\bar{V}^e} f(\bar{\xi},\bar{\eta},\bar{\zeta})\,|\boldsymbol{\phi}||\boldsymbol{\psi}|\mathrm{d}\bar{V}^e.\end{aligned} \tag{3.20}$$

The integral can be evaluated using a standard $(p+1)\times(q+1)\times(r+1)$ Gaussian quadrature, where p, q, and r are the order of the NURBS basis in the ξ, η and ζ directions, respectively. However, it must be noted that this quadrature rule is not optimal when considering IGA and a number of studies have been carried out in order to propose alternative and optimal quadrature rules suitable for IGA [Hughes 10, Auricchio 12, Schillinger 14].

Starting from the parent element domain $\bar{V}^e$, the transformation to the parametric space $[\xi_i,\xi_{i+1}]\times[\eta_j,\eta_{j+1}]\times[\zeta_k,\zeta_{k+1}]$ can be obtained from the knot vectors and parent element coordinates as

$$\xi = \frac{(\xi_{i+1}-\xi_i)\,\bar{\xi}+(\xi_{i+1}+\xi_i)}{2}, \tag{3.21}$$

$$\eta = \frac{(\eta_{j+1}-\eta_j)\,\bar{\eta}+(\eta_{j+1}+\eta_j)}{2}, \tag{3.22}$$

and

$$\zeta = \frac{(\zeta_{k+1} - \zeta_k)\,\bar{\zeta} + (\zeta_{k+1} + \zeta_k)}{2}, \tag{3.23}$$

where $\left(\bar{\xi}, \bar{\eta}, \bar{\zeta}\right)$ are the integration point coordinates, which are the same as the integration point coordinates in the natural system in a Finite Element code, if a standard Gaussian quadrature is employed.

The Jacobian matrix representing the mapping between the parent and parametric domains can then be seen as a simple scaling of the element, given as

$$\boldsymbol{\psi} = \begin{bmatrix} \frac{\partial \xi}{\partial \bar{\xi}} & \frac{\partial \xi}{\partial \bar{\eta}} & \frac{\partial \xi}{\partial \bar{\zeta}} \\ \frac{\partial \eta}{\partial \bar{\xi}} & \frac{\partial \eta}{\partial \bar{\eta}} & \frac{\partial \eta}{\partial \bar{\zeta}} \\ \frac{\partial \zeta}{\partial \bar{\xi}} & \frac{\partial \zeta}{\partial \bar{\eta}} & \frac{\partial \zeta}{\partial \bar{\zeta}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \xi_{i+1} - \xi_i & 0 & 0 \\ 0 & \eta_{j+1} - \eta_j & 0 \\ 0 & 0 & \zeta_{k+1} - \zeta_k \end{bmatrix}, \tag{3.24}$$

and the determinant for this transformation matrix can be easily calculated as

$$|\boldsymbol{\psi}| = \frac{1}{8}\,(\xi_{i+1} - \xi_i)\,\left(\eta_{j+1} - \eta_j\right)\,(\zeta_{k+1} - \zeta_k)\,. \tag{3.25}$$

Defining $n_\mathrm{c} = (p+1)\,(q+1)\,(w+1)$ as the number of control points that belongs to the element connectivity, and using the parametric coordinates obtained from Equations 3.21 to 3.23, the NURBS basis functions

$$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{n_c} \end{bmatrix}, \tag{3.26}$$

and its derivatives

$$\mathbf{R}_{,\xi} = \begin{bmatrix} \frac{\partial R_1}{\partial \xi} \\ \frac{\partial R_2}{\partial \xi} \\ \vdots \\ \frac{\partial R_{n_c}}{\partial \xi} \end{bmatrix},\ \mathbf{R}_{,\eta} = \begin{bmatrix} \frac{\partial R_1}{\partial \eta} \\ \frac{\partial R_2}{\partial \eta} \\ \vdots \\ \frac{\partial R_{n_c}}{\partial \eta} \end{bmatrix},\ \text{and}\ \mathbf{R}_{,\zeta} = \begin{bmatrix} \frac{\partial R_1}{\partial \zeta} \\ \frac{\partial R_2}{\partial \zeta} \\ \vdots \\ \frac{\partial R_{n_c}}{\partial \zeta} \end{bmatrix}, \tag{3.27}$$

can be calculated using the procedures from Sections 3.1 and 3.2.

Once the derivatives of the basis functions with respect to the parametric coordinates are computed, and together with the coordinates of the control points, it is possible to determine the Jacobian matrix representing the geometry mapping between the parametric and physical spaces in the form

$$\boldsymbol{\phi} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \sum_{i=1}^{n_\mathrm{c}} \begin{bmatrix} \frac{\partial R_i}{\partial \xi} x_i & \frac{\partial R_i}{\partial \eta} x_i & \frac{\partial R_i}{\partial \zeta} x_i \\ \frac{\partial R_i}{\partial \xi} y_i & \frac{\partial R_i}{\partial \eta} y_i & \frac{\partial R_i}{\partial \zeta} y_i \\ \frac{\partial R_i}{\partial \xi} z_i & \frac{\partial R_i}{\partial \eta} z_i & \frac{\partial R_i}{\partial \zeta} z_i \end{bmatrix}. \tag{3.28}$$

It is also possible to compute the Jacobian matrix representing the mapping between the parent domain and the physical space as

$$\mathbf{J} = \boldsymbol{\phi}\boldsymbol{\psi}. \tag{3.29}$$

Finally, the derivatives of the basis functions with respect to the physical coordinates can be obtained as

$$\left[\mathbf{R}_{,x}\ \mathbf{R}_{,y}\ \mathbf{R}_{,z}\right] = \left[\mathbf{R}_{,\xi}\ \mathbf{R}_{,\eta}\ \mathbf{R}_{,\zeta}\right]\boldsymbol{\phi}^{-1}. \tag{3.30}$$

These derivatives can be employed to build the strain-displacement operator **B**, which has the same structure as the one presented for three-dimensional displacement-based finite elements in Section 2.2. The procedures to determine the stress and strain fields, as well as the elemental stiffness and load vectors, are also the same as for a general 3D Finite Element code, but with the difference that NURBS basis functions are used, instead of Lagrangian shape functions. The algorithm for the implementation of a NURBS-based finite element for linear elastic analysis can be seen in Box 3.1.

3.4 The Developed Tools for Isogeometric Analysis

Being a relatively recent research subject, commercial numerical simulation codes employing Isogeometric Analysis and NURBS-based formulations are not available. At the time of the writing of this Thesis, LS-DYNA provides an initial approach to IGA, by means of a generalised element concept and some adaptation of the input files used for FEM [Benson 10]. Also, the Finite Element Analysis Program (FEAP) [Zienkiewicz 05, Taylor 13] provides a beta version, under request, with some initial implementations of NURBS-based finite elements. There are also some available open-source packages based on IGA. For instance, *GeoPDEs*[1] is an Octave/Matlab code for solving partial differential equations for linear elasticity, fluid mechanics and electromagnetism. More recently, *igatools*[2], an isogeometric analysis library, was also released. This C++ code supports parallel processing and presents a higher computational efficiency for solving partial differential equations. Another open-source Matlab code is the *igafem*[3] software which allows to solve linear elastic problem in one, two and three dimensions, as well as inclusion and crack modelling.

Since the present research work is devoted to the analysis of single and multipatch solid mechanics problems in the nonlinear regime, it was necessary to develop and implement a software which would be suitable for these kind of problems. The *Isogeometric COde* (ICO)

[1] The *GeoPDEs* software is available at http://geopdes.apnetwork.it/

[2] The *igatools* software is available at http://code.google.com/p/igatools/

[3] The *igafem* software is available at http://sourceforge.net/projects/cmcodes/

Box 3.1: Algorithm for the implementation of a general displacement-based 3D NURBS-based element.

1. Initialise elemental stiffness $\mathbf{K}^e$ matrix
2. **DO** integration points' cycle
    (a) Compute parametric coordinates (ξ, η, ζ) from parent element coordinates $\left(\bar{\xi}, \bar{\eta}, \bar{\zeta}\right)$ as

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

    $$\eta = \frac{\left(\eta_{j+1} - \eta_j\right)\bar{\eta} + \left(\eta_{j+1} + \eta_j\right)}{2}$$

    $$\zeta = \frac{(\zeta_{k+1} - \zeta_k)\,\bar{\zeta} + (\zeta_{k+1} + \zeta_k)}{2}$$

    (b) Compute the Jacobian for the transformation between the parent element and the parametric spaces

    $$\boldsymbol{\psi} = \frac{1}{2}\begin{bmatrix} \xi_{i+1} - \xi_i & 0 & 0 \\ 0 & \eta_{j+1} - \eta_j & 0 \\ 0 & 0 & \zeta_{j+1} - \zeta_j \end{bmatrix}$$

    (c) Compute the NURBS basis functions $\mathbf{R}$ (Equation 3.15) and derivatives $\mathbf{R}_{,\xi}$, $\mathbf{R}_{,\eta}$ and $\mathbf{R}_{,\zeta}$ (Equation 3.18)

    (d) Compute the Jacobian for the transformation between the parametric and physical space

    $$\boldsymbol{\phi} = \sum_{i=1}^{n_c}\begin{bmatrix} \frac{\partial R_i}{\partial \xi}x_i & \frac{\partial R_i}{\partial \eta}x_i & \frac{\partial R_i}{\partial \zeta}x_i \\ \frac{\partial R_i}{\partial \xi}y_i & \frac{\partial R_i}{\partial \eta}y_i & \frac{\partial R_i}{\partial \zeta}y_i \\ \frac{\partial R_i}{\partial \xi}z_i & \frac{\partial R_i}{\partial \eta}z_i & \frac{\partial R_i}{\partial \zeta}z_i \end{bmatrix}$$

    (e) Using $\boldsymbol{\phi}$, compute the basis functions derivatives $\mathbf{R}_{,x}$, $\mathbf{R}_{,y}$ and $\mathbf{R}_{,z}$

    $$[\mathbf{R}_{,x}\ \mathbf{R}_{,y}\ \mathbf{R}_{,z}] = \left[\mathbf{R}_{,\xi}\ \mathbf{R}_{,\eta}\ \mathbf{R}_{,\zeta}\right]\boldsymbol{\phi}^{-1}$$

    (f) Build the strain-displacement operator **B**

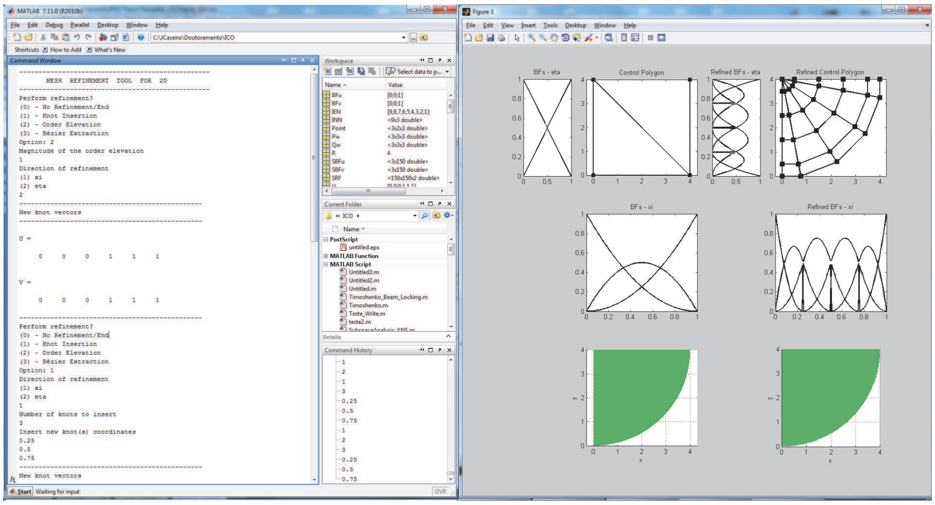
    (g) Compute the Gauss point contribution to the elemental stiffness matrix

    (h) If control points' displacements are available, compute the stress and strain fields
3. **END DO**

is an in-house developed code written in Fortran 90. The code is built in a modular fashion, allowing to easily include additional element formulations, material and contact models, *etc*. This was one of the main concerns when writing ICO, since the code intends to serve as a robust and easy to learn/modify tool, upon which future researchers can further use and improve. The code is currently composed of one main program and over 30 subroutines. In Appendix A the User's Manual of the Isogeometric COde is given.

3.4.1 ICO Pre-Processing Step

In order to perform a numerical analysis in ICO, an input file containing all the necessary data must be constructed. To create the mesh for the geometry, a mesh creation tool was developed using the Matlab programming language. As an input, the user must define the dimension, knot vectors and control points coordinates of the initial geometry. The user is then able to perform the refinement of the mesh using both the knot insertion and order elevation procedures. An example of a mesh refinement of a circular plate can be seen in Figure 3.10. The code then provides as an output the knot vectors and control points corresponding to the refined mesh. This information is then copied to the ICO input file, where all the information regarding boundary conditions, element types, material properties and analysis parameters are also defined.



```
---------------------------------------------------
       MESH  REFINEMENT  TOOL  FOR  2D
---------------------------------------------------
Perform refinement?
(0) - No Refinement/End
(1) - Knot Insertion
(2) - Order Elevation
(3) - Bézier Extraction
Option: 2
Magnitude of the order elevation
1
Direction of refinement
(1) xi
(2) eta
2
-------------------------------------------------
New knot vectors
-------------------------------------------------

U =

     0     0     0     1     1     1


V =

     0     0     0     1     1     1

-------------------------------------------------
Perform refinement?
(0) - No Refinement/End
(1) - Knot Insertion
(2) - Order Elevation
(3) - Bézier Extraction
Option: 1
Direction of refinement
(1) xi
(2) eta
1
Number of knots to insert
3
Insert new knot(s) coordinates
0.25
0.5
0.75
-------------------------------------------------
New knot vectors
```

Name	Value
BFu	[0;0;1]
BFv	[0;0;1]
IEN	[9,8,7,6,5,4,3,2,1]
INN	<9x3 double>
Point	<3x2x3 double>
Pw	<3x3x3 double>
Qw	<3x3x3 double>
R	4
SBFu	<3x150 double>
SBFv	<3x150 double>
SRF	<150x150x2 double>

Figure 3.10: Windows of the mesh creation tool.

3.4.2 ICO Analysis Step

The code architecture of the Isogeometric Code is very similar to that of a standard Finite Element code. The program starts by reading an input file containing all the information relative to the model. After the necessary matrices are allocated, the code then starts the analysis. The general flowchart of ICO can be seen in Figure 3.11. In a first step the connectivities of the elements are defined and all the global arrays are initialised and, afterwards, the code will enter the increment and iteration loops. In each iteration a element cycle is considered where the contributions of each element is added to the global system of equations. Iterations are performed until the solution is able to converge. In the end, all relevant data is written to an output file.

When considering a multipatch simulation, an additional cycle is introduced inside the iteration loop in order to allocate all the data relevant to the patch, with the remaining structure of the code being left unchanged. In the current version of the code, only compatible discretisations for the geometry can be employed in multipatch analyses of a single structure. Therefore, each control point on a face must be in a one-to-one correspondence with a control point from the adjoining face.

3.4.3 ICO Post-Processing Step

After performing the analysis, the displacements of the control points can be used to create the deformed geometry of the problem. To that end, a simple code was written in Matlab to read the displacement field and plot the deformed geometry of the structure under consideration.

3.4.4 Implementing NURBS-Based Elements in Abaqus

The developed software ICO is a valuable tool for implementing and testing different methodologies and formulations in the context of Isogeometric Analysis. However, the implementation of these methodologies within a commercial Finite Element code presents several advantages, such as lower computational costs for problems involving a high number of degrees-of-freedom and the possibility to use advanced solution techniques (as the Riks arc-length method) in nonlinear analysis and advanced material models.

In addition to the developed Isogeometric COde, a set of NURBS-based elements were also implemented in the commercial software Abaqus by means of developing User ELement (UEL) subroutines. The procedure to implement a NURBS-based formulation in Abaqus is similar to the one followed by standard Lagrangian-based formulations. In the Abaqus input file, all the information about the control points coordinates, elements connectivities and boundary conditions are defined. The UEXTERNALDB subroutine is then used in order to open and read external files with the objective of importing all the data relative to the numerical model which cannot be added to the input file (such as the knot vectors). This information is stored in a global module which is accessed whenever necessary. The UEL subroutine is afterwards responsible for the definition of the elemental stiffness matrix and

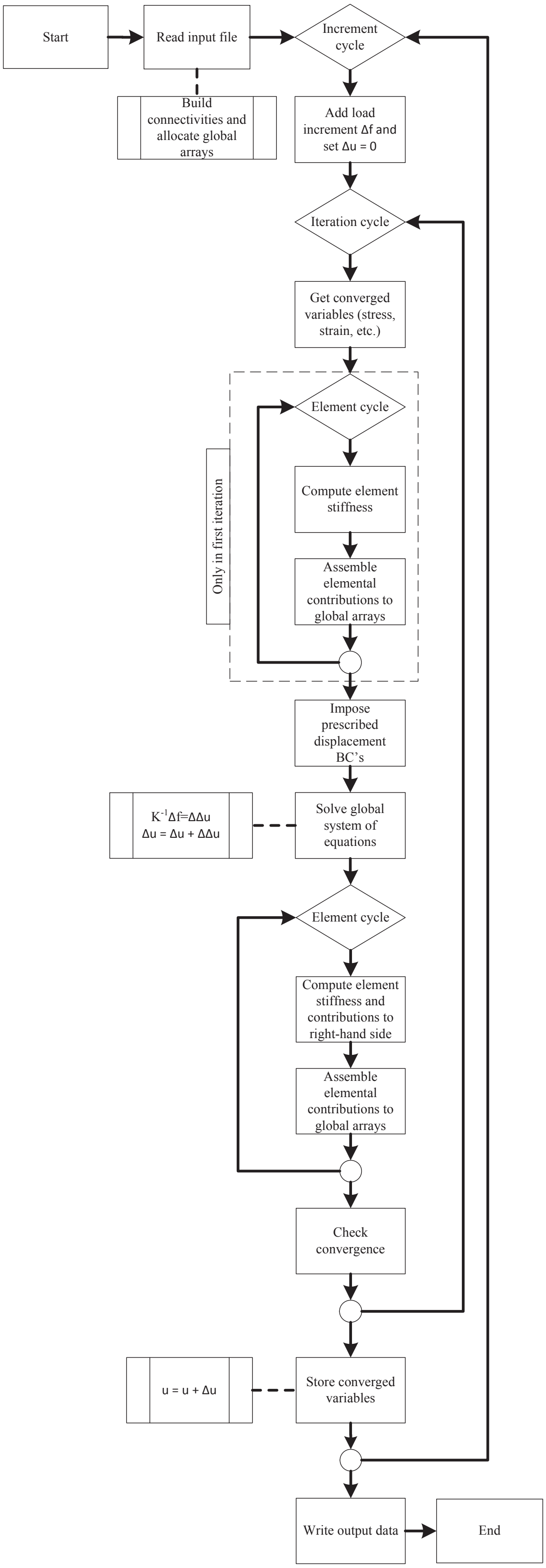

Start
Read input file
Increment cycle
Build connectivities and allocate global arrays
Add load increment Δf and set Δu = 0
Iteration cycle
Get converged variables (stress, strain, etc.)
Element cycle
Only in first iteration
Compute element stiffness
Assemble elemental contributions to global arrays
Impose prescribed displacement BC's
K^{-1}Δf=ΔΔu
Δu = Δu + ΔΔu
Solve global system of equations
Element cycle
Compute element stiffness and contributions to right-hand side
Assemble elemental contributions to global arrays
Check convergence
u = u + Δu
Store converged variables
Write output data
End

Figure 3.11: Workflow of the Isogeometric COde for a single patch analysis.

internal force vectors. Since all the NURBS data are available in the global module, these computations can be promptly performed.

The implementation in Abaqus was carried out in a way to make it possible to deal with multiple patches. To that end, each patch must be associated with different element types (U1, U2, ..., Un). When the UEL subroutine is called, the corresponding type of the element is read and all the data corresponding to the current patch is allocated and used to compute the elemental contribution. As in the ICO code, the multipatch methodology implemented in Abaqus is limited to compatible geometry discretisations, and the coincident control points are constrained using the *Multiple Point Constraint* (MPC) methodology.

In more recent versions, Abaqus also provides the possibility of codding user-defined elements with access to the entire material library of the software. The procedure to implement this subroutine (known as UELMAT) is similar to the standard UEL. The key difference resides in the fact that the user does not need to code the material subroutine by himself/herself, being the utility subroutine MATERIAL_LIB_MECH called instead. However, and considering for instance small strain plasticity, the UELMAT subroutine leads to higher computational costs when compared with the equivalent UEL. Nevertheless, the availability of using UELMAT subroutines opens up the very interesting possibility of using more advanced material models without the necessity of coding by the user. In Appendix B a detailed description of the implemented procedure using a single-patch UEL is given, together with examples of the necessary input files and the subroutine coding in Fortran.

Chapter 4

Topics in Nonlinear Formulations

A summary of nonlinear continuum mechanics is provided, with focus on the main topics that have been studied and implemented throughout the current work. The theoretical background of the adopted corotational approach is described, along with a detailed description concerning the implementation of numerical models for analysis including geometric nonlinearities, as well as the corresponding developed algorithms.

In general nonlinear analyses, a body subjected to a set of externally applied loads can undergo large rotations and/or deformations. In this situation, the final configuration of the body can significantly differ from the configuration at the start of the analysis. In order to solve this kind of problems, an incremental procedure is generally employed. This procedure consists in dividing the problem into small increments $(\ldots\ ,\ n-1,\ n,\ n+1,\ \ldots)$, where for each increment the equilibrium of the system is satisfied in an iterative way. The difficulty arises from the fact that, since the body can undergo large displacements and strains, the deformed configuration for the solution of step $n+1$ is not known (assuming that the solution is known up to step n). A solution can be nevertheless obtained by referring all the variables to a previously known equilibrium configuration. If the employed equilibrium configuration corresponds to the last converged increment n, then the formulation is termed Updated Lagrangian (UL). In contrast, if the initial configuration is assumed as the reference one, then a Total Lagrangian (TL) formulation is achieved. In the current work, an Updated Lagrangian formulation is considered for the nonlinearities' description.

4.1 Coordinate Systems

In a nonlinear analysis employing the UL formulation, a given particle can be referred to different coordinate systems. In the current work, four reference systems can be distinguished:

(i) A global coordinate system, defined by the coordinates $\left({}^{n}x, {}^{n}y, {}^{n}z\right)$ and $\left({}^{n+1}x, {}^{n+1}y, {}^{n+1}z\right)$, in the (updated) reference and current configurations, respectively. Note that the left upper index explicitly refers to the considered configuration (step);

(ii) A natural or parent coordinate system for integration purposes;

(iii) A parametric coordinate system, employed in Isogeometric Analysis to define the parametric space of a NURBS patch, as mentioned in the previous chapter;

(iv) A convective system, defined by $\mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \xi_i}$, where $\xi_1 = \xi$, $\xi_2 = \eta$ and $\xi_3 = \zeta$.

The components of the convective frame can be used to define a corotational reference system based on the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ as [Valente 04a]

$$\mathbf{r}_1 = \frac{\mathbf{g}_1}{|\mathbf{g}_1|}, \ \mathbf{r}_2 = \frac{\mathbf{g}_2}{|\mathbf{g}_2|}, \tag{4.1}$$

which can then be used to compute $\mathbf{r}_3$ as

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2. \tag{4.2}$$

This procedure can be used to determine an initial local coordinate system in the undeformed ($n = 0$) configuration. A representation of the different coordinate systems in the reference and current configurations can be seen in Figure 4.1.

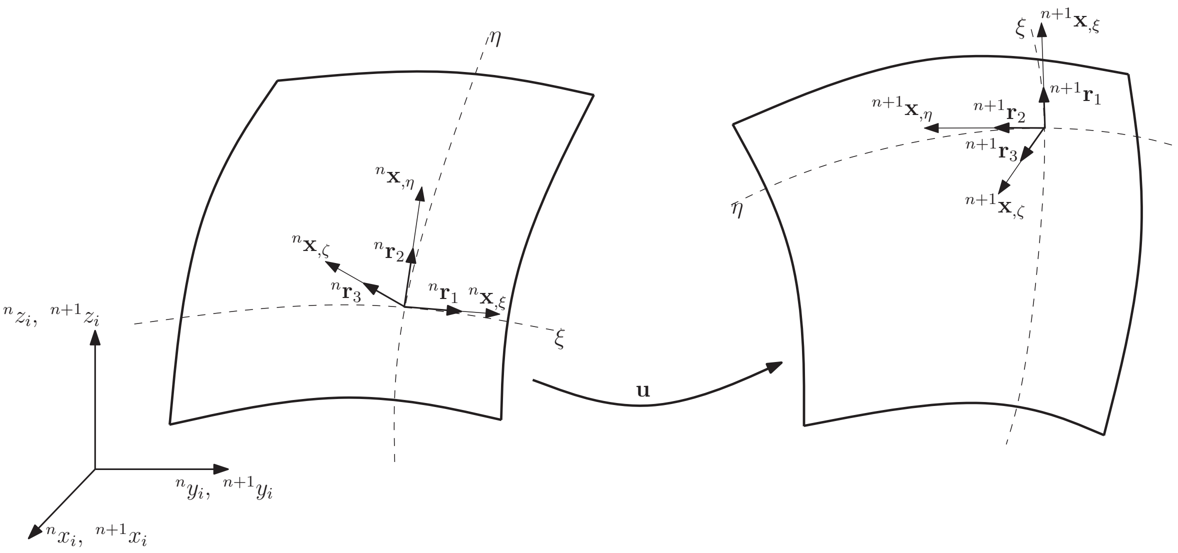

η
ξ
${}^{n+1}\mathbf{x}_{,\xi}$
${}^{n+1}\mathbf{r}_1$
${}^{n+1}\mathbf{x}_{,\eta}$
${}^{n+1}\mathbf{r}_2$
${}^{n+1}\mathbf{r}_3$
${}^{n+1}\mathbf{x}_{,\zeta}$
η
${}^{n}\mathbf{x}_{,\eta}$
${}^{n}\mathbf{x}_{,\zeta}$
${}^{n}\mathbf{r}_2$
${}^{n}\mathbf{r}_3$
${}^{n}\mathbf{r}_1$
${}^{n}\mathbf{x}_{,\xi}$
ξ
${}^{n}z_i,\ {}^{n+1}z_i$
$\mathbf{u}$
${}^{n}y_i,\ {}^{n+1}y_i$
${}^{n}x_i,\ {}^{n+1}x_i$

Figure 4.1: Coordinate systems in the (left) reference and (right) current configurations.

4.2 Nonlinear Continuum Mechanics

In the following, some concepts regarding stress and strain tensors are summarised. The analysis is far from being exhaustive and only focus on the fundamentals used in this work.

In order to describe the macroscopic deformation of a body, it is necessary to define a configuration at the start of the analysis (known as the reference configuration), with cartesian coordinates $\mathbf{X}$ and the configuration at an increment $(n+1)$ (assumed as the current configuration) with cartesian coordinates $\mathbf{x}$ (see Figure 4.2). For the sake of simplicity, the upper left indexes, shown in Figure 4.2 and defining the coordinates at each configuration, were dropped in the following equations, allowing to present a more general case and to adopt the nomenclature employed in many classical textbooks [Simo 98, Doghri 00, Belytschko 00]. The indexes will be recovered later on for the stress update procedure between states n and $n+1$.

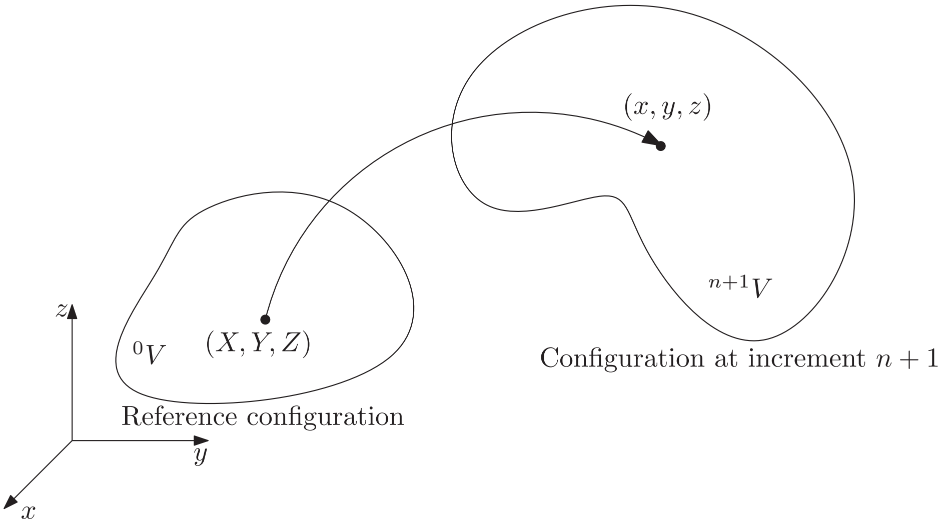


Figure 4.2: Position of a material particle at different configuration.

The deformation of a solid can be described by the mapping between the reference and current configurations by means of the so-called deformation gradient, which can be expressed as

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \tag{4.3}$$

or, alternatively,

$$\mathbf{F} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}, \tag{4.4}$$

where $\mathbf{u}$ is the displacement field and $\mathbf{I}$ the unit tensor. By making use of the deformation gradient, it is also possible to define, for instance, the right and left Cauchy-Green strain tensors as

$$\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}, \tag{4.5}$$

and

$$\mathbf{b}^{-} = \mathbf{F}\mathbf{F}^{\mathrm{T}}, \tag{4.6}$$

respectively. These strain tensors are important since they allow the definition of relevant strain measures, as presented in the following.

Since the deformation of a solid is described by the deformation gradient, this means that **F** contains the information about rigid body rotations as well as the stretches. To obtain this information in a decoupled way, the deformation gradient can be decomposed using the polar decomposition theorem. This theorem states that any non-singular, second-order tensor can be decomposed uniquely into the product of an orthogonal rotation tensor and a symmetric and positive definite stretch tensor [Dunne 06]. Applying the polar decomposition theorem to **F** results in

$$\mathbf{F} = \mathbf{R}\mathbf{U}, \tag{4.7}$$

where **R** is the orthogonal rotation tensor and **U** is the symmetric right stretch tensor. With this theorem it is therefore possible to obtain the rigid body rotation for any motion. This rotation tensor is especially useful in situations where a corotational coordinate system must be updated at each increment. Since **R** is an orthogonal tensor, it follows that

$$\mathbf{R}^{-1} = \mathbf{R}^{\mathrm{T}}. \tag{4.8}$$

The algorithm for the polar decomposition of the the deformation gradient **F** can be found in Box 4.1 [Valente 04a].

In analyses where large rotations and/or large strains are involved, it is necessary to employ a strain measure that must vanish in the presence of a rigid body motion (which is not the case of the left Cauchy-Green strain tensor). In the Finite Element Method (FEM), the Green-Lagrange strain **E** and the rate of deformation **D** tensors are the most widely used [Belytschko 00]. By means of the right Cauchy-Green strain tensor, it is possible to define the Green-Lagrange strain tensor in the form

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{C} - \mathbf{I}\right). \tag{4.9}$$

If a given body is subjected only to a rigid body motion, the deformation gradient results in $\mathbf{F} = \mathbf{R}$, which introduced into Equation 4.9 leads to

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{R}^{\mathrm{T}}\mathbf{R} - \mathbf{I}\right) = \frac{1}{2}\left(\mathbf{I} - \mathbf{I}\right) = 0, \tag{4.10}$$

proving that the **E** vanishes in the presence of rigid body motion only.

Substituting Equation 4.4 into 4.9 leads to

$$\mathbf{E} = \frac{1}{2}\left[\frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\right)^{\mathrm{T}}\right] + \frac{1}{2}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\right)^{\mathrm{T}}\frac{\partial \mathbf{u}}{\partial \mathbf{X}}, \tag{4.11}$$

showing that **E** is therefore composed by two terms. The first term is the infinitesimal (linear) strain tensor and the second term corresponds to the nonlinear part. The Green-Lagrange strain tensor can show the variation of length with respect to the reference configuration. If one requires the variation of the length with respect to the current configuration, the Almansi--Euler strain tensor can be employed, for example, which is defined as

$$\mathbf{e}^{\mathrm{A}} = \frac{1}{2}\left[\mathbf{I} - \left(\mathbf{b}^{-}\right)^{-1}\right] = \frac{1}{2}\left[\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right)^{\mathrm{T}}\right] - \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right)^{\mathrm{T}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}}. \tag{4.12}$$

The Almansi-Euler strain tensor is usually applied in Eulerian approaches, which is not focused in the current work.

4.2.1 Stress Measures

As happens with strain tensors, in nonlinear mechanics there are also different stress measures that can be employed. These stress measures can relate the deformed or undeformed configurations to the applied forces. Two of the most common stress measures in a nonlinear analysis are the Cauchy ($\boldsymbol{\sigma}$) and the second Piola-Kirchhoff (**S**) stresses. Since they are adopted in the present work, they will be detailed in the following.

The Cauchy stress can be interpreted as the ratio of the current force per unit of deformed area, *i.e*, it can be seen as a measure of the true stress in the deformed structure [Hinton 00]. Consider the body represented in Figure 4.3 in which the normal vector **n** of the surface of an elemental area dA is represented. The surface traction can then be expressed as

$$\mathbf{t} = \frac{\mathbf{f}}{\mathrm{d}A}, \tag{4.13}$$

where **f** is an elemental internal force which acts in the elemental area dA. The Cauchy stress tensor $\boldsymbol{\sigma}$ is then the projection of the surface traction into/over the unit normal vector **n**, as

$$\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}. \tag{4.14}$$

On the other hand, the first Piola-Kirchhoff stress relates forces in the present configuration with the (previous) undeformed configuration, and can be obtained from the Cauchy stress tensor as

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-\mathrm{T}}, \tag{4.15}$$

where $J = |\mathbf{F}|$ is the determinant of the deformation gradient. Finally, it is possible to define the second Piola-Kirchhoff stress tensor as

$$\mathbf{S} = J\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-\mathrm{T}}. \tag{4.16}$$

The second Piola-Kirchhoff **S** can be interpreted as the transformed current force which acts upon the undeformed area, being the work conjugate of the Green-Lagrange strain **E**, Box 4.1: Polar decomposition algorithm.

```
1. Compute the right Cauchy-Green strain tensor
       $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$
2. Compute the invariants of the right Cauchy-Green strain tensor
       $I_{\mathrm{C}} = \mathrm{tr}\left(\mathbf{C}\right), \quad II_{\mathrm{C}} = \frac{1}{2}\left[I_{\mathrm{C}}^2 - \mathrm{tr}\left(\mathbf{C}^2\right)\right], \quad III_{\mathrm{C}} = |\mathbf{C}|$
3. Compute variable $k$ as
       $k = I_{\mathrm{C}}^2 - 3II_{\mathrm{C}}$
4. IF $k \leq 10^{-5}$
    (a) Compute $\gamma$ as
           $\gamma = \sqrt{\frac{I_{\mathrm{C}}}{3}}$
    (b) Compute $\mathbf{U}$ and $\mathbf{U}^{-1}$
           $\mathbf{U} = \gamma\mathbf{I}$
           $\mathbf{U}^{-1} = \gamma^{-1}\mathbf{I}$
   ELSE
    (a) Directly compute the largest eigenvalue $\gamma$
           $l = I_{\mathrm{C}}^3 - \frac{9}{2}I_{\mathrm{C}}II_{\mathrm{C}} + \frac{27}{2}III_{\mathrm{C}}$
           $\theta = \cos^{-1}\left(\frac{l}{k^{\frac{3}{2}}}\right)$
           $\gamma^2 = \frac{1}{3}\left[I_{\mathrm{C}} + 2\sqrt{k}\cos\left(\frac{\theta}{3}\right)\right]$
    (b) Compute the invariants of $\mathbf{U}$
           $III_{\mathrm{U}} = \sqrt{III_{\mathrm{C}}}$
           $I_{\mathrm{U}} = \gamma + \sqrt{-\gamma^2 + I_{\mathrm{C}} + \frac{2III_{\mathrm{U}}}{\gamma}}$
           $II_{\mathrm{U}} = \frac{I_{\mathrm{U}}^2 - I_{\mathrm{C}}}{2}$
    (c) Compute $\mathbf{U}$ and $\mathbf{U}^{-1}$
           $\mathbf{U} = \left(\frac{1}{I_{\mathrm{U}}II_{\mathrm{U}} - III_{\mathrm{U}}}\right)\left[I_{\mathrm{U}}III_{\mathrm{U}}\mathbf{I} + \left(I_{\mathrm{U}}^2 - II_{\mathrm{U}}\right)\mathbf{C} - \mathbf{C}^2\right]$
           $\mathbf{U}^{-1} = \frac{1}{III_{\mathrm{U}}}\left(II_{\mathrm{U}}\mathbf{I} - I_{\mathrm{U}}\mathbf{U} + \mathbf{C}\right)$
   END IF
5. Compute the rotation tensor
       $\mathbf{R} = \mathbf{F}\mathbf{U}^{-1}$
```

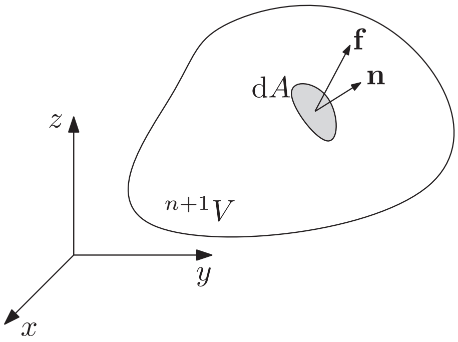


Figure 4.3: Body subject to elemental forces.

defined before. This stress measure is of extreme importance when considering an Updated Lagrangian formulation.

If the polar decomposition theorem is applied to the deformation gradient tensor in Equation 4.16, the following relation can be obtained:

$$\mathbf{S} = J\left(\mathbf{RU}\right)^{-1}\boldsymbol{\sigma}\left(\mathbf{RU}\right)^{-\mathrm{T}}. \tag{4.17}$$

When considering small strains but large displacements and arbitrary rotations, the stretch tensor can be approximately given as $\mathbf{U} \approx \mathbf{I}$, while the determinant of the deformation gradient turns to be equal to $J = |\mathbf{F}| = 1$. Also, from the property of the rotation tensor given in Equation 4.8, the second Piola-Kirchhoff stress can be approximated as

$$\mathbf{S} \approx \mathbf{R}^{\mathrm{T}}\boldsymbol{\sigma}\mathbf{R}. \tag{4.18}$$

Equation 4.18 shows that when considering small strains but large displacements and arbitrary rotations, the second Piola-Kirchhoff stress **S** is approximately equal to the rotated Cauchy stress $\boldsymbol{\sigma}$. For this reason, **S** is often known by the *material* or *co-rotational* stress [Doghri 00].

4.2.2 Constitutive Update

In the following, the constitutive update for small strains but arbitrarily large displacements and/or rotations is described. Consider the time derivative of the deformation gradient **F** as given by

$$\dot{\mathbf{F}} = \frac{\partial}{\partial t}\left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}}\right) = \frac{\partial \mathbf{v}}{\partial \mathbf{X}} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{LF}, \tag{4.19}$$

where

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \dot{\mathbf{F}}\mathbf{F}^{-1} \tag{4.20}$$

is known as the velocity gradient tensor. Applying the polar decomposition theorem to the Equation 4.20 leads to

$$\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1} = \left(\mathbf{R}\dot{\mathbf{U}} + \dot{\mathbf{R}}\mathbf{U}\right)\cdot\mathbf{U}^{-1}\mathbf{R}^{-1} = \dot{\mathbf{R}}\mathbf{R}^{\mathrm{T}} + \mathbf{R}\dot{\mathbf{U}}\mathbf{U}^{-1}\mathbf{R}^{\mathrm{T}}. \tag{4.21}$$

The tensor **L** can, on the other hand, be additively decomposed into a symmetric tensor **D** and an antisymmetric rotation tensor **W** as

$$\mathbf{L} = \mathbf{D} + \mathbf{W}, \tag{4.22}$$

in which

$$\mathbf{D} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^{\mathrm{T}}\right), \tag{4.23}$$

is known as the rate of deformation, and

$$\mathbf{W} = \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^{\mathrm{T}}\right) \tag{4.24}$$

is the spin tensor, representing the rate of deformation of the principal axes of tensor **D**. By substituting Equation 4.20 in 4.24, it comes that

$$\mathbf{W} = \frac{1}{2}\left[\dot{\mathbf{F}}\mathbf{F}^{-1} - \left(\mathbf{F}^{-1}\right)^{\mathrm{T}}\dot{\mathbf{F}}^{\mathrm{T}}\right]. \tag{4.25}$$

Applying the polar decomposition theorem of Equation 4.7 and after some manipulation, the spin tensor can now be written as [Dunne 06]

$$\mathbf{W} = \dot{\mathbf{R}}\mathbf{R}^{\mathrm{T}} + \frac{1}{2}\mathbf{R}\left[\dot{\mathbf{U}}\mathbf{U}^{-1}\left(\dot{\mathbf{U}}\mathbf{U}^{-1}\right)\mathbf{R}^{\mathrm{T}}\right], \tag{4.26}$$

where $\mathbf{\Omega} = \dot{\mathbf{R}}\mathbf{R}^{\mathrm{T}}$ is known as the rotation rate tensor.

In a corotational approach, a coordinate system is constructed for each point of the body. This coordinate system is rotated with the material by using the rotation tensor **R** coming from the polar decomposition of the deformation gradient **F**. In this situation, the principal material lines of **U** are assumed to be kept constant, while the product $\dot{\mathbf{U}}\mathbf{U}^{-1}$ is then equal to its symmetric part [Yoon 99a, Valente 04a, Alves de Sousa 06a]. Consequently, the rate of deformation and the spin tensors can be re-written as

$$\mathbf{D} = \mathbf{R}\dot{\mathbf{U}}\mathbf{U}^{-1}\mathbf{R}^{\mathrm{T}}, \tag{4.27}$$

and

$$\mathbf{W} = \mathbf{\Omega} = \dot{\mathbf{R}}\mathbf{R}^{\mathrm{T}}, \tag{4.28}$$

respectively. From Equation 4.27 it is now possible to define the corotational rate of the deformation tensor as

$$\hat{\mathbf{D}} = \mathbf{R}^{\mathrm{T}}\mathbf{D}\mathbf{R} = \frac{1}{2}\left(\dot{\mathbf{U}}\mathbf{U}^{-1} + \mathbf{U}^{-1}\dot{\mathbf{U}}\right), \tag{4.29}$$

which is energy conjugated with the corotational (rotated) Cauchy stress tensor, defined as

$$\hat{\boldsymbol{\sigma}} = \mathbf{R}^{\mathrm{T}}\boldsymbol{\sigma}\mathbf{R}. \tag{4.30}$$

The constitutive update between states (n) and $(n+1)$, now recovering the upper left indexes, can be written as

$$ {}^{n+1}\hat{\boldsymbol{\sigma}} = {}^{n}\hat{\boldsymbol{\sigma}} + \Delta t \frac{d\hat{\boldsymbol{\sigma}}}{dt}. \tag{4.31}$$

In an hypoelastic-plastic model, in which the elastic strains are small when compared to the plastic strains, it is possible to additively decompose the rate of deformation tensor into its elastic and plastic parts as [Belytschko 00]

$$\mathbf{D} = \mathbf{D}^{\text{el}} + \mathbf{D}^{\text{pl}}, \tag{4.32}$$

and, by making use of Equation 4.29, the corotational rate of deformation tensor comes as

$$\hat{\mathbf{D}} = \hat{\mathbf{D}}^{\text{el}} + \hat{\mathbf{D}}^{\text{pl}}. \tag{4.33}$$

The incremental rotated strain between states (n) and $(n+1)$ can now be calculated as [Yoon 99a, Yoon 99b]

$$ {}^{n+1}_{\;\;n}\hat{\mathbf{e}}^{\text{A}} = \int_{t_n}^{t_{n+1}} \hat{\mathbf{D}}\text{d}t \equiv {}^{n+\frac{1}{2}}_{\;\;n}\hat{\mathbf{D}}\Delta t. \tag{4.34}$$

The use of a *mid-point rule* ensures that a second-order accuracy in the stress update is achieved [Key 82, Hughes 84, Masud 00b]. This rule is based on a *mid-point configuration*, that can be obtained simply by dividing the incremental displacements by two. The constitutive update of the corotational (rotated) Cauchy stress tensor can now be obtained as

$$ {}^{n+1}\hat{\boldsymbol{\sigma}} = {}^{n}\hat{\boldsymbol{\sigma}} + {}^{n+1}_{\;\;n}\hat{\boldsymbol{\sigma}} = {}^{n}\hat{\boldsymbol{\sigma}} + \hat{\mathbf{C}}_4 \left({}^{n+1}_{\;\;n}\hat{\mathbf{e}}^{\text{A}} - {}^{n+1}_{\;\;n}\hat{\mathbf{e}}^{\text{A,pl}} \right), \tag{4.35}$$

which is formally identical to a constitutive update typically employed within an infinitesimal strain framework. By employing the relation defined in Equation 4.18, which is only valid for the case of small strains, the incremental corotational second Piola-Kirchhoff stress tensor can now be introduced, finally leading to [Masud 00b, Doghri 00]

$$ {}^{n+1}\hat{\boldsymbol{\sigma}} = {}^{n}\hat{\boldsymbol{\sigma}} + {}^{n+1}_{\;\;n}\hat{\mathbf{S}} = {}^{n}\hat{\boldsymbol{\sigma}} + \hat{\mathbf{C}}_4 \left({}^{n+1}_{\;\;n}\hat{\mathbf{E}} - {}^{n+1}_{\;\;n}\hat{\mathbf{E}}^{\text{pl}} \right), \tag{4.36}$$

demonstrating that the rotated increment of the second Piola-Kirchhoff can be directly summed to the converged Cauchy stress tensor. In the present work, the previous equation is employed to perform the constitutive update when small strain but large displacement and/or rotations are considered.

4.3 Geometric Nonlinearity

During nonlinear analyses, the change of the geometry of a given problem can significantly alter the nature of a problem. If geometric nonlinearities are being considered, the equilibrium conditions must then be written with respect to the current (deformed) configuration.

However, this deformed configuration in not known in advance. As mentioned before, geometric nonlinear problems can be analysed using a Total Lagrangian or an Updated Lagrangian formulation. The Updated Lagrangian formulation is employed in the current work and described in more detail in the following.

4.3.1 Updated Lagrangian Formulation

As stated before, in the UL formulation the configuration n is taken as the reference in order to obtain the new configuration $n+1$. According to Equation 2.12, the Principle of Virtual Work (PVW) at increment $n+1$, using an indicial tensorial form, can then be expressed as

$$\int_{^{n+1}V} {}^{n+1}\sigma_{ij}\delta\, {}^{n+1}\varepsilon_{ij}\mathrm{d}\,{}^{n+1}V = \delta\, {}^{n+1}\Pi^{\mathrm{ext}}, \tag{4.37}$$

where $\delta\, {}^{n+1}\Pi^{\mathrm{ext}}$ is the external virtual work at increment $n+1$. As can be seen, the integral on the left hand side of Equation 4.37 is computed over the volume ${}^{n+1}V$, which is unknown. Therefore, the equation cannot be directly solved in the current state since it is not possible to integrate over an unknown volume. Furthermore, it is not possible to directly work with the increments in the Cauchy stress $\boldsymbol{\sigma}$ because this tensor is always relative to the current geometry. This change in geometry can be dealt with by defining the appropriate stress and strain measures, which can be accomplished by employing the second Piola-Kirchhoff **S** stress and the Green-Lagrange **E** strain tensors. Therefore, Equation 4.37 can now be re-written as

$$\int_{^{n}V} {}^{n+1}S_{ij}\delta\, {}^{n+1}E_{ij}\mathrm{d}\,{}^{n}V = \delta\, {}^{n+1}\Pi^{\mathrm{ext}}. \tag{4.38}$$

The second Piola-Kirchhoff stress tensor at increment $n+1$ can be decomposed as

$${}^{n+1}S_{ij} = {}^{n}S_{ij} + {}^{n+1}_{n}S_{ij}, \tag{4.39}$$

where ${}^{n}S_{ij}$ is the stress at increment n and ${}^{n+1}_{n}S_{ij}$ is the increment in stress between states n and $n+1$. Similarly, the Green-Lagrange strain tensor can be decomposed as

$${}^{n+1}E_{ij} = {}^{n}E_{ij} + {}^{n+1}_{n}E_{ij}. \tag{4.40}$$

It is worth noting that because an UL formulation is being considered, all the quantities are referred to the last converged configuration n. Consequently, it is possible to admit that ${}^{n}S_{ij} = {}^{n}\sigma_{ij}$ since they both point to the (deformed) configuration at increment n, which is known. Additionally, the term ${}^{n}E_{ij} = 0$ since only the increments in displacements between state n to $n+1$ are used. Furthermore, and accordingly to Equation 4.11, the Green-Lagrange strain increment can be further decomposed into linear (**e**) and nonlinear ($\boldsymbol{\eta}$) components as

$${}^{n+1}_{n}E_{ij} = {}^{n+1}_{n}e_{ij} + {}^{n+1}_{n}\eta_{ij}. \tag{4.41}$$

By substituting Equations 4.39, 4.40 and 4.41 into 4.38, and after some manipulations, the PVW finally becomes [Bathe 96]

$$\int_{{}^{n}V} {}^{n+1}_{\;\;n}S_{ij}\,\delta\,{}^{n+1}_{\;\;n}E_{ij}\,\mathrm{d}\,{}^{n}V + \int_{{}^{n}V} {}^{n}\sigma_{ij}\,\delta\,{}^{n+1}_{\;\;n}\eta_{ij}\,\mathrm{d}\,{}^{n}V = \delta\,{}^{n+1}\Pi^{\mathrm{ext}} - \int_{{}^{n}V} {}^{n}\sigma_{ij}\,\delta\,{}^{n+1}_{\;\;n}e_{ij}\,\mathrm{d}\,{}^{n}V, \quad (4.42)$$

where the integral in the right-hand side of the equation corresponds to the internal virtual work associated to the stress tensor at increment n.

4.3.2 Finite Element Linearization

In order to obtain a Finite Element solution, Equation 4.42 must be linearized. To this end, the term ${}^{n+1}_{\;\;n}S_{ij}$ can be written using a Taylor expansion series in ${}^{n+1}_{\;\;n}E_{ij}$ as

$${}^{n+1}_{\;\;n}S_{ij} \approx \frac{\partial\,{}^{n+1}_{\;\;n}S_{ij}}{\partial\,{}^{n+1}_{\;\;n}E_{rs}}\,{}^{n+1}_{\;\;n}E_{rs} + (\ldots)\,, \quad (4.43)$$

where higher order terms were not considered. Furthermore, by neglecting the nonlinear terms in the Green-Lagrange strain ${}^{n+1}_{\;\;n}E_{ij}$, Equation 4.43 can now be re-written as

$${}^{n+1}_{\;\;n}S_{ij} \approx \frac{\partial\,{}^{n+1}_{\;\;n}S_{ij}}{\partial\,{}^{n+1}_{\;\;n}E_{rs}}\,{}^{n+1}_{\;\;n}e_{rs} = {}^{n+1}_{\;\;n}C_{ijrs}\,{}^{n+1}_{\;\;n}e_{rs}, \quad (4.44)$$

where ${}^{n+1}_{\;\;n}C_{ijrs}$ is the constitutive tensor. As a result, Equation 4.42 now becomes

$$\int_{{}^{n}V} {}^{n+1}_{\;\;n}C_{ijrs}\,{}^{n+1}_{\;\;n}e_{rs}\,\delta\,{}^{n+1}_{\;\;n}e_{ij}\,\mathrm{d}\,{}^{n}V + \int_{{}^{n}V} {}^{n}\sigma_{ij}\,\delta\,{}^{n+1}_{\;\;n}\eta_{ij}\,\mathrm{d}\,{}^{n}V = \delta\,{}^{n+1}\Pi^{\mathrm{ext}} - \int_{{}^{n}V} {}^{n}\sigma_{ij}\,\delta\,{}^{n+1}_{\;\;n}e_{ij}\,\mathrm{d}\,{}^{n}V, \quad (4.45)$$

which is the weak form needed for the development of the Finite Element model based on the UL formulation. When discretized using FEM (or IGA), Equation 4.45 can then be written as

$$({}^{n}\mathbf{K} + {}^{n}\mathbf{K}_{\mathrm{NL}})\,\Delta\mathbf{u} = {}^{n+1}\mathbf{f}^{\mathrm{ext}} - {}^{n}\mathbf{f}^{\mathrm{int}}, \quad (4.46)$$

where

$${}^{n}\mathbf{K} = \int_{{}^{n}V} ({}^{n}\mathbf{B})^{\mathrm{T}}\,{}^{n}\mathbf{C}_4\,{}^{n}\mathbf{B}\,\mathrm{d}\,{}^{n}V, \quad (4.47)$$

$${}^{n}\mathbf{K}_{\mathrm{NL}} = \int_{{}^{n}V} ({}^{n}\mathbf{B}_{\mathrm{NL}})^{\mathrm{T}}\,{}^{n}\boldsymbol{\sigma}\,{}^{n}\mathbf{B}_{\mathrm{NL}}\,\mathrm{d}\,{}^{n}V \quad (4.48)$$

and

$${}^{n}\mathbf{f}^{\mathrm{int}} = \int_{{}^{n}V} ({}^{n}\mathbf{B})^{\mathrm{T}}\,{}^{n}\bar{\boldsymbol{\sigma}}\,\mathrm{d}\,{}^{n}V. \quad (4.49)$$

Accordingly to Bathe [Bathe 96], the geometric nonlinear strain-displacement matrix can be written for the three-dimensional case in a compact form as

$${}^{n}\mathbf{B}_{\mathrm{NL}} = \begin{bmatrix} {}^{n}\widetilde{\mathbf{B}}_{\mathrm{NL}} & \widetilde{\mathbf{0}} & \widetilde{\mathbf{0}} \\ \widetilde{\mathbf{0}} & {}^{n}\widetilde{\mathbf{B}}_{\mathrm{NL}} & \widetilde{\mathbf{0}} \\ \widetilde{\mathbf{0}} & \widetilde{\mathbf{0}} & {}^{n}\widetilde{\mathbf{B}}_{\mathrm{NL}} \end{bmatrix}, \quad (4.50)$$

with

$$ {}^{n}\widetilde{\mathbf{B}}_{\mathrm{NL}} = \begin{bmatrix} {}^{n}N_{1,x} & 0 & 0 & {}^{n}N_{2,x} & \dots & {}^{n}N_{n_n,x} \\ {}^{n}N_{1,y} & 0 & 0 & {}^{n}N_{2,y} & \dots & {}^{n}N_{n_n,y} \\ {}^{n}N_{1,z} & 0 & 0 & {}^{n}N_{2,z} & \dots & {}^{n}N_{n_n,z} \end{bmatrix} \text{ and } \widetilde{\mathbf{0}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.51} $$

where n_n is the number of nodes of a given element and ${}^{n}N_{i,j}$ are the derivatives of the shape function i at increment n with respect to j. Finally, the Cauchy stress matrix takes the form

$$ {}^{n}\overline{\boldsymbol{\sigma}} = \begin{bmatrix} {}^{n}\widetilde{\boldsymbol{\sigma}} & \overline{\mathbf{0}} & \overline{\mathbf{0}} \\ \overline{\mathbf{0}} & {}^{n}\widetilde{\boldsymbol{\sigma}} & \overline{\mathbf{0}} \\ \overline{\mathbf{0}} & \overline{\mathbf{0}} & {}^{n}\widetilde{\boldsymbol{\sigma}} \end{bmatrix}, \tag{4.52} $$

in which

$$ {}^{n}\widetilde{\boldsymbol{\sigma}} = \begin{bmatrix} {}^{n}\sigma_{11} & {}^{n}\sigma_{12} & {}^{n}\sigma_{13} \\ {}^{n}\sigma_{21} & {}^{n}\sigma_{22} & {}^{n}\sigma_{23} \\ {}^{n}\sigma_{31} & {}^{n}\sigma_{32} & {}^{n}\sigma_{33} \end{bmatrix} \text{ and } \overline{\mathbf{0}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.53} $$

4.3.3 Finite Element Implementation

In a UL Finite Element code, it is convenient to consider a non-fixed, movable local coordinate system. Matrices and vectors in this coordinate system will be denoted using a hat superscript $(\hat{\cdot})$. Matrices and vectors in the convective system will, on the other hand, be denoted using a tilde superscript $(\tilde{\cdot})$. Since the strain-displacement operator **B**, defined in the previous sections, is calculated in the global configuration, it is important to define a transformation tensor **T** to map **B** into different coordinate systems. This transformation tensor can be defined as

$$ \mathbf{T} = \begin{bmatrix} A_{11}A_{11} & A_{12}A_{12} & A_{13}A_{13} & A_{11}A_{12} & A_{11}A_{13} & A_{11}A_{23} \\ A_{21}A_{21} & A_{22}A_{22} & A_{23}A_{13} & A_{21}A_{22} & A_{21}A_{23} & A_{21}A_{23} \\ A_{31}A_{31} & A_{32}A_{32} & A_{33}A_{33} & A_{31}A_{32} & A_{31}A_{33} & A_{31}A_{33} \\ 2A_{11}A_{21} & 2A_{12}A_{22} & 2A_{12}A_{23} & A_{11}A_{22}+A_{12}A_{21} & A_{11}A_{23}+A_{21}A_{13} & A_{12}A_{23}+A_{22}A_{13} \\ 2A_{11}A_{31} & 2A_{12}A_{32} & 2A_{13}A_{33} & A_{11}A_{32}+A_{12}A_{31} & A_{11}A_{33}+A_{31}A_{13} & A_{12}A_{33}+A_{32}A_{13} \\ 2A_{21}A_{31} & 2A_{22}A_{32} & 2A_{23}A_{33} & A_{21}A_{32}+A_{22}A_{31} & A_{21}A_{33}+A_{31}A_{23} & A_{22}A_{33}+A_{32}A_{23} \end{bmatrix}, \tag{4.54} $$

where

$$ \mathbf{A} = \mathbf{r}^{\mathrm{T}}, \tag{4.55} $$

for the global-to-local transformation, or

$$ \mathbf{A} = \mathbf{r}^{\mathrm{T}}\mathbf{J}^{-1}, \tag{4.56} $$

for the natural-to-local transformation, where **r** is defined as

$$ \mathbf{r} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}, \tag{4.57} $$

and **J** is the Jacobian matrix given from Equation 2.35 for FEM and from Equation 3.29 for IGA. The Finite Element implementation of a geometric nonlinear framework for a 3D solid element using the UL formulation can be seen in Box 4.2. It is worth mentioning that the algorithm in Box 4.2 is completely general and applicable for both the Finite Element Method and Isogeometric Analysis. It is also important to notice that a mid-point rule was employed in the relations presented in Box 4.2 for the calculation of the stress field that will lead to $\hat{\mathbf{K}}_{\mathrm{NL}}$ and $\hat{\mathbf{f}}^{\mathrm{int}}$ [Hughes 84, Masud 00b].

Box 4.2: Algorithm for the implementation of a geometric nonlinear 3D displacement-based formulation using an Updated Lagrangian formulation.

```
1. Initialise elemental stiffness matrix and internal force vector
2. Recover the local axis from the previous increment ${}^{n}\mathbf{r}$
3. DO integration points' cycle
    (a) Compute shape functions $N_i$ in the natural frame and its derivatives
    (b) Compute the Jacobian matrix for the mid-point ${}^{n+\frac{1}{2}}\mathbf{J}$ and end configurations ${}^{n+1}\mathbf{J}$
        $${}^{n+\frac{1}{2}}\mathbf{J}=\sum_{i=1}^{n_n}\begin{bmatrix}\frac{\partial N_i}{\partial\xi}{}^{n+\frac{1}{2}}x_i & \frac{\partial N_i}{\partial\xi}{}^{n+\frac{1}{2}}y_i & \frac{\partial N_i}{\partial\xi}{}^{n+\frac{1}{2}}z_i\\ \frac{\partial N_i}{\partial\eta}{}^{n+\frac{1}{2}}x_i & \frac{\partial N_i}{\partial\eta}{}^{n+\frac{1}{2}}y_i & \frac{\partial N_i}{\partial\eta}{}^{n+\frac{1}{2}}z_i\\ \frac{\partial N_i}{\partial\zeta}{}^{n+\frac{1}{2}}x_i & \frac{\partial N_i}{\partial\zeta}{}^{n+\frac{1}{2}}y_i & \frac{\partial N_i}{\partial\zeta}{}^{n+\frac{1}{2}}z_i\end{bmatrix}, \quad {}^{n+1}\mathbf{J}=\sum_{i=1}^{n_n}\begin{bmatrix}\frac{\partial N_i}{\partial\xi}{}^{n+1}x_i & \frac{\partial N_i}{\partial\xi}{}^{n+1}y_i & \frac{\partial N_i}{\partial\xi}{}^{n+1}z_i\\ \frac{\partial N_i}{\partial\eta}{}^{n+1}x_i & \frac{\partial N_i}{\partial\eta}{}^{n+1}y_i & \frac{\partial N_i}{\partial\eta}{}^{n+1}z_i\\ \frac{\partial N_i}{\partial\zeta}{}^{n+1}x_i & \frac{\partial N_i}{\partial\zeta}{}^{n+1}y_i & \frac{\partial N_i}{\partial\zeta}{}^{n+1}z_i\end{bmatrix}$$
    (c) Compute the deformation gradient for the mid-point ${}^{n+\frac{1}{2}}_{n}\mathbf{F}$ and end configurations ${}^{n+1}_{n}\mathbf{F}$
        $${}^{n+\frac{1}{2}}_{n}\mathbf{F}=\frac{\partial\,{}^{n+\frac{1}{2}}\mathbf{x}}{\partial\,{}^{n}\mathbf{x}}, \quad {}^{n+1}_{n}\mathbf{F}=\frac{\partial\,{}^{n+1}\mathbf{x}}{\partial\,{}^{n}\mathbf{x}}$$
    (d) Use the polar decomposition algorithm in Box 4.1 to obtain the rotation matrices ${}^{n+\frac{1}{2}}_{n}\mathbf{R}$ and ${}^{n+1}_{n}\mathbf{R}$
    (e) Update the local reference system as
        $${}^{n+\frac{1}{2}}\mathbf{r}={}^{n+\frac{1}{2}}_{n}\mathbf{R}\,{}^{n}\mathbf{r}$$
        $${}^{n+1}\mathbf{r}={}^{n+1}_{n}\mathbf{R}\,{}^{n}\mathbf{r}$$
    (f) Compute the strain-displacement operator in the global system in the mid-point ${}^{n+\frac{1}{2}}\mathbf{B}$ and end ${}^{n+1}\mathbf{B}$ configurations
    (g) Use the global-to-local transformation operator $\mathbf{T}$ (Equations 4.54 and 4.55) to obtain the strain displacement operators in the local coordinate system denoted as ${}^{n+\frac{1}{2}}\hat{\mathbf{B}}$ and ${}^{n+1}\hat{\mathbf{B}}$
    (h) Compute the stress and strain fields using ${}^{n+\frac{1}{2}}\hat{\mathbf{B}}$
    (i) Compute the stiffness matrix $\hat{\mathbf{K}}$ (Equation 4.47) and the internal force vector $\hat{\mathbf{f}}^{\text{int}}$ (Equation 4.49) using ${}^{n+1}\hat{\mathbf{B}}$
    (j) Compute the geometric nonlinear stiffness matrix $\hat{\mathbf{K}}_{\text{NL}}$ (Equation 4.48) and add this contribution to the elemental stiffness
4. END DO
5. Store the local axis in the end configuration ${}^{n+1}\mathbf{r}$ to use in the next increment
```

Chapter 5

Finite Element Technology

In this chapter, the locking phenomena that can pollute numerical analyses based on FEM and well as IGA are described. This is followed by a state-of-the-art review of the main methodologies used to alleviate these non-physical phenomena in the context of both approaches. A special focus is given to the Enhanced Assumed Strain (EAS) and Assumed Natural Strain (ANS) methods and their possible application in Isogeometric Analysis. An innovative extension of the Assumed Natural Strain method is proposed in the context of IGA, leading to the development of high-order NURBS-based solid-shell elements, suitable for the analysis of thin structures. Finally, some insight into volumetric locking in the context of IGA is also provided.

5.1 The Locking Phenomena

Standard displacement-based low-order Finite Element formulations are widely used in many applications mainly due to their simplicity and effectiveness. However, these formulations can often be affected by spurious strain or stress fields which lead to an overestimation of the stiffness matrix. As a consequence, this results on the underestimation of the nodal displacements, which are then said to become *locked*. Different types of locking can be related to the shape of the element employed when discretising the structure (high length-to-thickness ratio, trapezoidal shapes, *etc.*) or with material properties (near incompressibility) [Hughes 87]. These distinct types of locking will be explained in more detail in the following sections.

5.1.1 Volumetric Locking

When considering an incompressible (or near-incompressible) analysis, displacement-based low-order Finite Element formulations present an overly stiff behaviour, leading to inaccurate results. The compressible behaviour of a given solid continuum is defined by the Poisson's coefficient ν, and as this coefficient tends to the incompressibility limit $(\nu \to 0.5)$ the material is said to become incompressible. Values of ν in the vicinity of 0.5 occur, for instance, in rubber-like materials. The constitutive relation for an isotropic elastic situation can be expressed by

$$\sigma_{ij} = 2G\varepsilon_{ij} + \lambda\varepsilon_{kk}\delta_{ij}, \tag{5.1}$$

where δ_{ij} is the Kronecker delta. The Lamé parameters used in Equation 5.1 are given by

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \tag{5.2}$$

and

$$G = \frac{E}{2(1+\nu)}. \tag{5.3}$$

From Equation 5.2 it can be seen that λ will tend to infinity as ν tends to the incompressibility limit. However, since in this situation ε_{kk} tends to zero, the stress field will still have acceptable results.

Taking into account the Lamé parameters, the isotropic elastic constitutive tensor $\mathbf{C}_4$ can be written in matrix form as

$$\mathbf{C}_4 = \begin{bmatrix} 2G+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2G+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2G+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix}. \tag{5.4}$$

Since the elemental stiffness matrix for a given finite element is given as

$$\mathbf{K}^e = \int_{V^e} \mathbf{B}^{\mathrm{T}}\mathbf{C}_4\mathbf{B}\mathrm{d}V^e,$$

it can be concluded that some terms of $\mathbf{K}$ will become very large as ν tends to 0.5. The elemental stiffness will then be assembled into the global system, with the nodal displacements being obtained as

$$\mathbf{d} = \mathbf{K}^{-1}\mathbf{f}. \tag{5.5}$$

If the coefficients in the stiffness matrix $\mathbf{K}$ are excessively high, the terms in its inverse will tend to zero, which may lead to a null displacement field. This material-based phenomena is known as *volumetric locking*.

5.1.2 Shear Locking

From the continuum mechanics theory, a given solid subjected to a pure bending situation will show no transverse shear strain in the deformed configuration. However, as can be seen in Figure 5.1, when considering a linear finite element subjected to a pure bending load, the element is not able to properly represent the displacement field. Since the element topology is only capable of mimicking a linear displacement field, the top and bottom surfaces will not present a curved pattern, leading instead to the appearance of spurious shear strains. As a consequence, this non-physical shear strain leads to inaccurate stress fields, with the corresponding underestimation of the displacement field. This phenomena is known as *transverse shear locking*.

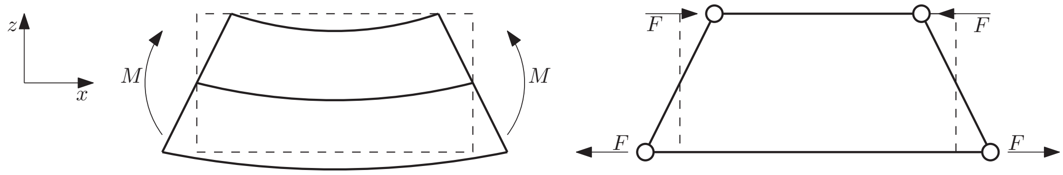


Figure 5.1: Structure under bending load in (left) continuum mechanics and (right) Finite Element discretisation.

Shear locking effects are usual in plate and shell Finite Element formulations based on the Reissner-Mindlin theory as thickness values tend to zero. This is due to the fact that the ratio between the overall dimensions of the elements, compared to the thickness, can become excessive which can on turn lead to the ill-conditioning of the numerical solution. This phenomena is also observable when using solid or solid-shell finite elements in the analysis of structures with low thickness values and/or in bending dominated problems.

5.1.3 Thickness Locking

Considering, once again, the structure subjected to a pure bending deformation, as depicted in Figure 5.1, from the continuum mechanics theory, the structure will be subjected to a linear stress component σ_{xx}, while all other stress components are zero. Following the Hooke's Law

$$\boldsymbol{\sigma} = \mathbf{C}_4 : \boldsymbol{\varepsilon},$$

the strain components will be given as

$$\varepsilon_{xx} = \frac{\sigma_{xx}}{E}, \tag{5.6}$$

$$\varepsilon_{yy} = \varepsilon_{zz} = -\nu\varepsilon_{xx}, \tag{5.7}$$

$$\gamma_{xy} = \gamma_{xz} = \gamma_{yz} = 0. \tag{5.8}$$

From the previous equations it can be concluded that, when considering a non-zero Poisson's coefficient, ε_{zz} presents a linear behaviour since σ_{xx} also presents a linear distribution.

Discretising the structure using a low-order finite element will lead to a constant strain ε_{zz}. Additionally, due to the inextensibility assumption of the normal fibres in pure bending deformation cases, this will lead to $\varepsilon_{zz} = 0$. Consequently, since ε_{xx} and ε_{yy} will have a linear distribution, the coupling between the in-plane and normal strain components will also enforce a linear distribution of the normal stress as

$$\sigma_{zz} = \lambda \varepsilon_{xx} + \lambda \varepsilon_{yy} + (\lambda + 2G)\, \varepsilon_{zz} = \lambda \left(\varepsilon_{xx} + \varepsilon_{yy}\right), \tag{5.9}$$

which, according to the continuum mechanics equations, should be zero. This leads to an overly stiff response of the system, which is known as *thickness locking*.

5.1.4 Trapezoidal Locking

The modelling of curved structures using low-order solid elements will result in finite elements with a trapezoidal shape. Due to this mesh distortion effect, when considering pure bending states, spurious transverse normal strains will appear. The appearance of these non-physical strains when oblique element edges are present is labelled as *trapezoidal* or *curvature thickness locking*.

To illustrate this phenomena, MacNeal [MacNeal 94] proposed the following example, consisting on a trapezoidal structure under a pure bending loading and with $\nu = 0.0$ discretised by a single quadrilateral element, as shown in Figure 5.2, along with its representation in the parent element domain. The parameter a represents the curvature of the structure. The corresponding strain components in the element are given as

$$\varepsilon_{xx} = \frac{\zeta - a}{1 - a\zeta}, \tag{5.10}$$

$$\varepsilon_{zz} = \Lambda a, \tag{5.11}$$

$$\gamma_{xz} = \Lambda \xi \left[1 + \frac{a\,(\zeta - a)}{1 - a\zeta}\right]. \tag{5.12}$$

On the other hand, from continuum mechanics the analytical solutions are

$$\varepsilon_{xx}^{\text{ref}} = \zeta, \tag{5.13}$$

$$\varepsilon_{zz}^{\text{ref}} = 0, \tag{5.14}$$

$$\gamma_{xz}^{\text{ref}} = 0. \tag{5.15}$$

Comparing the analytical and numerical solutions, it can be seen that if no distortion is considered (*i.e.* $a = 0$) the in-plane strain components are the same. However, in the shear strain component it can be seen a spurious component, which is responsible for the shear locking phenomena described before. When mesh distortion appears the strain components are affected by spurious numerical strains, underestimating the displacement field and thus leading to a locked solution.

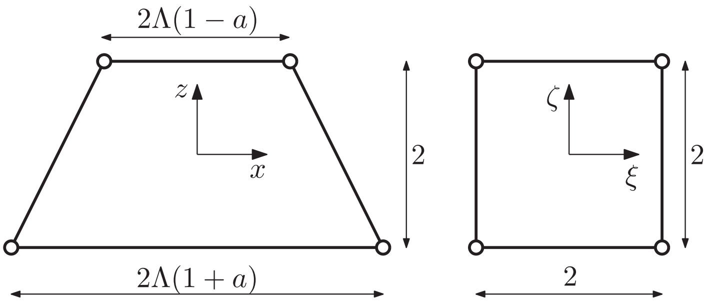


Figure 5.2: Low-order quadrilateral element with trapezoidal shape in (left) global space and in (right) parent domain.

5.1.5 Membrane Locking

Membrane locking is a pathology which results from the inability of an element to bend without stretching, appearing only in curved beams and curved shell elements. When the curved element is unable to represent the inextensional behaviour typical of pure bending, spurious membrane strain energy terms appear, causing the element to lock. Nevertheless, if a flat element is used to model a curved structure it will not be affected by membrane locking unless the element becomes warped. Linear triangles, for instance, are always flat and therefore free from membrane locking.

5.2 Treatment of Locking in Finite Element Analysis - A Review

The above mentioned non-physical locking pathologies are undesirable phenomena which affect the efficiency of finite elements. As a consequence, in the past decades an extensive amount of work has been performed in order to alleviate locking pathologies in finite elements. Some of the most relevant contributions are briefly presented in the following paragraphs.

The *reduced integration* (RI) and *selective reduced integration* (SRI) techniques were among the first ones used to alleviate locking problems [Zienkiewicz 71, Hughes 78]. Due to lower quadrature rules employed, the elements are able to represent deformation patterns that fully integrated elements cannot. Relevant work in the field of RI/SRI methodologies can be found in some classical papers in the literature [Belytschko 91, Belytschko 92, Belytschko 94, Liu 94, Wriggers 96, Liu 98, Reese 00, Reese 02]. However, given the rank deficiency of the resultant stiffness matrix, these schemes have poor performances in bending-dominated problems and can often lead to spurious deformation patterns. Moreover, SRI can only be applied to models where the small-strain tensor can be decomposed into volumetric and deviatoric parts. In addition, when modelling nonlinear effects in thin-walled structures these formulations can present a lower computational performance due to the need of multiple layers along the thickness direction in order to provide more integration points along this direction.

Introduced by Sussman and Bathe [Sussman 87], the *mixed displacement-pressure* (u-p) formulation consists in independently interpolating both displacement and pressure degrees-of-freedom. This formulation arises from the fact that when near incompressibility is considered, in order to accurately determine the volumetric stress, the pressure must also be taken as a solution variable. However, one of the key points of this formulation is the correct choice of the displacement and pressure interpolations which can lead to an efficient Finite Element formulation. When considering the pressure interpolation, the formulation can be separated into two main categories: (i) the pressure is taken as an elemental variable and can be statically condensed out, prior to the element assembly, or (ii) the pressure is defined as a nodal variable, leading to pressure continuity between elements. In the latter case, the pressure variables cannot be statically condensed. Consequently, a variety of interpolation schemes for displacement and pressure can be considered [Bathe 96, Zienkiewicz 00].

However, in order to guarantee that the Finite Element formulation is stable and convergent, the element must satisfy the Babuška-Brezzi (BB) condition which is a fundamental test in mixed finite element formulations [Brezzi 91]. Due to its strictness, many two field u-p elements (such as the linear triangle and quadrilateral) do not pass the BB condition. Consequently, these elements present instabilities in the pressure field, leading to the necessity of employing stabilisation techniques. For instance, the *mini-element* [Arnold 84] is a linear triangle with continuous piecewise linear interpolation function for velocity and pressure. In this formulation, the pressure and velocity are interpolated using the same functions. However, the velocity degrees-of-freedom are increased by adding an interpolation point in the element centre, where the velocity field is enhanced by means of a cubic bubble function. Nevertheless, the mini formulation is affected by small oscillations in pressure and the inertial terms are affected by the bubble mode, when considering transient problems [Cisloiu 08]. However, by introducing a stabilising parameter, Lee and co-workers [Lee 09] were able to apply triangular and tetrahedral mini-elements to forging simulation. Although a good agreement between the numerical and experimental data was found, the authors concluded that the solution was dependent, in some extent, of the employed stabilisation parameter. Another technique for stabilising the mixed u-p formulation is the sub-grid scale stabilisation approach, first proposed by Hughes [Hughes 95] for incompressible fluid dynamics and later on applied to solid mechanics by Chiumenti [Chiumenti 02], as well as to incompressible J_2-pasticity and damage problems with strain localisation [Cervera 03, Chiumenti 04, Cervera 04a, Cervera 04b, Cervera 09]. In this method, the continuous field is decomposed into fine and coarse components, corresponding to different length scales. Although being able to circumvent the BB condition, this method is dependent on material and geometric parameters and is computationally expensive, due to the introduction of additional degrees-of-freedom [Cisloiu 08].

Developed by Oñate [Oñate 04], and following a different approach, the *finite calculus* method can be employed to tackle volumetric locking. The basis of this method consists in the satisfaction of the equations of balance of momentum in a finite size domain. Volumetric locking is overcame by adding enhancing terms to equations obtained from a Taylor expansion where only the high-order ones are retained.

The $\bar{\mathbf{B}}$ (read *B-bar*) approach introduced by Hughes [Hughes 77] was also proposed to solve nearly incompressible problems. This method consists in splitting the strain-displacement matrix into its dilatational and deviatoric components. The former term is then replaced by another (under evaluated) one in order to reduce the contribution of the volumetric component to the solution. In a later work, Simo and co-authors [Simo 85] showed that the $\bar{\mathbf{B}}$ method resulted for Finite Element approximations constructed based on a three-field variational formulation. The authors then formulated the *mixed variational methods*, where the goal was to construct an assumed-strain approach in which only the dilatational part of the displacement gradient would be the independent variable. The formulation was developed in order to account for the incompressibility constraint which results from the plastic flow volume preservation [Simo 98].

With the goal of overcoming volumetric locking, de Souza Neto *et al.* [de Souza Neto 96] introduced the $\bar{\mathbf{F}}$ (read *F-bar*) method. This method consists in modifying the standard finite element internal force vector by replacing the deformation gradient **F** with an assumed modified gradient, $\bar{\mathbf{F}}$, when computing the Cauchy stress tensor. The result is a constraint relaxation, allowing to overcome volumetric locking pathologies in large strain hyperelastic and plastic problems. The $\bar{\mathbf{F}}$ method was later applied to linear triangular and tetrahedral elements in the large strain analysis of nearly incompressible solids by de Souza Neto and co-workers [de Souza Neto 05].

Another technique to tackle locking pathologies is the *Enhanced Assumed Strain* (EAS) method, firstly introduced by Simo and co-authors [Simo 90b, Simo 92]. Starting from a three-field variational formulation, the strain field of each finite element can be enlarged with the inclusion of a set of enhancing internal variables, gathering a wider dimension for the deformation subspace of the element and, therefore, resulting in additional deformation modes. However, the use of a large number of enhancing variables will inevitably lead to a computationally ineffective element. As examples of EAS-based Finite Element formulations, the 21-EAS variables approach of Andelfinger and Ramm [Andelfinger 93], as well as the 18-EAS variables solid [Alves de Sousa 02] and 12-EAS variables solid-shell [Alves de Sousa 03] elements can be cited, among many others [Korelc 96, Roehl 96, César de Sá 99, Armero 00, Kasper 00, Piltner 00, César de Sá 02, Korelc 10, Caseiro 13]. Pantuso and Bathe [Pantuso 95] presented a linear 2D quadrilateral element with continuous pressure interpolation enhanced with a 6 parameter field. The element was applied to linear problems in incompressibility and fluid flow, and extensions to axisymmetry and three-dimensional cases were also presented. In a later work [Pantuso 97], the formulation was expanded to incompressible problems in the finite strain regime. However, the authors concluded that the element was not suitable for this kind of analysis due to the development of hourglass modes, corresponding to the appearance of excessively large non-physical eigenvalues.

When considering enhanced strain techniques, low-order displacement based triangle/tetrahedral elements show no improvement when considering the additional strain degrees-of-freedom [Reddy 95]. However, for the case of a u-p formulation, it is possible to select effective enhanced strain modes leading to stable mixed formulations [Lovadina 03, Auricchio 05]. In this context, Zienkiewicz *et al.* [Zienkiewicz 00] introduced a *mixed-enhanced strain stabilisation technique* that was later on applied by Taylor [Taylor 00]. The latter author employed a three-field form involving continuous displacements and pressures and discontinuous volume change in the numerical analysis of small and finite deformation problems using low-order tetrahedral elements. In addition, an enhanced strain technique was used to stabilise the formulation when considering nearly incompressible problems. In their work, Mahnken and co-workers [Mahnken 08a, Mahnken 08b] applied volume and area bubble functions to enhance the displacement and strain fields, respectively, leading to significant damping of oscillations in mixed tetrahedrons in the small strains regime. This work was further extended by Caylak and Mahnken [Caylak 12] to hyper-elasticity at large deformations by introducing constant stabilisation matrices in the iterative Newton algorithm.

In order to specifically eliminate transverse shear locking, the *Assumed Natural Strain* (ANS) method was developed. The methodology was first implemented by Hughes and Tezduyar [Hughes 81] for Mindlin plates and later for shell elements by Dvorkin and Bathe [Dvorkin 84]. The ANS method consists in interpolating the strain field at a set of distinct points (known as *tying points*) whose strain terms will replace the standard strain values coming from the quadrature points. This technique has been applied in the improvement of reduced integration shell elements [Belytschko 94] as well as to fully and reduced integration solid-shell elements, as can be seen in [Hauptmann 98, Sze 00, Cardoso 08, Schwarze 09, Schwarze 11], to name only a few.

In their work, Bonet and Burton [Bonet 98] proposed an *averaged nodal pressure* tetrahedron in an explicit framework. In this technique nodal volumes are defined, which are then used to average the nodal pressures over each element. Based on this concept, a nodally averaged strain field formulation was proposed by Dohrmann *et al.* [Dohrmann 00] for small strain applications. This formulation was then extended to finite strain problems by using the deformation gradient tensor as the main kinematic variable [Bonet 01] or by employing an additional stabilisation term based on a modified material law [Puso 00]. Gee *et al.* [Gee 09] improved the formulation proposed by Puso and Solberg [Puso 00] by introducing a general splitting of the stress into volumetric and isochoric components in a variational consistent manner. By applying the stabilisation to the isochoric components, the uniform nodal strain method becomes stable while maintaining the benefits coming of the nodally averaged approach with respect to the volumetric stress components. Andrade Pires *et al.* [Andrade Pires 04] derived an implicit version of the averaged nodal pressure formulation, and a linear triangle for implicit plane strain and axisymmetric analysis of nearly incompressible solids under finite strains was then obtained. The *nodally integrated continuous element* (NICE), developed by Krysl and Zhu [Krysl 08], was derived from a weighted residual statement that weakly enforces both the balance and the kinematic equations, being proposed to specifically solve volumetric locking. In this assumed-strain technique, the weak kinematic equation is separately considered from the weak balance equation, aimed to satisfy it *a priori*. This methodology was successfully applied to triangular, tetrahedral and hexahedral elements [Krysl 08], as well as to Reissner-Mindlin plates [Castellazzi 09]. More recently, Krysl and Kagey [Krysl 12] proposed a modification to the NICE elements, in order to eliminate the sensitivity to mesh distortion present in the original formulation. Castellazzi and Krysl [Castellazzi 12] improved the NICE linear elements by deriving, in a consistent manner, a patch-averaged strain matrix for each node, leading to a smooth representation of the stress and strain fields.

In the last years, significant research effort has been employed in the development of the so-called *solid-shell* class of elements. The main goal of these elements is to combine the advantages of both solid and shell elements. This type of formulation is particularly attractive as only displacement degrees-of-freedom are used in its kinematic description, allowing to automatically account for 3D constitutive relations (*e.g.*, plasticity) and to obtain, as a consequence, a correct prediction of thickness changes in shell-like structures. Solid-shell elements also show strong advantages in numerical simulations involving double-sided contact situations, due, once again, to the correct modelling of the stress and strain fields through the thickness direction. In addition, also as a result of rotational degrees-of-freedom not being employed, the coupling with other solid elements in the mesh is straightforward and, most importantly, there is no need for non-trivial update procedures for rotation-like nodal variables in nonlinear geometric formulations. Nevertheless, this class of elements is also affected by locking pathologies when considering incompressible materials, high length-to-thickness ratios and/or when modeling curved structures. Relevant Finite Element solid-shell formulations are described in [Hauptmann 98, Vu-Quoc 03a, Valente 04b, Alves de Sousa 05, Alves de Sousa 06b, Harnau 06, Reese 07, Schwarze 09], and references therein.

5.3 Treatment of Locking in Isogeometric Analysis - A Review

Since the introduction of IGA, it has been shown that the high regularity properties of the employed functions can lead in many cases to superior accuracy per degree-of-freedom with respect to standard FEM (see, for example, [Cottrell 06, Cottrell 07, Cottrell 09]).

However, it is well-known that NURBS-based element formulations are not completely free from locking pathologies. This can be seen, for instance, in the work of Echter and Bischoff [Echter 10] where the performance of classical Finite Elements and NURBS-based elements was compared. In this work, convergence rates were analyzed, as well as the appearance of transverse shear and membrane locking. The authors concluded that the higher order continuity of the NURBS basis functions can significantly improve the quality of the numerical results. Nevertheless, the authors also state that the use of linear, quadratic or cubic basis functions can still lead to results that are not locking-free.

Therefore, the alleviation of pathologies such as volumetric, shear and membrane locking in NURBS-based elements is still an open issue. Elguedj *et al.* [Elguedj 08] employed the $\bar{\mathbf{B}}$ and $\bar{\mathbf{F}}$ projection methods to avoid volumetric locking in small and large deformation elasticity and plasticity problems in high-order solid NURBS elements. This projection methodology consists in splitting the volumetric and deviatoric components of the strain-displacement/deformation gradient matrix, then calculating a new volumetric counterpart in a projected space of one order lower than the displacement space. Due to the higher inter-element continuity in the IGA formulation, this projection must be performed at the patch level. Numerical results show that the methodology is able to obtain good convergence rates and good quality solutions. It was also shown that the $\bar{\mathbf{F}}$ method can alleviate shear locking for quadratic and higher-order basis functions. Taylor [Taylor 11] proposed a formulation based on a three-field variational structure for the analysis of near incompressible solids in the large deformation regime. It is shown that a formulation where displacements, mean stress and volume variables are independently approximated may be used to efficiently solve this kind of problems. Cardoso and César de Sá [Cardoso 12] combined the EAS method with isogeometric analysis to alleviate volumetric locking in 2D elastic problems. The choice of the EAS parameter was motivated by a subspace analysis of the incompressible deformation subspace [Alves de Sousa 03]. However, this formulation requires an *ad-hoc* stabilization term to prevent spurious solutions arising when higher-order NURBS polynomials are employed.

Focusing specifically in the alleviation of transverse shear locking, Echter and Bischoff [Echter 10] have extended the Discrete Shear Gap (DSG) method to NURBS-based beam elements. Beirão da Veiga *et al.* [Beirão da Veiga 12] implemented an isogeometric collocation method for straight planar Timoshenko beams, based on a mixed formulation scheme and leading to a shear locking-free formulation, which has been extended to spatial rods by Auricchio *et al.* [Auricchio 13]. Bouclier *et al.* [Bouclier 12] investigated the use of selective reduced integration and the $\bar{\mathbf{B}}$ strain projection methods as means of alleviating shear and membrane locking in planar curved beams. In a later work [Bouclier 13b], the same authors employed this methodology to alleviate locking pathologies in 2D solid elements for the analysis of both thick and thin beams. In addition, a simple extension to 3D NURBS based solid-shell elements was also presented. More recently, Bouclier and co-workers [Bouclier 13a], proposed two solid-shell NURBS elements. The first uses a $\bar{\mathrm{B}}$-formulation, leading to a high quality element but with a fully populated global stiffness matrix. To overcome this drawback, the authors then proposed a local least-squares type of procedure to create a locally projected $\bar{\mathbf{B}}$. This methodology allowed to obtain the global stiffness matrix in a simpler and more effective manner, but at the expense of a decrease in the element accuracy.

In the scope of plate/shell elements, Echter and co-workers [Echter 13] have proposed a hierarchic family of isogeometric shell formulations. Although being based on a non-mixed concept, these methods are able to remove transverse shear and curvature thickness locking. Membrane locking is, in this case, alleviated by means of the DSG method or, alternatively, by a hybrid-mixed formulation based on a two-field Hellinger-Reissner variational principle (displacements and stress fields). To alleviate shear locking in Reissner-Mindlin plate elements, Thai *et al.* [Thai 12] have implemented a stabilization technique that consists in modifying the shear terms of the constitutive matrix. Hosseini *et al.* [Hosseini 13] proposed a solid-like shell element, a class of shell elements characterized by possessing only displacement degrees-of-freedom but shell kinematics. In order to obtain a complete 3D representation of the shell, the authors employed NURBS/T-Splines basis functions to parametrize the mid-surface and linear Lagrange shape functions in the thickness direction. Benson and co-workers [Benson 11] proposed a quadratic rotation-free isogeometric shell formulation with a 2×2 reduced integration, reporting a significant reduction in the computational costs. In a later work, Benson *et al.* [Benson 13] proposed an isogeometric quadratic blended shell formulation and concluded that the use of uniformly reduced integration leads to a computationally efficient formulation. Kiendl *et al.* [Kiendl 09] developed an isogeometric Kirchhoff-Love shell element for geometric nonlinear applications. This formulation relies on displacement degrees-of-freedom only and due to the Kirchhoff kinematics shear locking is precluded *ab initio*.

In the following, the EAS and ANS methodologies for locking treatment in FEM are described in detail. In addition, it is proposed the extension of the ANS method in order to alleviate locking pathologies in Isogeometric Analysis.

5.4 The Enhanced Assumed Strain Method

Introduced by Simo and co-authors [Simo 90b, Simo 92], the Enhanced Assumed Strain (EAS) method propose the improvement of the compatible strain field $\mathbf{E}^u$ by means of an enhanced strain field $\mathbf{E}^\alpha$. The basis of the EAS method is the Veubeke-Hu-Washizu three--field functional [Bischoff 97, Valente 04b] that, for static cases, is written as

$$\Pi^{\mathrm{VHW}}(\mathbf{u},\mathbf{E},\mathbf{S}) = \int_V W(\mathbf{E})\mathrm{d}V + \int_V \mathbf{S} : \left[\frac{1}{2}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}_2\right) - \mathbf{E}\right] \mathrm{d}V - \Pi^{\mathrm{ext}}, \tag{5.16}$$

in which W is the strain energy and the virtual work of the external loading Π^{ext} is given by

$$\Pi^{\mathrm{ext}} = \int_V \mathbf{u} \cdot \mathbf{b}\rho \mathrm{d}V + \int_{S_{\mathrm{N}}} \mathbf{u} \cdot \mathbf{t}\mathrm{d}S, \tag{5.17}$$

where **b** and **t** are the prescribed volume and traction vectors over the control volume V and surface S_{N}. In Equation 5.16, the displacement vector **u**, the Green-Lagrange strain tensor **E** and the Second Piola-Kirchhoff stress tensor **S** are taken as the independent variables.

The enhanced strain field can be decomposed into a displacement-based and an incompatible (enhanced) part, expressed as

$$\mathbf{E} = \mathbf{E}^u + \mathbf{E}^\alpha. \tag{5.18}$$

This additive approach for the total strain field, introduced in [Simo 90b] for linear problems, can be still applied in nonlinear problems [Valente 04b, Miehe 04], being computationally simpler than the multiplicative decomposition of the deformation gradient originally introduced in [Simo 92].

By imposing the orthogonality condition [Simo 90b] between the stress field and the enhancing strain field

$$\int_V \mathbf{S} : \mathbf{E}^\alpha = 0, \tag{5.19}$$

the functional in Equation 5.16 can be re-written as

$$\Pi^{\mathrm{HWV}}(\mathbf{u},\mathbf{E}^{\alpha}) = \int_V W(\mathbf{E}^u + \mathbf{E}^{\alpha})\mathrm{d}V - \Pi^{\mathrm{ext}}, \tag{5.20}$$

which only has two independent variables. By applying the Gâteaux derivative, one can obtain the weak form of the modified functional [Vu-Quoc 03b], expressed as

$$\delta\Pi\left(\mathbf{u},\mathbf{E}^{\alpha}\right) = \delta\Pi^{\mathrm{int}} - \delta\Pi^{\mathrm{ext}}, \tag{5.21}$$

where

$$\delta\Pi^{\mathrm{int}} = \int_V \left(\delta\mathbf{E}^u + \delta\mathbf{E}^{\alpha}\right) : \frac{\partial W\left(\mathbf{E}^u + \mathbf{E}^{\alpha}\right)}{\partial\left(\mathbf{E}^u + \mathbf{E}^{\alpha}\right)} \tag{5.22}$$

and

$$\delta\Pi^{\mathrm{ext}} = \int_V \delta\mathbf{u}\cdot\mathbf{b}\rho\mathrm{d}V + \int_{S_N} \delta\mathbf{u}\cdot\mathbf{t}\mathrm{d}S. \tag{5.23}$$

Developing a truncated Taylor series around an arbitrary n^{th} state, the weak form can be expanded to obtain [Bischoff 97]:

$$\delta\Pi\left({}^{n+1}\mathbf{u},{}^{n+1}\mathbf{E}^{\alpha}\right) \approx \delta\Pi\left({}^{n}\mathbf{u},{}^{n}\mathbf{E}^{\alpha}\right) + \Psi\left[\delta\Pi\right]\left({}^{n}\mathbf{u},{}^{n}\mathbf{E}^{\alpha}\right).\left({}^{n+1}_{\ \ n}\mathbf{u},{}^{n+1}_{\ \ n}\mathbf{E}^{\alpha}\right). \tag{5.24}$$

5.4.1 Implementing the EAS method

In the element domain, and by making use of the standard isoparametric compatible shape function arranged in matrix form $(\mathbf{N})$, the displacement field can be interpolated as

$$\mathbf{u} \approx \mathbf{u}^{\mathrm{h}} = \mathbf{N}\mathbf{d},$$

$$\delta\mathbf{u} \approx \delta\mathbf{u}^{\mathrm{h}} = \mathbf{N}\delta\mathbf{d}, \tag{5.25}$$

$${}^{n+1}_{\ \ n}\mathbf{u} \approx {}^{n+1}_{\ \ n}\mathbf{u}^{\mathrm{h}} = \mathbf{N}\,{}^{n+1}_{\ \ n}\mathbf{d}, \tag{5.26}$$

where $\mathbf{d}$ is the vector of elemental degrees-of-freedom and the superscript $(\cdot)^{\mathrm{h}}$ represents, as stated before, the Finite Element approximation. Moreover, it is possible to define the relation between the enhanced Green-Lagrange strain tensor and $\mathbf{d}$, over the element domain, in the form

$$\mathbf{E} = \mathbf{B}\mathbf{d} + \mathbf{B}^{\alpha}\boldsymbol{\alpha}, \tag{5.27}$$

where $\mathbf{B}$ and $\mathbf{B}^{\alpha}$ are the strain-displacement operators for the displacement and enhanced variables, respectively. The second member of the right-hand side of the linearised weak form presented in Equation 5.24 can be re-written as [Valente 04b, Bischoff 97, Vu-Quoc 03b, Klinkel 97]

$$\Psi\left[\delta\Pi\right]\left(\mathbf{d},\boldsymbol{\alpha}\right)\cdot\left({}^{n+1}_{\ \ n}\mathbf{d},{}^{n+1}_{\ \ n}\boldsymbol{\alpha}\right) = \frac{\partial\left(\delta\Pi^{\mathrm{int}} - \delta\Pi^{\mathrm{ext}}\right)}{\partial\left(\mathbf{d},\boldsymbol{\alpha}\right)}\cdot\left({}^{n+1}_{\ \ n}\mathbf{d},{}^{n+1}_{\ \ n}\boldsymbol{\alpha}\right). \tag{5.28}$$

According to the interpolation functions, the two variations introduced before can be expressed as

$$\delta\Pi^{\text{int}}\left(\mathbf{d},\boldsymbol{\alpha}\right)=\delta\mathbf{d}^{\text{T}}\int_V\hat{\mathbf{B}}^{\text{T}}\hat{\mathbf{S}}\ \text{d}V+\delta\boldsymbol{\alpha}^{\text{T}}\int_V\left(\hat{\mathbf{B}}^{\alpha}\right)^{\text{T}}\hat{\mathbf{S}}\ \text{d}V, \tag{5.29}$$

and

$$\delta\Pi^{\text{ext}}\left(\mathbf{d}\right)=\delta\mathbf{d}^{\text{T}}\int_V\mathbf{N}^{\text{T}}\mathbf{b}\rho\ \text{d}V+\delta\mathbf{d}^{\text{T}}\int_{S_{\text{N}}}\mathbf{N}^{\text{T}}\mathbf{t}\ \text{d}S. \tag{5.30}$$

Taking Equation 5.29 into more detail, it can be stated that

$$\begin{aligned}\Psi\left[\delta\Pi^{\text{int}}\right]\cdot\left({}^{n+1}_{\ \ n}\mathbf{d},{}^{n+1}_{\ \ n}\boldsymbol{\alpha}\right)&=\frac{\partial\left(\delta\Pi^{\text{int}}\right)}{\partial\mathbf{d}}\cdot{}^{n+1}_{\ \ n}\mathbf{d}+\frac{\partial\left(\delta\Pi^{\text{int}}\right)}{\partial\boldsymbol{\alpha}}\cdot{}^{n+1}_{\ \ n}\boldsymbol{\alpha}=\\&=\delta\mathbf{d}^{\text{T}}\left[\left(\hat{\mathbf{K}}^{uu}+\hat{\mathbf{K}}^{uu}_{\text{NL}}\right){}^{n+1}_{\ \ n}\mathbf{d}+\hat{\mathbf{K}}^{u\alpha}\,{}^{n+1}_{\ \ n}\boldsymbol{\alpha}\right]+\delta\boldsymbol{\alpha}^{\text{T}}\left[\hat{\mathbf{K}}^{\alpha u}\,{}^{n+1}_{\ \ n}\mathbf{d}\hat{\mathbf{K}}^{\alpha\alpha}\,{}^{n+1}_{\ \ n}\boldsymbol{\alpha}\right].\end{aligned} \tag{5.31}$$

The linear $\hat{\mathbf{K}}^{uu}$ and geometric nonlinear stiffness $\hat{\mathbf{K}}^{uu}_{\text{NL}}$ matrices are defined in the same manner as in a conventional displacement-based formulation and as described in Section 4.3.2. Due to the fact that the enhanced parameters are included in the variational formulation, two coupling stiffness matrices $\mathbf{K}^{u\alpha}$ and $\mathbf{K}^{\alpha u}$ as well as a fully-enhanced stiffness operator $\mathbf{K}^{\alpha\alpha}$ must be employed. These matrices have the same structure as those defined for the linear formulation and presented by Simo and Rifai [Simo 90b], *i.e.*,

$$\hat{\mathbf{K}}^{u\alpha}=\left(\hat{\mathbf{K}}^{\alpha u}\right)^{\text{T}}=\int_V\hat{\mathbf{B}}^{\text{T}}\mathbf{C}_4\hat{\mathbf{B}}^{\alpha}\text{d}V, \tag{5.32}$$

and

$$\hat{\mathbf{K}}^{\alpha\alpha}=\int_V\left(\hat{\mathbf{B}}^{\alpha}\right)^{\text{T}}\mathbf{C}_4\hat{\mathbf{B}}^{\alpha}\text{d}V. \tag{5.33}$$

Each EAS parameter that is added to a given the Finite Element formulation will increase the number of columns of $\mathbf{B}^{\alpha}$ by one.

Due to the fact that an additive approach is being employed, there is no need to include geometric nonlinear stiffness matrices associated with the enhancing variables, leading to a straightforward algorithmic extension of the linear case [Valente 04b]. The formulation leads to an equivalent system of equations that, in matrix form, can be expressed as

$$\begin{bmatrix}\hat{\mathbf{K}}^{uu}+\hat{\mathbf{K}}^{uu}_{\text{NL}} & \hat{\mathbf{K}}^{u\alpha}\\ \hat{\mathbf{K}}^{\alpha u} & \hat{\mathbf{K}}^{\alpha\alpha}\end{bmatrix}\begin{Bmatrix}{}^{n+1}_{\ \ n}\mathbf{d}\\ {}^{n+1}_{\ \ n}\boldsymbol{\alpha}\end{Bmatrix}=\begin{Bmatrix}\int_V\mathbf{N}^{\text{T}}\mathbf{b}\rho\ \text{d}V+\int_{S_{\text{N}}}\mathbf{N}^{\text{T}}\mathbf{t}\ \text{d}S-\int_V\hat{\mathbf{B}}^{\text{T}}\hat{\mathbf{S}}\ \text{d}V\\ -\int_V\left(\hat{\mathbf{B}}^{\alpha}\right)^{\text{T}}\hat{\mathbf{S}}\ \text{d}V\end{Bmatrix}. \tag{5.34}$$

5.4.2 Subspace Analysis Framework

When developing new EAS-based finite elements, it is important to take into account the number and type of enhancing parameters used. Many authors employed this methodology, but, in some cases, the choice of the enhanced parameters were not fully justified and are mostly based on *trial and error* or inspection. César de Sá and Owen [César de Sá 86] developed the framework of *subspace of deformation* and based on this concept, concluded that volumetric locking occurs when a solution does not appear properly represented in the subspace of approximated incompressible deformations.

Mathematically, a given Finite Element formulation has a characteristic space of admissible deformations with a dimension corresponding to the number of element degrees--of-freedom. However, different subspaces can be defined as a function of the constraints to be considered. A subspace defines the finite set of deformation modes that a single element can represent under general loading and boundary conditions. If it is imposed a deformation pattern to the element which cannot be reproduced by a given combination of the deformation modes from its subspace, then the formulation will suffer from locking effects.

Considering a linear space of admissible solutions U, the idea of the subspace analysis methodology is to determine the displacement field $\mathbf{u}$ that minimizes the energy of the system. The displacement field $\mathbf{u}$ must be contained in a subspace of U. The constraint for isochoric deformations, necessary for instance to model plasticity or some rubber-like material $(\nu \rightarrow 0.5)$ defines a new subspace, here denoted as I. This condition can be expressed as $I \subset U$ for the space of incompressible deformations. Taking U^{h}, I^{h} and $\mathbf{u}^{\mathrm{h}}$ as finite element approximations of U, I and $\mathbf{u}$, respectively, it is then possible to state that

$$\mathbf{Q}\mathbf{u}^{\mathrm{h}} = 0, \tag{5.35}$$

as a possible way to define the subspace of incompressible deformations I^{h}, such as

$$I^{h} = \{\mathbf{u}^{\mathrm{h}} \in U^{\mathrm{h}} : \mathbf{Q}\mathbf{u}^{\mathrm{h}} = 0\}. \tag{5.36}$$

To avoid the trivial solution $\left(\mathbf{u}^{\mathrm{h}} = 0\right)$, the displacement field $\mathbf{u}^{\mathrm{h}}$ should belong to the subspace of incompressible deformations I^{h} or, in other words, $\mathbf{u}^{\mathrm{h}}$ should lie in the nullspace of $\mathbf{Q}$. If this condition is satisfied, then the approximated displacements $\mathbf{u}^{\mathrm{h}}$ will result from a linear combination of a given basis of I^{h} elements. If, under a set of external forces and/or boundary conditions, the solution does not belong to the subspace I^{h} (defined in the previous equation) volumetric locking will occur.

When considering small strains, the incompressibility condition can be written in the three-dimensional space as

$$\left\{ \begin{matrix} \varepsilon_{\xi\xi} \\ \varepsilon_{\eta\eta} \\ \varepsilon_{\zeta\zeta} \end{matrix} \right\} = \int_V \left(\frac{\delta u}{\delta \xi} + \frac{\delta v}{\delta \eta} + \frac{\delta w}{\delta \zeta} \right) \mathrm{d}V = 0. \tag{5.37}$$

By assuring that the integrand function in Equation 5.37 is zero, the incompressibility condition is respected. After a Finite Element discretization, this condition results in

$$\frac{\delta u}{\delta \xi} + \frac{\delta v}{\delta \eta} + \frac{\delta w}{\delta \zeta} = \left[N_{i,\xi} \quad N_{i,\eta} \quad N_{i,\zeta} \right] \mathbf{d}_i = 0, \tag{5.38}$$

where $N_{i,j}$ is the derivative of the shape function N_i with respect to j, while $\mathbf{d}_i$ represents the vector of nodal displacements.

Following the reasoning presented for volumetric locking, it is possible to extend the methodology for the analysis of shear locking in three-dimensional solid elements [Caseiro 13]. Thus, for the subspace analysis of transverse shear locking the transverse shear strain energy must vanish for low thickness values, *i.e.*,

$$\mathbf{Hu}^{\mathrm{h}} = 0, \tag{5.39}$$

where the subspace of transverse shear deformation can be defined as T^{h}, in the form

$$T^{h} = \{\mathbf{u}^{\mathrm{h}} \in U^{\mathrm{h}} : \mathbf{Hu}^{\mathrm{h}} = 0\}. \tag{5.40}$$

Equation 5.39 is a simplified way of imposing that the out-of-plane deformation energy, approximated by FEM, must tend to zero when the element thickness also tends to zero (*i.e.*, Kirchhoff hypothesis). In its discrete form, Equation 5.39 will lead to three sets of equations, that is

$$\begin{Bmatrix} \varepsilon_{\xi\eta} \\ \varepsilon_{\xi\zeta} \end{Bmatrix} = \mathbf{H}^{\xi}\mathbf{u}^{\mathrm{h}} = 0, \tag{5.41}$$

$$\begin{Bmatrix} \varepsilon_{\xi\eta} \\ \varepsilon_{\eta\zeta} \end{Bmatrix} = \mathbf{H}^{\eta}\mathbf{u}^{\mathrm{h}} = 0, \tag{5.42}$$

$$\begin{Bmatrix} \varepsilon_{\xi\zeta} \\ \varepsilon_{\eta\zeta} \end{Bmatrix} = \mathbf{H}^{\zeta}\mathbf{u}^{\mathrm{h}} = 0, \tag{5.43}$$

depending if the normal direction is aligned with the ξ, η or ζ direction, respectively. Therefore, as for the incompressible subspace detailed before, the displacement field $\mathbf{u}^{\mathrm{h}}$ should be contained in the nullspace of $\mathbf{H}$ in order the numerical solution can avoid transverse shear locking effects. Based on an analysis performed for shell elements [César de Sá 02], this corresponds to the condition

$$\frac{1}{2}\begin{bmatrix} 0 & N_{i,\xi} + N_{i,\zeta} & N_{i,\eta} + N_{i,\zeta} \end{bmatrix} \mathbf{d}_i = 0, \tag{5.44}$$

that must be respected for the ζ direction. In an analogous way, the following conditions

$$\frac{1}{2}\begin{bmatrix} N_{i,\xi} + N_{i,\eta} & 0 & N_{i,\eta} + N_{i,\zeta} \end{bmatrix} \mathbf{d}_i = 0, \tag{5.45}$$

and

$$\frac{1}{2}\begin{bmatrix} N_{i,\xi} + N_{i,\eta} & N_{i,\xi} + N_{i,\zeta} & 0 \end{bmatrix} \mathbf{d}_i = 0, \tag{5.46}$$

can also be stated for the η and ξ directions, respectively, which – combined – would apply to general purpose 3D finite elements. Therefore, a shear-locking free solid finite element (*i.e.*, without a preferred thickness direction) can be formulated by assuring that the conditions in Equations 5.44 to 5.46 are simultaneously respected.

5.4.3 The Enhanced Assumed Strain Method for Isogeometric Analysis

The Enhanced Assumed Strain method can be applied to NURBS-based elements using the same procedure as proposed for the standard Lagrangian-based element in Finite Element Analysis. In the work of Cardoso and César de Sá [Cardoso 12], the EAS method was used to alleviate volumetric locking in two-dimensional NURBS-based quadratic elements. The authors employed the subspace methodology to determine the isochoric deformation subspace. It was concluded that using a 3×3 Gaussian integration scheme, the resulting subspace would have a dimension of 10. In order to improve the behaviour of the quadratic NURBS element, six enhanced parameter were then added in order to obtain an isochoric deformation subspace with dimension 16. However, in order to obtain stable results, a stabilization parameter was employed.

In a more comprehensive way and by making use of the subspace analysis framework described in the previous section, it is possible to perform a general analysis for three-dimensional NURBS-based elements and determine the dimension of the subspaces associated with both volumetric and shear locking effects, with the results being presented in Table 5.1. It can be seen that, for the case of the quadratic NURBS-based element, by making use of a $3 \times 3 \times 3$ Gaussian integration scheme, the isochoric deformation subspace will have a dimension of 55. In order to alleviate volumetric locking effects, the isochoric subspace must have a dimension of 80. Therefore, it is necessary to introduce 25 enhancing parameters into the element formulation. Consequently, for each element a 25×25 matrix (corresponding to the $\hat{\mathbf{K}}^{\alpha\alpha}$ stiffness matrix in Equation 5.34) must be inverted. As a consequence, the computational cost of the 3D-EAS formulation will be higher. In addition, the use of such a high number of enhancing parameter would lead to numerical instabilities and the need of using stabilization parameters. A very similar result is obtained for the case of shear locking. It can also been seen from Table 5.1 that the adoption of a lower integration scheme would require the use of a lower number of enhancing parameters. However, these reduced integration schemes may also lead to numerical instabilities.

Initial attempts to provide a high-order NURBS-based three-dimensional solid element employing the EAS method were performed during the current research work. However, due to the high computational costs required and stability issues that plagued the obtained solutions, the implementation was not considered as successful. Nevertheless, the author of this Thesis believes that the extension of the EAS method to IGA may be an interesting future research topic that should be explored, since this methodology is able to alleviate diverse non-physical pathologies such as volumetric, shear and thickness locking. In that sense, the subspace analysis framework presented in the previous section can prove to be a valuable tool.

Table 5.1: Number of deformation modes obtained by different NURBS-based formulations.

Element Order	Integration Scheme	Nullity(Q)	Nullity(H^i)
1	$2 \times 2 \times 2$	17 of 23	18 of 23
1	$1 \times 1 \times 1$	23 of 23	23 of 23
2	$3 \times 3 \times 3$	55 of 80	57 of 80
2	$2 \times 2 \times 2$	73 of 80	73 of 80
2	$1 \times 1 \times 1$	80 of 80	80 of 80
3	$4 \times 4 \times 4$	129 of 191	132 of 191

5.5 The Assumed Natural Strain Method

As mentioned before, the Assumed Natural Strain (ANS) approach was firstly introduced in the works of Hughes and Tezduyar [Hughes 81] and MacNeal [MacNeal 82] in the context of plate elements for linear analysis. The key idea behind the ANS method consists of selecting a set of tying (alternative interpolation) points that will replace the standard integration ones for the calculation of the strain components.

In their work, Dvorkin and Bathe [Dvorkin 84] proposed a general 4-node shell element, leading to the well known MITC4 shell element, where MITC stands for *mixed interpolation of tensorial components*. In the MITC4 only the transverse shear strain components are interpolated in order to alleviate shear locking. In a later work [Bathe 86], the same authors proposed an extension of the formulation to quadratic 8-node shell elements, leading to the MITC8 element. In this element, in addition to the transverse shear strain components, also the in-layer strains were interpolated in order to avoid membrane locking. Bucalem and Bathe [Bucalem 93] further extended this methodology to a 16-node shell element. In the numerical examples presented, the resulting MITC16 formulation did not exhibit shear and membrane locking effects. A variational basis for these proposes can be found, for example, in [Militello 90].

In the years that followed, the ANS methodology was applied in order to alleviate shear and membrane locking in different applications of the FEM, as for example in the works of [Belytschko 94, Hauptmann 98, Sze 00, Cardoso 08, Schwarze 09], among many others.

5.5.1 The ANS Method for Isogeometric Analysis

In the following, a detailed description of the proposed extension of the Assumed Natural Strain method to IGA will be given. With the objective of facilitating the exposition of the methodology, a notation for defining the various frames that slightly differs from the one employed in Chapters 3 and 4 is used. However, the new notation is clearly presented throughout the text and in the accompanying figures.

In the small strain regime, a given strain component at each integration point, in the covariant frame, can be expressed as

$$\varepsilon_{ij}\left(\xi,\eta,\zeta\right)=\frac{1}{2}\left(\frac{\partial\mathbf{u}}{\partial\xi_i}\mathbf{g}_j+\frac{\partial\mathbf{u}}{\partial\xi_j}\mathbf{g}_i\right), \tag{5.47}$$

where $\xi_1=\xi$, $\xi_2=\eta$ and $\xi_3=\zeta$ are the natural coordinate system (described in Chapter 4), while the covariant base vectors are given as $\mathbf{g}_i=\frac{\partial\boldsymbol{x}}{\partial\xi_i}$. Equation 5.47 can be also expressed in matrix form as

$$\boldsymbol{\varepsilon}\left(\xi,\eta,\zeta\right)=\tilde{\mathbf{B}}\left(\xi,\eta,\zeta\right)\mathbf{d}, \tag{5.48}$$

in which $\tilde{\mathbf{B}}\left(\xi,\eta,\zeta\right)$ is the standard compatible strain-displacement matrix in the covariant frame computed at each integration point. In the framework of IGA, $\mathbf{d}$ corresponds to the vector of displacement degrees-of-freedoms at the control points (control variables).

The Choice of Tying Points

The current research work is focused on the quadratic NURBS-based element, which will be employed in the following to present a detailed description of the proposed methodology. Following the original work of Bucalem and Bathe [Bucalem 93] for Lagrangian basis functions, the selection of the tying points for the second-order element is given in Figure 5.3. To define the ANS strain-displacement matrix in the context of IGA, a set of local bivariate basis functions must be created.

The NURBS patch consisting of four second-order elements as presented in Figure 5.4 is considered in order to present the different spaces more clearly. In this figure, the integration points (circles) and the tying points (triangles) for the interpolation of $\varepsilon_{\xi\xi}$ and $\varepsilon_{\xi\zeta}$ strain components in the top left element are represented. The univariate basis functions coming from the knot vectors that define the mesh are denoted as the *global space*. For each element, it is also possible to define two local knot vectors that will be used to define the *local space*. These new knot vectors are open and contain only one non-zero knot span. It is important to note that the basis functions along the ξ-direction is of one order lower than the one along the η-direction, due to the fact that a lower number of tying points is considered in the latter.

As mentioned before, following the work of Bucalem and Bathe [Bucalem 93], the choice of the tying points is closely related to the order of the quadrature employed in the Finite Element formulation. In the current work, following classical 3D solid Lagrangian formulations, a *full integration* scheme is defined when $(p+1)$, $(q+1)$ and $(r+1)$ quadrature points are used in a given element for the ξ, η and ζ-directions, respectively. As can be seen in Figure 5.4, for $\varepsilon_{\xi\xi}$ and $\varepsilon_{\xi\zeta}$ components, the points from a one-order lower Gaussian quadrature are employed in the ξ-direction, while the points corresponding to full Gaussian integration are employed in the η-direction. An analogous reasoning is performed for the $\varepsilon_{\eta\eta}$ and $\varepsilon_{\eta\zeta}$ components of the strain-displacement operator. For the in-plane component $\varepsilon_{\xi\eta}$, the points from a one-order lower Gaussian integration scheme are considered. Experimentations using a lower number of integration points were also performed, leading, however, to the appearance of numerical instabilities and spurious hourglass deformation modes.

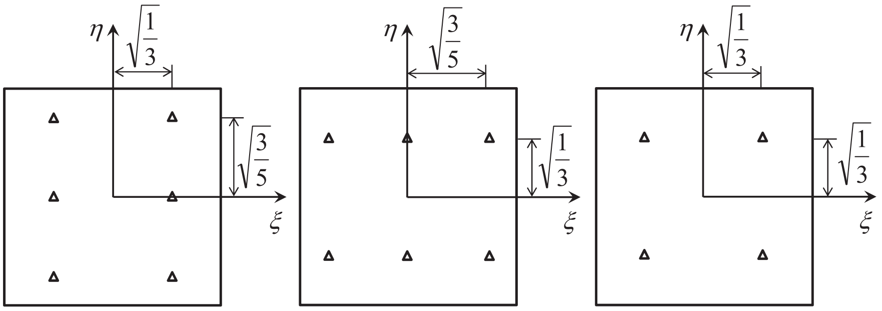


Figure 5.3: Representation of the tying points for the integration of $\varepsilon_{\xi\xi}$ and $\varepsilon_{\xi\zeta}$ (left), $\varepsilon_{\eta\eta}$ and $\varepsilon_{\eta\zeta}$ (centre) and $\varepsilon_{\xi\eta}$ (right).

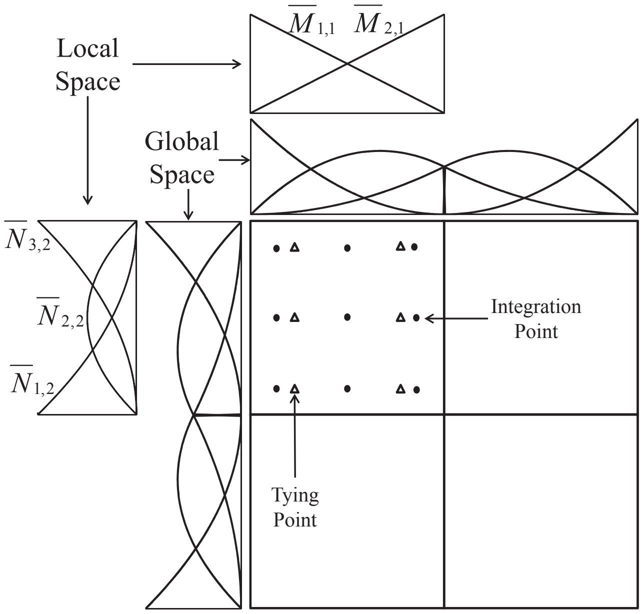


Figure 5.4: Global and local spaces for the quadratic NURBS element (interpolation of $\varepsilon_{\xi\xi}$ and $\varepsilon_{\xi\zeta}$ components).

The Assumed Strain Field

In standard Lagrange-based elements, after computing the strain-displacement matrix at the tying points, a set of interpolation functions are used to associate the tying points with the integration points. This procedure leads, in the end, to assumed covariant strain components. Following the tying point sets defined in Figure 5.3, each component of the assumed strains can then be expressed as

$$\varepsilon_i^{\mathrm{ANS}}(\xi,\eta,\zeta) = \sum_{j=1}^{n_{\mathrm{t}}^i} N_j \bar{\varepsilon}_i\left(\hat{\xi}_j,\hat{\eta}_j,\zeta\right), \tag{5.49}$$

where n_{t}^i is the number of tying points associated with the i^{th} strain component and N_j is the j^{th} component of the vector $\mathbf{N}$ which arises from the tensor product of the local basis functions calculated at each conventional integration point. In the previous equation, $\bar{\varepsilon}_i\left(\hat{\xi},\hat{\eta},\zeta\right)$ are the coefficients of the local space which interpolate the compatible strains at the tying points with coordinates $\left(\hat{\xi},\hat{\eta},\zeta\right)$. Note that the third coordinate follows the one obtained by the quadrature rule being employed. Using the notation presented in Figure 5.4, the vector $\mathbf{N}$ can be expressed as

$$\mathbf{N} = \left[\bar{N}_{3,2}^k\bar{M}_{2,1}^k \quad \bar{N}_{2,2}^k\bar{M}_{2,1}^k \quad \bar{N}_{1,2}^k\bar{M}_{2,1}^k \quad \bar{N}_{3,2}^k\bar{M}_{1,1}^k \quad \bar{N}_{2,2}^k\bar{M}_{1,1}^k \quad \bar{N}_{1,2}^k\bar{M}_{1,1}^k\right]^{\mathrm{T}}, \tag{5.50}$$

where $\bar{N}_{i,p}^k$ and $\bar{M}_{j,q}^k$ are the local univariate NURBS basis functions calculated at the current integration point k. It is then possible to project the local compatible strain $\bar{\boldsymbol{\varepsilon}}\left(\hat{\xi},\hat{\eta},\zeta\right)$ onto the global space, leading now to a global compatible strain $\boldsymbol{\varepsilon}\left(\hat{\xi},\hat{\eta},\zeta\right)$, by performing the following operation

$$\boldsymbol{\varepsilon}\left(\hat{\xi},\hat{\eta},\zeta\right) = \mathbf{M}\,\bar{\boldsymbol{\varepsilon}}\left(\hat{\xi},\hat{\eta},\zeta\right), \tag{5.51}$$

where $\mathbf{M}$, with number of rows and columns equal to the number of tying points, is obtained from the tensor product of the local basis function calculated at each tying point. As an example, this matrix can be computed for the tying point set given in Figure 5.4 as

$$\mathbf{M} = \begin{bmatrix}
\bar{N}_{3,2}^1\bar{M}_{2,1}^1 & \bar{N}_{2,2}^1\bar{M}_{2,1}^1 & \bar{N}_{1,2}^1\bar{M}_{2,1}^1 & \bar{N}_{3,2}^1\bar{M}_{1,1}^1 & \bar{N}_{2,2}^1\bar{M}_{1,1}^1 & \bar{N}_{1,2}^1\bar{M}_{1,1}^1 \\
\bar{N}_{3,2}^2\bar{M}_{2,1}^2 & \bar{N}_{2,2}^2\bar{M}_{2,1}^2 & \bar{N}_{1,2}^2\bar{M}_{2,1}^2 & \bar{N}_{3,2}^2\bar{M}_{1,1}^2 & \bar{N}_{2,2}^2\bar{M}_{1,1}^2 & \bar{N}_{1,2}^2\bar{M}_{1,1}^2 \\
\bar{N}_{3,2}^3\bar{M}_{2,1}^3 & \bar{N}_{2,2}^3\bar{M}_{2,1}^3 & \bar{N}_{1,2}^3\bar{M}_{2,1}^3 & \bar{N}_{3,2}^3\bar{M}_{1,1}^3 & \bar{N}_{2,2}^3\bar{M}_{1,1}^3 & \bar{N}_{1,2}^3\bar{M}_{1,1}^3 \\
\bar{N}_{3,2}^4\bar{M}_{2,1}^4 & \bar{N}_{2,2}^4\bar{M}_{2,1}^4 & \bar{N}_{1,2}^4\bar{M}_{2,1}^4 & \bar{N}_{3,2}^4\bar{M}_{1,1}^4 & \bar{N}_{2,2}^4\bar{M}_{1,1}^4 & \bar{N}_{1,2}^4\bar{M}_{1,1}^4 \\
\bar{N}_{3,2}^5\bar{M}_{2,1}^5 & \bar{N}_{2,2}^5\bar{M}_{2,1}^5 & \bar{N}_{1,2}^5\bar{M}_{2,1}^5 & \bar{N}_{3,2}^5\bar{M}_{1,1}^5 & \bar{N}_{2,2}^5\bar{M}_{1,1}^5 & \bar{N}_{1,2}^5\bar{M}_{1,1}^5 \\
\bar{N}_{3,2}^6\bar{M}_{2,1}^6 & \bar{N}_{2,2}^6\bar{M}_{2,1}^6 & \bar{N}_{1,2}^6\bar{M}_{2,1}^6 & \bar{N}_{3,2}^6\bar{M}_{1,1}^6 & \bar{N}_{2,2}^6\bar{M}_{1,1}^6 & \bar{N}_{1,2}^6\bar{M}_{1,1}^6
\end{bmatrix}, \tag{5.52}$$

where $\bar{N}_{i,p}^t$ and $\bar{M}_{j,q}^t$ are the local univariate NURBS basis functions calculated at the tying point t. Matrix $\mathbf{M}$ presented in Equation 5.52 is computed using the local basis functions at the tying point coordinates, which are the same for each element of the patch. Consequently, this matrix needs only to be computed once for each tying point set at the beginning of the analysis, leading to lower computational costs.

Combining Equations 5.49 and 5.51 leads to the final form of the assumed natural strain field as

$$\varepsilon_i^{\mathrm{ANS}}(\xi,\eta,\zeta) = \sum_{j=1}^{n_{\mathrm{t}}^i} L_j \varepsilon_i\left(\hat{\xi}_j,\hat{\eta}_j,\zeta\right), \tag{5.53}$$

where L_j is the j^{th} component of the vector $\mathbf{L} = \mathbf{N}^{\mathrm{T}}\mathbf{M}^{-1}$. For the numerical implementation in the Isogeometric Analysis framework, the previous equation can be written in terms of strain-displacement operators as

$$\tilde{\mathbf{B}}_i^{\mathrm{ANS}}(\xi,\eta,\zeta) = \sum_{j=1}^{n_{\mathrm{t}}^i} L_j \tilde{\mathbf{B}}_i\left(\hat{\xi}_j,\hat{\eta}_j,\zeta\right), \tag{5.54}$$

where $\tilde{\mathbf{B}}_i$ corresponds to the i^{th} line of the $\tilde{\mathbf{B}}$ matrix. The numerical implementation of the ANS procedure implies the substitution of lines of the compatible strain-displacement operator by the ones coming from the ANS strain-displacement operator, which were computed in the associated tying points. Note that, in accordance with the tying point sets given in Figure 5.3, the third line of the strain-displacement operator, corresponding to the $\varepsilon_{\zeta\zeta}$ strain component, remains unchanged, *i.e.*, is the same as in the compatible strain-displacement matrix.

The interpolation based on the tying points, for the NURBS-based formulation, is independent of the element-based (natural) ζ coordinate. This is typical for shell formulations, and is adopted in the present work for trivariate NURBS constructions, thus justifying the so-called *solid-shell* concept.

The extension of the ANS methodology to the nonlinear regime is straightforward. Once the ANS strain-displacement operator in the corotational frame $\hat{\mathbf{B}}^{\mathrm{ANS}}$ is computed it will replace the standard strain-displacement operator in the computation of the strain components and for the tangential stiffness matrix $\hat{\mathbf{K}}$. It should be noted that, when accounting for geometric nonlinearities, the corresponding stiffness matrix $\hat{\mathbf{K}}_{\mathrm{NL}}$ (given by Equation 4.48) remains unchanged.

In Box 5.2, the algorithm to obtain the Assumed Natural Strain strain-displacement operator $\mathbf{B}^{\mathrm{ANS}}$ is presented. A detailed algorithm describing the implementation of the ANS methodology for NURBS-based elements in the nonlinear regime is presented in Box 5.3.

Box 5.2: Algorithm for the computation of the Assumed Natural Strain strain-displacement operator.

```
1. Compute the strain-displacement operator in the covariant frame for the current integration point $\tilde{\mathbf{B}}(\xi,\eta,\zeta)$
2. DO tying point set cycle
    (a) Compute $\mathbf{N}$ based on the local basis functions and the integration points coordinates (Equation 5.50)
    (b) Compute $\mathbf{M}$ based on the local basis functions and the tying points coordinates (Equation 5.52)
    (c) DO tying point cycle
        i. Compute the strain-displacement operator in the covariant frame at the tying points coordinates $\tilde{\mathbf{B}}\left(\hat{\xi},\hat{\eta},\zeta\right)$
        ii. Compute the ANS strain-displacement operator $\tilde{\mathbf{B}}^{\mathrm{ANS}}(\xi,\eta,\zeta)$ (Equation 5.54)
    (d) END DO
    (e) Replace the appropriate line of $\tilde{\mathbf{B}}(\xi,\eta,\zeta)$ by the ones from $\tilde{\mathbf{B}}^{\mathrm{ANS}}(\xi,\eta,\zeta)$
3. END DO
```

Note: The operators **N** and **M** can be precomputed outside the element cycle for lower computational costs. In this algorithm, they are included in order to more clearly present the developed approach.

Box 5.3: Algorithm for the Assumed Natural Strain method for NURBS-based formulations (should be used in conjunction with the algorithm from Box 4.2).

```
1. DO element cycle
    (a) Initialise elemental stiffness matrix and internal force vector
    (b) Recover the local axis ${}^{n}\mathbf{r}$ from the last converged increment
    (c) DO integration point cycle
        i. Calculate the deformation gradient for the mid-point ${}^{n+\frac{1}{2}}_{n}\mathbf{F}$ and end ${}^{n+1}_{n}\mathbf{F}$ configurations
        ii. Use the polar decomposition algorithm in Box 4.1 to obtain the rotation matrices ${}^{n+\frac{1}{2}}_{n}\mathbf{R}$ and ${}^{n+1}_{n}\mathbf{R}$
        iii. Update the corotational reference system as
                 $${}^{n+\frac{1}{2}}\mathbf{r} = {}^{n+\frac{1}{2}}_{n}\mathbf{R}\,{}^{n}\mathbf{r}$$
                 $${}^{n+1}\mathbf{r} = {}^{n+1}_{n}\mathbf{R}\,{}^{n}\mathbf{r}$$
        iv. Compute the compatible strain-displacement matrix in the covariant frame for both mid--point ${}^{n+\frac{1}{2}}\tilde{\mathbf{B}}$ and end ${}^{n+1}\tilde{\mathbf{B}}$ configurations in the current integration point
        v. Compute ${}^{n+\frac{1}{2}}\tilde{\mathbf{B}}^{\mathrm{ANS}}$ and ${}^{n+1}\tilde{\mathbf{B}}^{\mathrm{ANS}}$ using the algorithm presented in Box 5.2
        vi. Use the natural-to-local transformation operator $\mathbf{T}$ (Equations 4.54 and 4.56) to obtain the strain-displacement operators in the corotational coordinate system denoted as ${}^{n+\frac{1}{2}}\hat{\mathbf{B}}^{\mathrm{ANS}}$ and ${}^{n+1}\hat{\mathbf{B}}^{\mathrm{ANS}}$
        vii. Compute the stress and strain fields using ${}^{n+\frac{1}{2}}\hat{\mathbf{B}}^{\mathrm{ANS}}$ and the elastic/elastoplastic constitutive tensor $\hat{\mathbf{C}}_4$
        viii. Compute stiffness matrix $\hat{\mathbf{K}}$ (Equation 4.47) and the internal forces $\hat{\mathbf{f}}^{\mathrm{int}}$ (Equation 4.49) using ${}^{n+1}\hat{\mathbf{B}}^{\mathrm{ANS}}$
        ix. Compute the geometric nonlinear stiffness matrix $\hat{\mathbf{K}}_{\mathrm{NL}}$ (Equation 4.48) and add this contribution to the elemental stiffness
        x. Store the local axis in the end configuration ${}^{n+1}\mathbf{r}$ to be used in the next increment
    (d) END DO
2. END DO
```

5.5.2 The Proposed H2ANS Formulation

When developing new Finite Element formulations, it is important to take into account its computational efficiency. From the literature survey performed at the beginning of the current chapter, it can be concluded that a great amount of research effort was dedicated in the improvement of the performance and treatment of locking pathologies in low-order Lagrangian-based finite elements. One of the main advantages of these Finite Element formulations is the fact that they usually present a reduced computational cost, especially when employing reduced integration techniques.

In the context of Isogeometric Analysis, it is well known that the linear NURBS-based formulation provides exactly the same results as standard Lagrangian-based formulations. Consequently, the lowest order formulation that can take advantage of the NURBS basis functions corresponds to the development of a quadratic element. In addition, since it is possible to use quadratic NURBS-based elements to represent most of the usual shapes studied [Piegl 97, Cottrell 09, Bouclier 13a], it seems to be of great importance to invest some research effort to improve the performance of such formulations.

This is the motivation which led to the development of a quadratic NURBS-based solid-shell element extending the concept of the Assumed Natural Strain to IGA. The proposed formulation will be denoted as H2ANS from hereafter and all the details for its implementation were given in the previous section.

It is worth mentioning that the procedure to implement the ANS method in NURBS--based formulations presented herein is entirely performed at the element level. As a consequence, this strategy would allow for an easier implementation within available commercial finite element codes in combination with a Bézier extraction approach in a similar way, as carried out by Borden *et al.* [Borden 11]. The presented formulation can also be extended to higher-order solid-shell elements in a straightforward manner. One only needs to define the tying points coordinates accordingly to the degree of the element under consideration and the strain component being interpolated. The computation of the vector **N** and matrix **M** is simply obtained from the tensor product of the local basis functions, as detailed before. However, studies performed during the current research work demonstrated that no significant gains were obtained when applying the ANS methodology to cubic NURBS-based elements. Nevertheless, the development of different element-based quadrature rules which are optimal for IGA may open up the possibility of extending the proposed ANS methodology to these higher-order formulations.

5.6 A Note on the Alleviation of Volumetric Locking

The Assumed Natural Strain methodology was originally devised in the context of plate and shell elements to alleviate pathological effects such as shear and membrane locking. Thus, the ANS method is not adequate to solve or attenuate volumetric locking. In the literature, this material-based locking is usually dealt with by employing, for instance, the EAS method, the $\bar{\mathbf{B}}$ or $\bar{\mathbf{F}}$ techniques.

In the following it is shown that the methodology described in Section 5.5 can be employed to alleviate volumetric locking effects in NURBS-based formulations. The starting point is based on the $\bar{\mathbf{B}}$ approach, originally proposed by Hughes [Hughes 80] for the analysis of nearly-incompressible media in FEM.

From the equations dealt with in the previous chapters, the strain field can be expressed in terms of the strain-displacement operator $\mathbf{B}$ as

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{d},$$

which can be written as

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \dots & \mathbf{B}_{n_\mathrm{n}} \end{bmatrix},$$

where n_n is, in the context of IGA, the number of control points of the element. Each sub-matrix $\mathbf{B}_i$ in the previous equation can also be expressed as

$$\mathbf{B}_i = \begin{bmatrix} R_{i,x} & 0 & 0 \\ 0 & R_{i,y} & 0 \\ 0 & 0 & R_{i,z} \\ R_{i,y} & R_{i,x} & 0 \\ R_{i,z} & 0 & R_{i,y} \\ 0 & R_{i,z} & R_{i,y} \end{bmatrix}. \tag{5.55}$$

Furthermore, the strain-displacement operator $\mathbf{B}$ can be additively decomposed into its volumetric $\mathbf{B}^{\mathrm{vol}}$ and deviatoric $\mathbf{B}^{\mathrm{dev}}$ contributions as

$$\mathbf{B} = \mathbf{B}^{\mathrm{vol}} + \mathbf{B}^{\mathrm{dev}}, \tag{5.56}$$

which are given by the sub-matrices

$$\mathbf{B}_i^{\mathrm{vol}} = \frac{1}{3} \begin{bmatrix} R_{i,x} & R_{i,y} & R_{i,z} \\ R_{i,x} & R_{i,y} & R_{i,z} \\ R_{i,x} & R_{i,y} & R_{i,z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.57}$$

and

$$\mathbf{B}_i^{\text{dev}} = \frac{1}{3}\begin{bmatrix} 2R_{i,x} & -R_{i,y} & -R_{i,z} \\ -R_{i,x} & 2R_{i,y} & -R_{i,z} \\ -R_{i,x} & -R_{i,y} & 2R_{i,z} \\ 3R_{i,y} & 3R_{i,x} & 0 \\ 3R_{i,z} & 0 & 3R_{i,y} \\ 0 & 3R_{i,z} & 3R_{i,y} \end{bmatrix}, \tag{5.58}$$

respectively. To obtain an improved performance when dealing with incompressible problems where volumetric locking can be an issue, the $\mathbf{B}^{\text{vol}}$ matrix can be replaced by an improved volumetric contribution $\bar{\mathbf{B}}^{\text{vol}}$ (which must be formally identical), leading to the new strain-displacement operator as

$$\bar{\mathbf{B}} = \bar{\mathbf{B}}^{\text{vol}} + \mathbf{B}^{\text{dev}}. \tag{5.59}$$

The methodology proposed in Section 5.5 can be promptly used to compute the improved volumetric strain-displacement operator $\bar{\mathbf{B}}^{\text{vol}}$. Once again, this study will be focused on quadratic elements dealt with a full integration rule consisting of $(p+1)\times(q+1)\times(r+1)$ integration points. A new tying point scheme is selected, where the location of these points are given by a reduced Gaussian integration scheme. A schematic representation of these tying points can be seen in Figure 5.5. These points will then be used to compute the components of $\bar{\mathbf{B}}^{\text{vol}}$ using the same procedure as the one presented in Section 5.5.1. The performance of this methodology is assessed in Section 7.3 using two numerical examples in the linear elastic range, where the newly proposed formulation is denoted as H2PV.

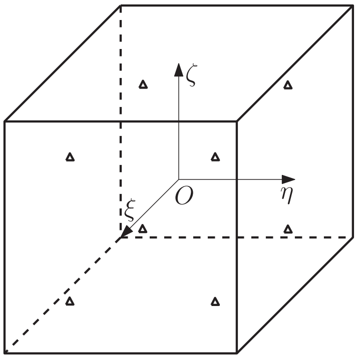


Figure 5.5: Representation of the tying points (triangles) for the computation of the $\bar{\mathbf{B}}^{\text{vol}}$ matrix.

Chapter 6

Contact for Isogeometric Analysis

A brief state-of-the-art review of the main developments in the context of contact mechanics for Isogeometric Analysis is presented. The description of a general two-dimensional frictionless contact problem is given, followed by a detailed description of the Point-to-Segment algorithm where special attention is provided to the main aspects of the implementation procedure.

In the context of the Finite Element Method, a contact problem involving large sliding can be seriously affected by numerical instabilities which are often associated with non-smooth contact surface discretizations. This problem can be tackled by employing Hermite, spline or Bézier interpolations to discretize the master surface. Relevant contributions in this field can be found in the work of Pietrzak and Curnier [Pietrzak 99], Wriggers *et al.* [Wriggers 01], Krstulovic-Opara *et al.* [Krstulovic-Opara 02] and Stadler *et al.* [Stadler 03].

Within an Isogeometric Analysis, and since NURBS are used to describe the geometry of the problem under consideration, the surface description is already available and, therefore, no smoothing procedures are required. Lu [Lu 11] introduced a NURBS Isogeometric formulation for frictionless contact and concluded that this discretization alleviates the non-physical contact force oscillations often detected in contact with faceted surfaces. In a simultaneous parallel study, Temizer and co-workers [Temizer 11] proposed a Knot-to-Surface (KTS) algorithm as an extension of the classical Node-to-Surface algorithm. Although the KTS algorithm led to satisfactory qualitative results in various examples, it delivered excessively stiff contact constraints enforcement. To alleviate this issue, a mortar KTS approach was also developed, which was able to attain robust and accurate results. In a later work, the same authors [Temizer 12] extended their previous contribution to the large deformation regime using a 3D mortar-based frictional contact treatment. The proposed approach presented robust local results even when considering coarse meshes, leading to smooth pressure and tangential traction distributions. Dittmann and co-authors [Dittmann 14] proposed an extension to a fully coupled thermomechanically consistent frictional mortar contact formulation suitable for the analysis of contact/impact problems, allowing to model the energy transfer between the contact surfaces.

De Lorenzis *et al.* [de Lorenzis 11] proposed a 2D contact formulation based on a mortar approach for normal and frictional contact, combined with a simple integration scheme in the large deformations regime. Results showed that the proposed methodology presents a significantly superior performance, when compared with Lagrange discretizations. It was also shown that in large frictional sliding problems, the tractions histories obtained are much smoother. In a later work, de Lorenzis *et al.* [de Lorenzis 12] employed a mortar-based approach in combination an Augmented Lagrangian method to solve large deformation frictionless problem in 3D analysis. The authors demonstrated that the NURBS-based approach can lead to significantly better predictions of the contact pressures, while Lagrangian ones present spurious oscillations and, in some cases, non-physical negative values.

Kim and Youn [Kim 12] proposed a novel contact matching algorithm for linear elastic frictionless Isogeometric Analysis contact problems using a mortar method. The employed methodology resulted in an excellent performance for curved contact surface problems with nonconforming meshes.

The use of T-Splines for modelling contact presents an advantage over NURBS-based formulations since T-Spline interpolations are able to represent complex geometries with a single parametrisation. Following this reasoning, Dimitri *et al.* [Dimitri 14] employed T-Splines to model two- and three-dimensional frictionless contact problems between deformable bodies in the large deformation regime. The problem was solved employing a Gauss-Point-to-Surface (GPTS) method, while the frictionless contact constraints were regularized by the penalty method. T-Splines and NURBS presented similar orders of convergence although T-Splines shows a superior accuracy for a given number of degrees-of-freedom. However, due to the high number of locations at which the contact constraints are enforced, numerical instabilities can occur when these constrains are enforced exactly or nearly exactly. Consequently, the GPTS algorithm should not be used in conjunction with the Lagrange Multiplier method or with the penalty method when very large values of the penalty parameter are considered.

Matzen *et al.* [Matzen 13] developed a Point-to-Segment (PTS) algorithm as an straightforward extension of the Node-to-Segment (NTS) algorithm used in two-dimensional analysis. In this formulation, a set of collocation points is defined in the slave segment in order to collocate the contact integrals. The numerical examples demonstrated that the NURBS-based PTS algorithms present superior performance in large sliding contact problems when compared with Lagrange discretizations.

In this work, an introductory study of contact mechanics in the context of Isogeometric

Analysis is performed. In the following, the PTS algorithm developed by Matzen and co-workers [Matzen 13] is described in detail, along with its numerical implementation. This chapter serves as the starting point for the analysis of problems in the field of contact mechanics within the research group in which the author of this Thesis is inserted. Therefore, an effort was carried out to develop a program which can be used in the future for further developments and implementations in contact mechanics using IGA.

6.1 Normal Contact in Two Dimensions

Consider that two deformable bodies, that occupy distinct positions in the initial configuration, come into contact after a finite deformation process, as seen in Figure 6.1. These bodies can be classified as master and slave, according to the upperscript index $i = 1$ and $i = 2$, respectively. The position vectors of a material point belonging to the master and slave bodies in the current configuration are given as $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively.

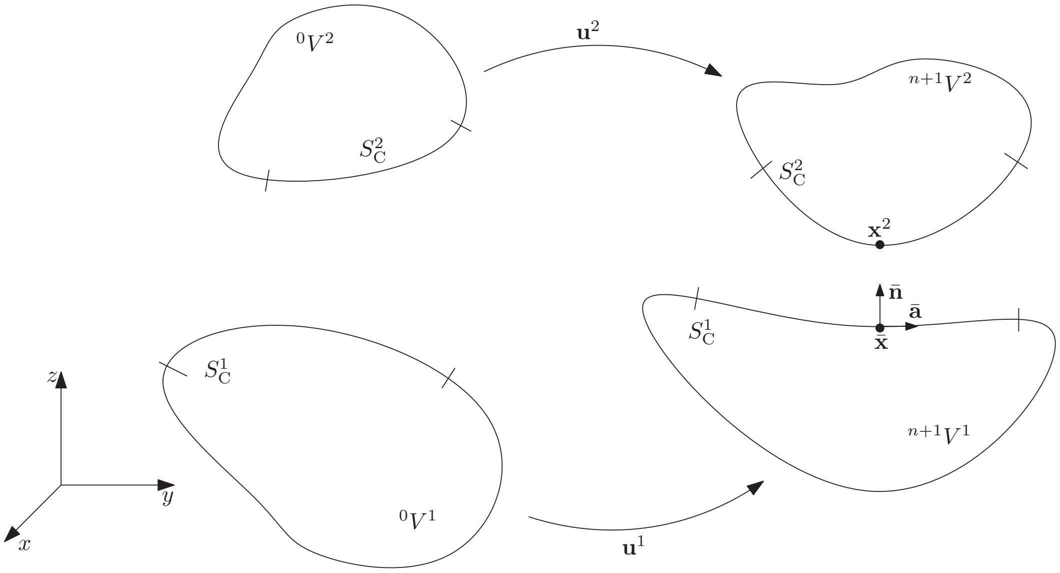


Figure 6.1: Finite deformation of bodies in a contact problem.

In order to determine the gap between the two bodies, it is required to determine the Closest Point Projection (CPP) of the slave point $\mathbf{x}^2$ onto the master contact segment S^1_{C}, which can be mathematically expressed as

$$\bar{\mathbf{x}} = \min_{\mathbf{x}^1 \subseteq S^1_{\mathrm{C}}} ||\mathbf{x}^2 - \mathbf{x}^1(\xi)||, \tag{6.1}$$

where ξ is the parametric coordinate of the contact boundary of the master body S^1_{C}. Quantities with an overbar $\overline{(\cdot)}$ are evaluated at the CPP in the parametric space ξ.

Once the projection $\bar{\mathbf{x}}$ is known, the normal gap can be defined as

$$g_{\mathrm{N}} = \left(\mathbf{x}^2 - \bar{\mathbf{x}}\right) \cdot \bar{\mathbf{n}}, \tag{6.2}$$

where $\bar{\mathbf{n}}$ is the outward unit normal on the current master segment at the CPP $\bar{\mathbf{x}}$. Contact takes place when $g_\mathrm{N} = 0$, leading to the appearance of a normal contact pressure $p_\mathrm{N} < 0$. In the case of frictional contact a tangential pressure is also present, but this scenario will not be considered in the current work. On the other hand, if there exists a gap between the bodies, then $g_\mathrm{N} \geq 0$ and $p_\mathrm{N} = 0$. Therefore, the contact conditions can be stated as

$$g_\mathrm{N} \geq 0, \tag{6.3}$$

$$p_\mathrm{N} \leq 0, \tag{6.4}$$

$$g_\mathrm{N} p_\mathrm{N} = 0, \tag{6.5}$$

which are known as the Hertz-Signorini-Moreu (or alternatively, the Karush-Kuhn-Tucker) conditions for frictionless contact.

The variation of the normal gap δg_N follows from Equation 6.2 as

$$\delta g_\mathrm{N} = \delta \left[\left(\mathbf{x}^2 - \bar{\mathbf{x}} \right) \cdot \bar{\mathbf{n}} \right], \tag{6.6}$$

which leads to

$$\delta g_\mathrm{N} = \left(\delta \mathbf{x}^2 - \delta \bar{\mathbf{x}} - \bar{\mathbf{x}}_{,\xi} \delta \xi \right) \cdot \bar{\mathbf{n}} + \left(\mathbf{x}^2 - \bar{\mathbf{x}} \right) \cdot \delta \bar{\mathbf{n}}. \tag{6.7}$$

6.2 Description of the Frictionless Contact Problem

The Principle of Virtual Work (PVW) for each body i, and neglecting inertia terms, can be expressed as

$$\delta \Pi \left(\mathbf{u}, \delta \mathbf{u} \right) = \sum_{i=1}^{2} \int_{V^i} \mathbf{S}^i : \delta \mathbf{E}^i \mathrm{d}V - \int_{V^i} \rho \mathbf{b}^i \cdot \delta \mathbf{u}^i \mathrm{d}V - \int_{S_\mathrm{N}^i} \mathbf{t}^i \cdot \delta \mathbf{u}^i \mathrm{d}S = 0. \tag{6.8}$$

where the first term corresponds to virtual internal work, while the second and third terms are related to the virtual work of the external forces. Thus, Equation 6.8 can be re-written as

$$\delta \Pi \left(\mathbf{u}, \delta \mathbf{u} \right) = \sum_{i=1}^{2} \delta \Pi_\mathrm{int}^i + \delta \Pi_\mathrm{ext}^i = 0, \tag{6.9}$$

where

$$\delta \Pi_\mathrm{int}^i = \int_{V^i} \mathbf{S}^i : \delta \mathbf{E}^i \mathrm{d}V, \tag{6.10}$$

and

$$\delta \Pi_\mathrm{ext}^i = - \int_{V^i} \rho \mathbf{b}^i \cdot \delta \mathbf{u}^i \mathrm{d}V - \int_{S_\mathrm{N}^i} \mathbf{t}^i \cdot \delta \mathbf{u}^i \mathrm{d}S. \tag{6.11}$$

Once the contact interface is known, an additional term must be added to Equation 6.9 in order to avoid penetration of the bodies. The PVW including the term dealing with the contact contributions can then be written as

$$\sum_{i=1}^{2} \delta \Pi_\mathrm{int}^i + \delta \Pi_\mathrm{ext}^i + \delta \Pi_\mathrm{C} = 0, \tag{6.12}$$

where $\delta\Pi_C$ is associated with the active master-slave contact boundaries set.

There are different strategies that can be applied to solve the contact problem. For example, the penalty and the Lagrange Multiplier methods, often used in optimisation theory, can be employed. In the penalty method a contact penalty is added to the active contact constrains by means of a large penalty parameter. This methodology is easy to implement, however it only approximates the solution of the problem and, additionally, the use of large penalty parameters can lead to an ill-conditioned numerical problem. In the Lagrange Multiplier method the contact constraints are fulfilled in an exact matter, but at the expense of additional variables. In the current work, the Lagrange Multiplier method is applied to solve the contact problem.

6.2.1 The Lagrange Multiplier Method

The Lagrange Multiplier method is employed in optimization theory to determine a minimum (or maximum) of a constrained functional Π. Mathematically, the minimisation of the scalar functional $\Pi(\mathbf{x})$ under constraint $g(\mathbf{x})$ can be expressed as

$$\min \Pi(\mathbf{x}), \tag{6.13}$$

subjected to

$$g(\mathbf{x}) = 0. \tag{6.14}$$

Using the Lagrange Multiplier method, the constrained minimization problem can be reformulated as a saddle point problem by employing the Lagrange functional as

$$\operatorname{grad}\left(\mathscr{L}\left(\mathbf{x},\lambda\right)\right) = 0, \tag{6.15}$$

where λ is known as the Lagrange multiplier. The Lagrangian is constructed as

$$\mathscr{L}\left(\mathbf{x},\lambda\right) = \Pi\left(\mathbf{x}\right) + \lambda g(\mathbf{x}),\ \lambda \leq 0, \tag{6.16}$$

and its gradient is given as

$$\operatorname{grad}\left(\mathscr{L}\left(\mathbf{x},\lambda\right)\right) = \begin{bmatrix} \frac{\partial \mathscr{L}}{\partial \mathbf{x}} \\ \frac{\partial \mathscr{L}}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} \frac{\partial \Pi(\mathbf{x})}{\partial \mathbf{x}} + \lambda \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \\ g(\mathbf{x}) \end{bmatrix} = 0. \tag{6.17}$$

The lower equation is the constrain function given in Equation 6.14. The replacement of a single argument functional $\Pi(\mathbf{x})$ by the two argument Lagrange functional $\mathscr{L}\left(\mathbf{x},\lambda\right)$ implies a higher number of unknowns in the latter [Yastrebov 13].

The Lagrange Multiplier Method for Normal Contact

Using the Lagrange Multiplier method, the contact contribution Π_{C} in Equation 6.12 can be defined for normal contact as

$$\Pi_{\mathrm{C}}^{\mathrm{LM}} = \int_{S_{\mathrm{C}}} \lambda_{\mathrm{N}} g_{\mathrm{N}} \mathrm{d}S, \tag{6.18}$$

where, once again, λ_{N} is the Lagrange multiplier which can be interpreted as the normal contact pressure in the contact interface. The variation of Π_{C} leads to

$$\delta \Pi_{\mathrm{C}}^{\mathrm{LM}} = \int_{S_{\mathrm{C}}} \lambda_{\mathrm{N}} \delta g_{\mathrm{N}} \mathrm{d}S + \int_{S_{\mathrm{C}}} \delta \lambda_{\mathrm{N}} g_{\mathrm{N}} \mathrm{d}S, \tag{6.19}$$

where the first term is associated with the virtual work of the Lagrange multipliers along the variation of the normal gap function. The second term in Equation 6.19 represents the enforcement of the contact constraints.

6.3 Point-to-Segment Contact Formulation

In the following, the Point-to-Segment (PTS) contact formulation proposed by Matzen and co-workers [Matzen 13] is described. This formulation can be seen as an extension to NURBS-based formulations of the classic Node-to-Segment (NTS) algorithm frequently used in the context of FEM. In the NTS formulation, the non-penetration conditions are enforced by preventing that the nodes on the slave segment penetrate the master segments. Due to its simplicity, clear physical meaning and flexibility, the NTS formulation is widely used in problems involving contact. For details on the NTS algorithms, the reader is referred to [Hughes 76, Wriggers 85, Papadopoulos 92, Zavarise 09b], and references therein.

In the PTS algorithm, a set of points on the slave segment must be defined. These are known as *collocation points* and will be denoted as $\mathbf{x}_s$ in the following Sections. The need to define collocation points arises from the fact that, contrary to standard Finite Element formulations, the control points are not, in general, interpolatory and, therefore, are not part of the geometry. In the following, the term $\bar{\mathbf{x}}$ will be used to define the CPP on the master curve, while $\mathbf{x}_i^1$ will denote the control point i of the master segment.

6.3.1 Kinematics

Consider a discrete collocation point $\mathbf{x}_{\mathrm{S}}$ belonging to the slave curve. The normal gap g_{N} can be defined as the minimum distance between the slave point and the master segment as

$$g_{\mathrm{N}} = (\mathbf{x}_{\mathrm{S}} - \bar{\mathbf{x}}) \cdot \bar{\mathbf{n}}, \tag{6.20}$$

where $\bar{\mathbf{x}}$ is the CPP of $\mathbf{x}_\mathrm{s}$ into the master segment, with outward unit normal $\bar{\mathbf{n}}$, as depicted in Figure 6.2. The tangent vector at the CPP can be computed as

$$\bar{\mathbf{a}} = \frac{1}{l}\sum_{i=1}^{n_\mathrm{m}} R_{i,\xi}\left(\bar{\xi}\right)\mathbf{x}_i^1 = \frac{1}{l}\sum_{i=1}^{n_\mathrm{m}} \bar{R}_{i,\xi}\mathbf{x}_i^1, \tag{6.21}$$

where

$$l = \left|\sum_{i=1}^{n_\mathrm{m}} \bar{R}_{i,\xi}\mathbf{x}_i^1\right|. \tag{6.22}$$

In Equations 6.21 and 6.22, n_m is the number of basis functions, including vanishing terms, on the master curve of order p_m and $\bar{R}_{i,\xi}$ are the derivatives of the master basis function i computed at the CPP with respect to ξ. It is now possible to define a local frame $(\bar{\mathbf{n}}, \bar{\mathbf{a}}, \mathbf{e}_3)$, where $\mathbf{e}_3$ is the unit vector orthogonal to the plane containing the contact element. For the two-dimensional case, the unit normal vector can then be obtained as

$$\bar{\mathbf{n}} = \bar{\mathbf{a}} \times \mathbf{e}_3. \tag{6.23}$$

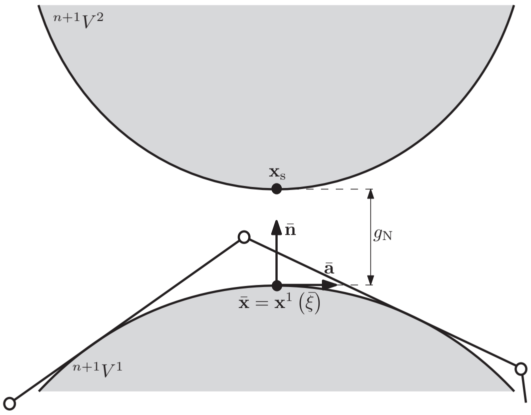


Figure 6.2: Point-to-Segment contact element.

The coordinates of the projection of the slave point $\mathbf{x}_\mathrm{s}$ on the master segment are given as

$$\mathbf{x}\left(\bar{\xi}\right) = \bar{\mathbf{x}} = \sum_{i=1}^{n_\mathrm{m}} \bar{R}_i\mathbf{x}_i^1, \tag{6.24}$$

which can be computed using a numerical iterative procedure on

$$\left(\mathbf{x}_\mathrm{s} - \bar{\mathbf{x}}\right)\cdot\bar{\mathbf{a}} = 0. \tag{6.25}$$

This equation guarantees that orthogonality between the vectors $\left(\mathbf{x}_\mathrm{s} - \bar{\mathbf{x}}\right)$ and $\bar{\mathbf{a}}$ is achieved. The variation of the gap can now be obtained from Equation 6.7 by taking into account that

$$\bar{\mathbf{x}}_{,\xi}\cdot\bar{\mathbf{n}} = 0, \tag{6.26}$$

and

$$(\mathbf{x}_\mathrm{s} - \bar{\mathbf{x}}) = 0, \tag{6.27}$$

leading to

$$\delta g_\mathrm{N} = (\delta \mathbf{x}_\mathrm{s} - \delta \bar{\mathbf{x}}) \cdot \bar{\mathbf{n}}. \tag{6.28}$$

6.3.2 Choice of Contact Collocation Points

In the literature, there are various sets of collocation points that can be selected, such as Greville [De Boor 78], Demko [Demko 85] or Botella [Botella 02] points. In the current work, Greville points are considered to collocate the contact integrals. The coordinate of the Greville points can be obtained as

$$\breve{\xi}_i = \frac{\xi_{i+1} + \cdots + \xi_{i+p+1}}{p}, \tag{6.29}$$

where ξ_i are the knots contained in the knot vector $\boldsymbol{\Xi}$ which defines the curve of degree p. According to Matzen and co-workers [Matzen 13], the choice of the collocation points is motivated by two main reasons: (i) Demko points have to be computed by a complex iterative algorithm, while Greville and Botella points can be more easily obtained; and (ii) the number of Greville and Botella points are the same as the number of control points used to define the surface. The latter presents an advantage since a higher number of collocation points results in a over-constrained system which can cause convergence problems. Moreover, results show that Greville abscissae present better results when compared with Botella points.

In the remainder of this work, the upperscript $\breve{(\cdot)}$ is used to denote variables computed at the collocation point $\mathbf{x}_\mathrm{s}$. Thus, the coordinates of the contact collocation point $\mathbf{x}_\mathrm{s}$ can be obtained as

$$\mathbf{x}_\mathrm{s} = \sum_{i=1}^{n_\mathrm{s}} R_i \left(\breve{\xi}\right) \mathbf{x}^2 = \breve{R}_i \mathbf{x}^2, \tag{6.30}$$

where n_s is the number of control points that define the slave curve of order p_s.

6.3.3 Linearisation

When using a Newton-Raphson iterative scheme, it is required to linearise the contact contributions is order to obtain a quadratic convergence. The normal contact term given in Equation 6.19 can be rewritten as

$$\delta \Pi_\mathrm{C}^\mathrm{LM} = \int_{S_\mathrm{C}} c_\mathrm{N} \mathrm{d}S, \tag{6.31}$$

where

$$c_\mathrm{N} = \lambda_\mathrm{N} \delta g_\mathrm{N} + \delta \lambda_\mathrm{N} g_\mathrm{N}. \tag{6.32}$$

The linearisation of c_N can be computed as

$$\frac{\partial c_\mathrm{N}}{\partial \mathbf{u}}\Delta\mathbf{u} = \frac{\partial c_\mathrm{N}}{\partial \lambda_\mathrm{N}}\Delta\lambda_\mathrm{N} + \frac{\partial c_\mathrm{N}}{\partial g_\mathrm{N}}\Delta g_\mathrm{N} + \frac{\partial c_\mathrm{N}}{\partial \delta g_\mathrm{N}}\Delta\delta g_\mathrm{N} + \frac{\partial c_\mathrm{N}}{\partial \delta\lambda_\mathrm{N}}\Delta\delta\lambda_\mathrm{N}, \tag{6.33}$$

and since the term $\Delta\delta\lambda_\mathrm{N}$ is equal to zero, the previous equation results in

$$\frac{\partial c_\mathrm{N}}{\partial \mathbf{u}}\Delta\mathbf{u} = \delta g_\mathrm{N}\Delta\lambda_\mathrm{N} + \delta\lambda_\mathrm{N}\Delta g_\mathrm{N} + \lambda_\mathrm{N}\Delta\delta g_\mathrm{N}. \tag{6.34}$$

By substituting Equation 6.24 into 6.28, the variation of g_N can be expressed, in matrix form, as

$$\delta g_\mathrm{N} = \begin{bmatrix}\delta\mathbf{x}_\mathrm{s} & \delta\mathbf{x}^1\end{bmatrix}^\mathrm{T}\mathbf{N}_\mathrm{s}, \tag{6.35}$$

where, from Equations 6.24 and 6.30,

$$\mathbf{N}_\mathrm{s} = \begin{bmatrix} \breve{R}_1\bar{\mathbf{n}} \\ \vdots \\ \breve{R}_{n_\mathrm{s}}\bar{\mathbf{n}} \\ -\bar{R}_1\bar{\mathbf{n}} \\ \vdots \\ -\bar{R}_{n_\mathrm{m}}\bar{\mathbf{n}} \end{bmatrix}. \tag{6.36}$$

In an analogous way, the term Δg_n can be written as

$$\Delta g_\mathrm{N} = \begin{bmatrix}\Delta\mathbf{x}_\mathrm{s} & \Delta\mathbf{x}^1\end{bmatrix}^\mathrm{T}\mathbf{N}_\mathrm{s}. \tag{6.37}$$

The linearisation of the variation of the normal gap can be obtained for the two dimensional case as [Wriggers 02]

$$\begin{aligned}\Delta\delta g_\mathrm{N} = &- \left(\delta\bar{\mathbf{x}}_{,\xi}\Delta\xi + \Delta\bar{\mathbf{x}}_{,\xi}\delta\xi + \bar{\mathbf{x}}_{,\xi\xi}\Delta\xi\delta\xi\right)\cdot\bar{\mathbf{n}} + \\ &\frac{g_\mathrm{N}}{l^2}\left(\delta\bar{\mathbf{x}}_{,\xi} + \bar{\mathbf{x}}_{,\xi\xi}\delta\xi\right)\cdot\bar{\mathbf{n}}\cdot\left(\Delta\bar{\mathbf{x}}_{,\xi} + \bar{\mathbf{x}}_{,\xi\xi}\Delta\xi\right)\cdot\bar{\mathbf{n}},\end{aligned} \tag{6.38}$$

which requires the linearisation of $\bar{\xi}$. This can be obtained by linearising Equation 6.25 and solving for $\Delta\bar{\xi}$ as

$$\Delta\bar{\xi} = \frac{1}{\bar{a}_{11} - g_\mathrm{N}\bar{b}_{11}}\left[(\Delta\mathbf{x}_\mathrm{s} - \Delta\bar{\mathbf{x}})\cdot\bar{\mathbf{x}}_{,\xi} + g_\mathrm{N}\bar{\mathbf{n}}\cdot\Delta\bar{\mathbf{x}}_{,\xi}\right] \tag{6.39}$$

where the metric $\bar{a}_{11}$ and the curvature of the boundary $\bar{b}_{11}$ are given as

$$\bar{a}_{11} = \bar{\mathbf{x}}_{,\xi}\cdot\bar{\mathbf{x}}_{,\xi} = l^2, \tag{6.40}$$

and

$$\bar{b}_{11} = \bar{\mathbf{x}}_{,\xi\xi}\cdot\bar{\mathbf{n}}, \tag{6.41}$$

respectively.

By introducing the vectors

$$\mathbf{N}_{0\mathrm{s}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \bar{R}_{1,\xi}\bar{\mathbf{n}} \\ \vdots \\ \bar{R}_{n_\mathrm{m},\xi}\bar{\mathbf{n}} \end{bmatrix}, \; \mathbf{T}_s = \begin{bmatrix} \breve{R}_1\bar{\mathbf{a}} \\ \vdots \\ \breve{R}_{n_\mathrm{s}}\bar{\mathbf{a}} \\ -\bar{R}_1\bar{\mathbf{a}} \\ \vdots \\ -\bar{R}_{n_\mathrm{m}}\bar{\mathbf{a}} \end{bmatrix}, \; \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{x}_\mathrm{s} \\ \delta\mathbf{x}^1 \end{bmatrix} \text{ and } \Delta\mathbf{x} = \begin{bmatrix} \Delta\mathbf{x}_\mathrm{s} \\ \Delta\mathbf{x}^1 \end{bmatrix}, \tag{6.42}$$

the linearisation of the variation of the gap can be written in matrix form as

$$\Delta\delta g_\mathrm{N} = \delta\mathbf{x}^\mathrm{T}\mathbf{K}_{\Delta\delta}\Delta\mathbf{x}, \tag{6.43}$$

in which

$$\begin{aligned} \mathbf{K}_{\Delta\delta} = & \left(-\frac{l}{m} - \frac{\bar{b}_{11} l g_\mathrm{N}}{m^2} + \frac{\bar{b}_{11} g_\mathrm{N}}{ml} + \frac{\bar{b}_{11}^2 g_\mathrm{N}^2}{m^2 l}\right)\mathbf{N}_{0\mathrm{s}}\mathbf{T}_\mathrm{s}^\mathrm{T} + \\ & \left(-\frac{l}{m} - \frac{\bar{b}_{11} l g_\mathrm{N}}{m^2} + \frac{\bar{b}_{11} g_\mathrm{N}}{ml} + \frac{\bar{b}_{11}^2 g_\mathrm{N}^2}{m^2 l}\right)\mathbf{T}_\mathrm{s}\mathbf{N}_{0\mathrm{s}}^\mathrm{T} + \\ & \left(-\frac{2g_\mathrm{N}}{m} - \frac{\bar{b}_{11} g_\mathrm{N}^2}{m^2} + \frac{g_\mathrm{N}}{l^2} + \frac{2\bar{b}_{11} g_\mathrm{N}^2}{ml^2} + \frac{\bar{b}_{11}^2 g_\mathrm{N}^3}{m^2 l^2}\right)\mathbf{N}_{0\mathrm{s}}\mathbf{N}_{0\mathrm{s}}^\mathrm{T} + \\ & \left(-\frac{\bar{b}_{11} l^2}{m^2} + \frac{\bar{b}_{11}^2 g_\mathrm{N}}{m^2}\right)\mathbf{T}_\mathrm{s}\mathbf{T}_\mathrm{s}^\mathrm{T}, \end{aligned} \tag{6.44}$$

where $m = \bar{a}_{11} - g_\mathrm{N}\bar{b}_{11}$. Finally, it is possible to establish the matrix form of Equation 6.34 as

$$\mathbf{K}_\mathrm{C} = \begin{bmatrix} \delta\mathbf{x} & \delta\lambda_\mathrm{N} \end{bmatrix}^\mathrm{T} \begin{bmatrix} \lambda_\mathrm{N}\mathbf{K}_{\Delta\delta} & \mathbf{N}_\mathrm{s} \\ \mathbf{N}_\mathrm{s}^\mathrm{T} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\lambda_\mathrm{N} \end{bmatrix}. \tag{6.45}$$

The contribution to the right-hand side of the global system of equations stems from Equation 6.18 as

$$\mathbf{f}_\mathrm{C} = \delta\lambda_\mathrm{N} g_\mathrm{N} + \lambda_\mathrm{N}\delta g_\mathrm{N} = \begin{bmatrix} \delta\mathbf{x} & \delta\lambda_\mathrm{N} \end{bmatrix}^\mathrm{T} \begin{bmatrix} \lambda_\mathrm{N}\mathbf{N}_\mathrm{s} \\ g_\mathrm{N} \end{bmatrix}. \tag{6.46}$$

Both the global stiffness matrix and the right-hand side vector will receive additional entries from each collocation point, increasing the size of the system to be solved.

6.3.4 Contact Stress

When solving the global system of equations in conjunction with the Lagrange Multiplier method, the additional entries that arise from Equations 6.45 and 6.46 correspond to the normal contact pressures λ_N that act upon each collocation point. These contact pressures can then be used to compute the contact stresses that act on the contacting slave curve. In standard linear Lagrangian-based formulations, since the slave points are the element nodes, in order to obtain the contact stresses it is only required to compute the area corresponding to half of the length of the adjacent elements. In the context of Isogeometric Analysis employing the Point-to-Segment algorithm a different approach must be considered. In the current work, a method which consists in dividing a point's equivalent normal contact force by the physical length (associated with the same point) is employed.

The first step is to distribute the contribution of each Lagrange Multiplier $\lambda_{\mathrm{N}j}$ from the collocation points $\breve{\xi}_j$ as

$$P_i = \sum_{j=1}^{n_{\mathrm{c}}} R_i\left(\breve{\xi}_j\right)\lambda_{\mathrm{N}j}. \tag{6.47}$$

In the slave segment, a given control point $\mathbf{B}_i\left(x,y\right)$ only affects the curve in the range $\left[\xi_i, \xi_{i+p_{\mathrm{s}}+1}\right]$ since the basis functions $R_i\left(\xi\right) = 0$ for $\xi \not\subset \left[\xi_i, \xi_{i+p_{\mathrm{s}}+1}\right]$. Consequently, the parametric length l_i^{s} of the segment associated with P_i is given as

$$l_i^{\mathrm{s}} = \frac{\Xi^{\mathrm{s}}\left(\xi_{i+p_{\mathrm{s}}+1}\right) - \Xi^{\mathrm{s}}\left(\xi_i\right)}{\sum_{j=1}^{n_{\mathrm{s}}} l_j^{\mathrm{s}}}, \tag{6.48}$$

where $\boldsymbol{\Xi}^{\mathrm{s}}$ is the knot vector defining the slave segment. It should be noted that the parameters defined in the previous equation are normalised. This normalisation arises from the fact that, since there exists inter-element continuity, the sum of the parametric slave curve lengths would be superior to 1.0.

The physical length of the slave segment can be obtained by numerical integration as

$$l^{\mathrm{ph}} = \sum_{i=1}^{n_{\mathrm{s}}+p_{\mathrm{s}}} \int_{S_{\mathrm{C}}} \sqrt{d_{\mathrm{x}}^2 + d_{\mathrm{y}}^2}\mathrm{d}\xi, \tag{6.49}$$

where

$$d_{\mathrm{x}} = \sum_{j=1}^{n_{\mathrm{G}}} R_{i,\xi} x_i, \tag{6.50}$$

and

$$d_{\mathrm{y}} = \sum_{j=1}^{n_{\mathrm{G}}} R_{i,\xi} y_i, \tag{6.51}$$

in which x_i and y_i define the physical coordinates of the slave curve control point and $n_{\mathrm{G}} = p_{\mathrm{s}} + 1$ is the number of integration points in each knot span. By making use of Equations 6.48 and 6.49, the physical length of the slave curve segment associated with point $\mathbf{P}_i$ can then be written as

$$l_i^{\mathrm{s,ph}} = l_i^{\mathrm{s}} \times l^{\mathrm{ph}}. \tag{6.52}$$

Finally, the normal contact stress can be computed as

$$\sigma_i^{\mathrm{C}} = \frac{P_i}{l_i^{\mathrm{s,ph}}}. \tag{6.53}$$

6.4 Implementation of the Contact Algorithm

The Point-to-Segment contact algorithm was implemented in the in-house developed Isogeometric Analysis code ICO. The flowchart of the code's structure can be seen in Figure 6.3. A detailed description of the steps performed in boxes containing contributions to the contact problem is given in the following.

6.4.1 Initialise

In this step, all the data is read from the input file and all the variables are allocated. The coordinates of the Greville points in the parametric curve are computed from the curve's knot vector as

$$\breve{\xi}_i = \frac{\xi_{i+1} + \cdots + \xi_{i+p+1}}{p}. \tag{6.54}$$

Additionally, the contact status of each collocation point is set to `Not Active` and the global system of equations is augmented in order to accommodate the additional degrees of freedom coming from the Lagrange Multiplier method.

6.4.2 Compute Contact Contributions

The *compute contact contributions* box contains the core of the contact module using the PTS algorithm in conjunction with the Lagrange Multiplier method. The main steps are represented inside the dashed box on the left-hand side of Figure 6.3.

For each collocation point, the physical coordinates of the slave point $\mathbf{x}_\mathrm{S}$ are computed based on its parametric coordinates $\breve{\xi}_i$ as

$$\mathbf{x}_\mathrm{S} = \sum_{i=1}^{n_\mathrm{s}} R_i\left(\breve{\xi}_i\right)\mathbf{x}_i^2.$$

The code will then compute the closest point projection of the slave point onto the master segment $\bar{\mathbf{x}}$ by iteratively solving

$$(\mathbf{x}_\mathrm{S} - \bar{\mathbf{x}}) \cdot \bar{\mathbf{a}} = 0,$$

using the Newton algorithm presented in Box 6.1.

Once the coordinates of the CPP are determined, it is possible to compute the gap as

$$g_\mathrm{N} = (\mathbf{x}_\mathrm{S} - \bar{\mathbf{x}}) \cdot \bar{\mathbf{n}},$$

where, for the two-dimensional case, the normal vector is obtained as

$$\bar{\mathbf{n}} = \bar{\mathbf{a}} \times \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^\mathrm{T}. \tag{6.55}$$

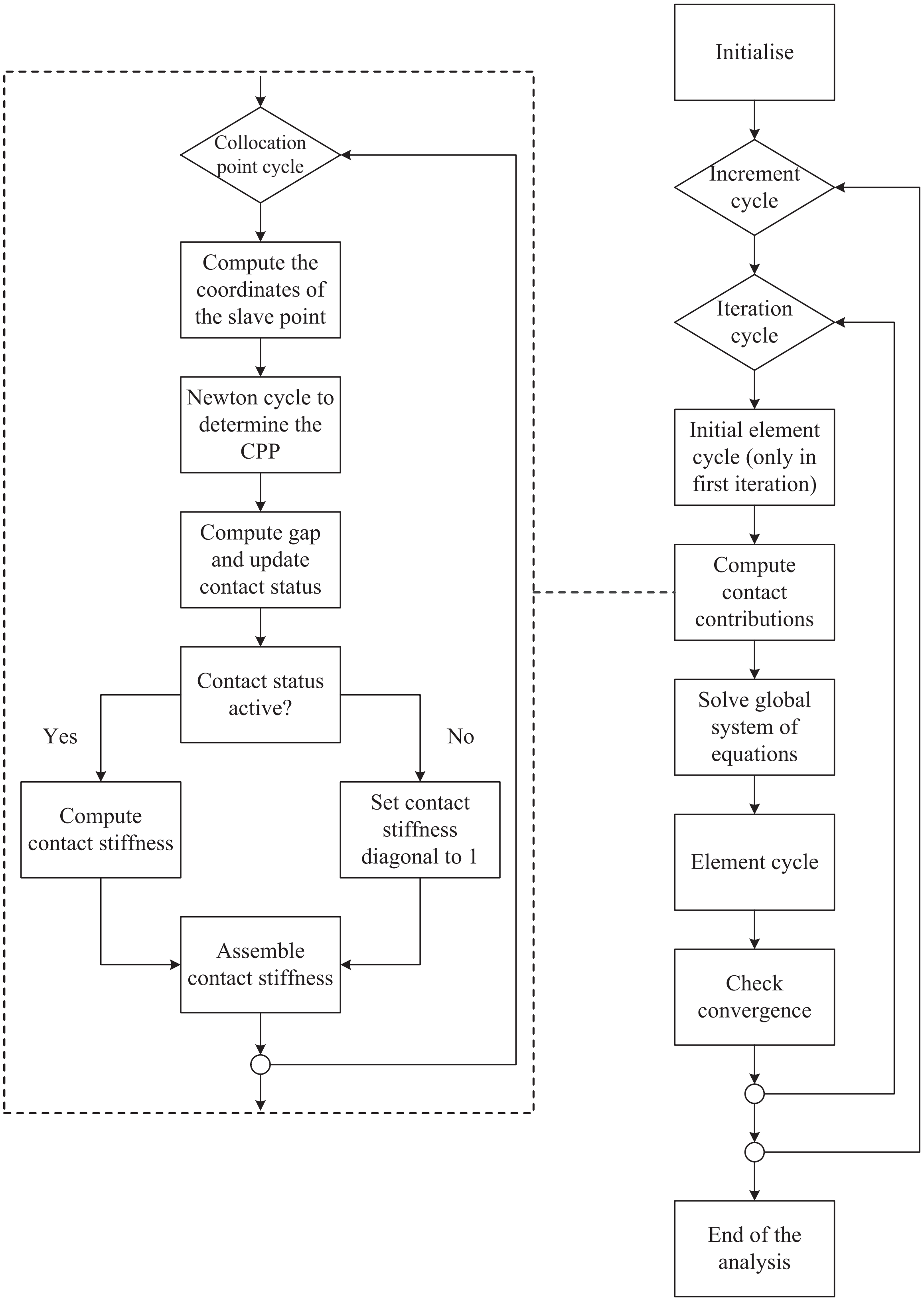

Initialise
Increment cycle
Iteration cycle
Initial element cycle (only in first iteration)
Compute contact contributions
Solve global system of equations
Element cycle
Check convergence
End of the analysis
Collocation point cycle
Compute the coordinates of the slave point
Newton cycle to determine the CPP
Compute gap and update contact status
Contact status active?
Yes
No
Compute contact stiffness
Set contact stiffness diagonal to 1
Assemble contact stiffness

Figure 6.3: Flowchart of the ICO code including contact.

If $g_{\mathrm{N}} \leq 0$, then penetration will occur and the contact status of the collocation point is set to `Active`. The contact stiffness is computed using Equations 6.44 and 6.45 and assembled into the global system of equations. Otherwise, if $g_{\mathrm{N}} > 0$, the contact status will be set as `Not Active` and the diagonal of the contact stiffness matrix will be set to 1.0 and all other entries are set to 0.0.

Box 6.1: Newton algorithm to determine the CPP of the slave point onto the master segment.

```
1. DO Newton iteration ($k$)
    (a) compute the physical coordinates of the master point and its derivatives
            $\bar{\mathbf{x}} = \sum_{i=1}^{n_{\mathrm{m}}} \bar{R}_i \left(\bar{\xi}^k\right) \mathbf{x}_i^1$
            $\bar{\mathbf{x}}_{,\xi} = \sum_{i=1}^{n_{\mathrm{m}}} \bar{R}_{i,\xi} \left(\bar{\xi}^k\right) \mathbf{x}_i^1$
            $\bar{\mathbf{x}}_{,\xi\xi} = \sum_{i=1}^{n_{\mathrm{m}}} \bar{R}_{i,\xi\xi} \left(\bar{\xi}^k\right) \mathbf{x}_i^1$
    (b) compute master segment length and its derivative
            $l = |\bar{\mathbf{x}}_{,\xi}| = \left| \sum_{i=1}^{n_{\mathrm{m}}} \bar{R}_{i,\xi} \left(\bar{\xi}^k\right) \mathbf{x}_i^1 \right|$
            $l_{,\xi} = |\bar{\mathbf{x}}_{,\xi\xi}| = \left| \sum_{i=1}^{n_{\mathrm{m}}} \bar{R}_{i,\xi\xi} \left(\bar{\xi}^k\right) \mathbf{x}_i^1 \right|$
    (c) compute the tangent to the master segment and its derivative
            $\bar{\mathbf{a}} = \frac{\bar{\mathbf{x}}_{,\xi}}{l}$
            $\bar{\mathbf{a}}_{,\xi} = \frac{\bar{\mathbf{x}}_{,\xi}}{l_{,\xi}}$
    (d) compute new CPP parametric coordinate
            $\bar{\xi}^{k+1} = \bar{\xi}^k - \frac{(\mathbf{x}_{\mathrm{s}} - \bar{\mathbf{x}}) \cdot \bar{\mathbf{a}}}{\left(\mathbf{x}_{\mathrm{s},\xi} - \bar{\mathbf{x}}_{,\xi}\right) \cdot \bar{\mathbf{a}}_{,\xi}}$
    (e) IF $(\mathbf{x}_{\mathrm{s}} - \bar{\mathbf{x}}) \cdot \bar{\mathbf{a}} \leq 1.0 \times 10^{-8}$ then exit cycle, otherwise set $k = k + 1$ and perform another iteration
2. END DO
```

6.4.3 Solve Global System of Equations

After assembling all the contact stiffness contribution of each collocation point, the global system of equations

$$\begin{bmatrix} \mathbf{K}(\mathbf{u}) + \mathbf{K}_\mathrm{C}(\mathbf{u},\boldsymbol{\lambda}) & \mathbf{C}_\mathrm{C}(\mathbf{u}) \\ [\mathbf{C}_\mathrm{C}(\mathbf{u})]^\mathrm{T} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u} \\ \Delta\boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^\mathrm{ext} - \mathbf{f}^\mathrm{int} \\ 0 \end{Bmatrix} - \begin{Bmatrix} \mathbf{C}_\mathrm{C}(\mathbf{u})\boldsymbol{\lambda} \\ \mathbf{G}_\mathrm{C}(\mathbf{u}) \end{Bmatrix} \tag{6.56}$$

is solved for the incremental displacements $\Delta\mathbf{u}$ and incremental Lagrange multipliers $\Delta\boldsymbol{\lambda}$. In Equation 6.56, matrix $\mathbf{K}(\mathbf{u})$ is the standard tangential stiffness matrix described in previous chapters, and $\mathbf{f}^\mathrm{ext}$ and $\mathbf{f}^\mathrm{int}$ are the external and internal forces, respectively. Matrices $\mathbf{K}_\mathrm{C}(\mathbf{u},\boldsymbol{\lambda})$ and $\mathbf{C}_\mathrm{C}(\mathbf{u})$ arise from the contributions of each collocation point to the global system of equations and $\mathbf{G}_\mathrm{C}(\mathbf{u})$ defines the normal contact constraint.

Chapter 7

Numerical Examples

The performance of the NURBS-based formulations proposed in Chapter 5 are assessed using a set of well-known benchmark problems in both linear and nonlinear regimes. Additionally, in the context of contact mechanics, the validation of the implemented Point-to-Segment algorithm described in Chapter 6 in the linear elastic regime is performed, also by means of various benchmark problems.

In this chapter, the methodologies described in Chapter 5 and 6 are validated using various benchmark problems. In particular, the first two sections are related to the assessment of the performance of the H2ANS element (see Section 5.5) in the linear and nonlinear regimes for thin plate and shell structures, followed by a couple of numerical problems where the volumetric locking pathology is dominant. The remainder of the chapter is dedicated to the analysis of contact problems in the linear elastic range in the context of Isogeometric Analysis.

In the numerical examples presented in the following, except the ones involving contact mechanics, a single NURBS patch was considered for modelling each problem (unless otherwise stated). In every example, the initial geometry was defined using the lowest order and number of control points possible. Successive refined meshes were obtained by the process of k-refinement (see Section 3.1.4) using the in-house developed code written in Matlab and described in Section 3.4.1. Standard Gaussian quadrature is employed in all the presented examples.

7.1 Linear Elastic Problems

In the present section, the performance of the proposed H2ANS formulation is assessed in the analysis of shell-like structures in the linear elastic range. In particular, the proposed numerical experiments consist of the study of a straight and a curved cantilever beam, as well as of the solution of the well-known *shell obstacle course*, proposed by Belytschko *et al.* [Belytschko 85] as a set of benchmarks for the assessment of shell analysis procedures. In all numerical examples, the proposed formulation is compared with quadratic and cubic NURBS-based solid and Kirchhoff-Love shell elements. Whenever possible, other NURBS-based shell and solid-shell results available in the literature are also considered, for comparison purposes. In this section the following nomenclature for the different employed formulations is adopted:

.: H*n*: Standard solid NURBS-based element of degree *n*;

.: KL*n*: Kirchhoff-Love shell element of degree *n*, as proposed by Kiendl *et al.* [Kiendl 09];

.: 3p-HS: Quadratic 3-parameter Kirchhoff-Love shell element with a Hybrid Stress modification of the membrane part, as proposed by Echter *et al.* [Echter 13];

.: 3p-DSG: Quadratic 3-parameter Kirchhoff-Love shell element with a Discrete Strain Gap modification of the membrane part, as proposed by Echter *et al.* [Echter 13];

.: 5p-stand(-DSG): Quadratic 5-parameter Reissner-Mindlin shell element (with a Discrete Strain Gap modification of the membrane part), as proposed by Echter *et al.* [Echter 13];

.: 5p-hier(-HS): Quadratic 5-parameter Reissner-Mindlin shell element with hierarchic difference vector (and a Hybrid Stress modification of the membrane part), as proposed by Echter *et al.* [Echter 13];

.: Mixed 2: Quadratic solid-shell element employing a mixed method, as proposed by Bouclier *et al.* [Bouclier 13a];

.: Local Bbar 2: Quadratic solid-shell element employing a mixed method with modified $\bar{\mathbf{B}}$-projection, as proposed by Bouclier *et al.* [Bouclier 13a].

In addition, whenever possible, a comparison with high performance Lagrangian-based solid and solid-shell formulations is carried out. The nomenclature employed is defined as follows:

.: Sch09: Solid-shell formulation with in-plane reduced integration and stabilization, as proposed by Schwarze and Reese [Schwarze 09];

.: Ree07: Solid-shell using reduced integration with hourglass stabilization and EAS [Reese 07];

- Kim05: Solid-shell based on the ANS method with plane stress assumptions [Kim 05];
- RESS: Reduced Enhanced Solid-Shell element with stabilization of hourglass modes and one EAS mode [Alves de Sousa 05];
- HCiS18(12): Solid(solid-shell) element with 18(12) EAS enhancing parameters, as proposed by [Alves de Sousa 03];
- Are03: EAS solid element with penalty stabilization [Areias 03];
- Leg03: EAS solid-shell element with stabilisation of hourglass modes [Legay 03];
- H1/ME9: Mixed-enhanced fully integrated eight-node element with 9 enhanced modes [Kasper 00].

7.1.1 Straight Cantilever Beam

In this first example, a straight beam clamped at one end is subjected to a vertical load F at the opposite free end, as can be seen in Figure 7.1. From the Bernoulli beam theory, the strain energy U of the structure is given as

$$U = \frac{2F^2L^3}{Ewt^3}, \tag{7.1}$$

where E is the elastic modulus and L, w, and t are the beam's length, width, and thickness, respectively. By expressing the results in terms of the strain energy, it is possible to assess the accuracy of the stress and strain fields predicted by the formulations. For a deeper insight of the performance of the proposed formulation, the current problem is subdivided into two cases.

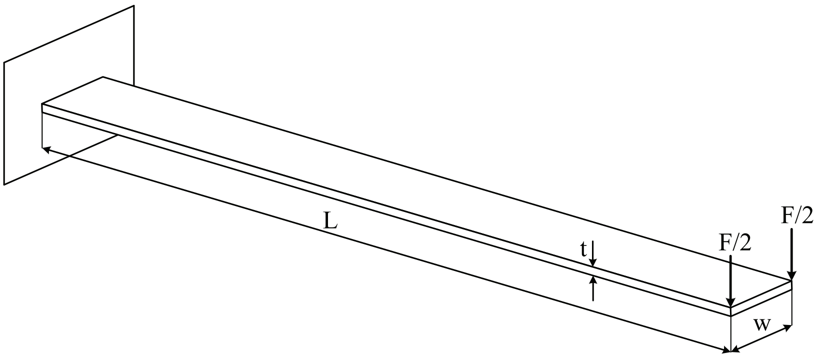


Figure 7.1: Scheme of the straight beam problem.

In the first case, the convergence of distinct formulations is analysed for a beam of $L = 100.0$ and $w = t = 1.0$. The material properties are taken as $E = 1000.0$ and $\nu = 0.0$. The problem is discretized with only one element along the width and thickness directions. The results for the normalized strain energy versus the number of elements along the length direction are presented in Figure 7.2, for distinct NURBS-based elements. It can be seen that the proposed H2ANS formulation is able to reproduce the reference solution, even when considering a very coarse mesh. The results are superior to those attained by quadratic solid and Kirchhoff-Love shell elements. The results for cubic formulations are not reported due to the fact that a cubic polynomial interpolation is, in this case, enough to reproduce the exact solution.

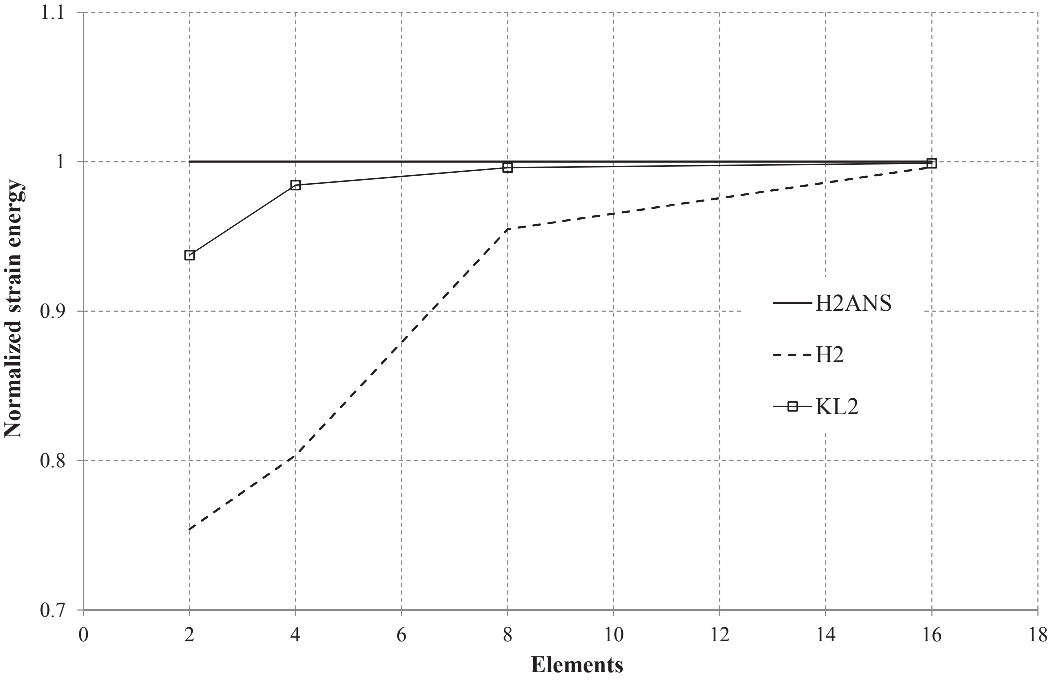


Figure 7.2: Normalized strain energy versus mesh density for the straight cantilever beam problem with a constant slenderness of $L/t = 100.0$.

In the second case, a mesh composed of eight elements is considered, and the problem is studied for different beam thickness values. As the beam becomes thinner, transverse shear locking effects will be increasingly dominant, making this example a valuable tool for evaluating the capability of a given formulation to alleviate this kind of locking phenomenon. The results for the normalized strain energy versus slenderness are presented in Figure 7.3, for the same formulations as before. The proposed NURBS-based solid-shell element is able to obtain good results for both thick and thin beams, demonstrating a very low sensitivity to shear locking effects. As expected, as the thickness of the beam decreases, the results for the standard quadratic NURBS-based solid element tend to deteriorate. It can also be seen that the KL2 formulation can be considered as free from shear locking (as being based in the Kirchhoff-Love rationale). It should be highlighted that, when higher slenderness ratios are considered, the stiffness matrices resulting from the solid elements become ill-conditioned, leading to difficulties when solving the global system of equations. This situation is not detected when shell elements are instead used.

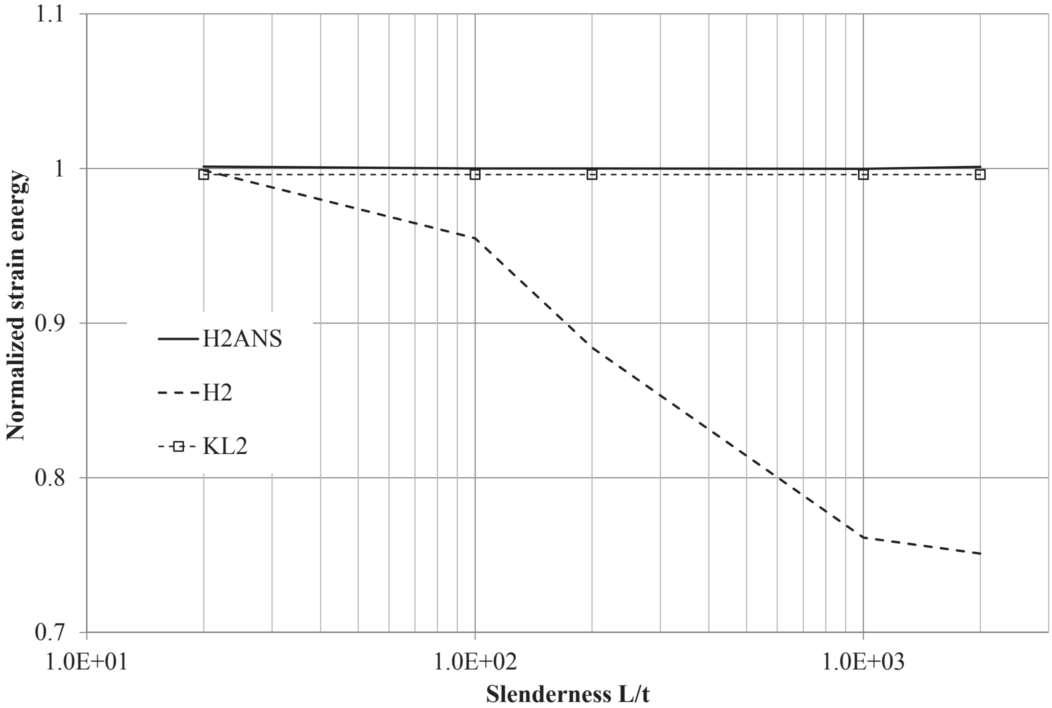


Figure 7.3: Normalized strain energy versus beam slenderness for the straight cantilever beam problem for a eight NURBS element mesh.

7.1.2 Curved Cantilever Beam

In this example a curved beam, consisting of a quarter of a circle, is clamped at one end and subjected to a transversal load at its the free end. Due to the curvature of the beam, membrane locking will be the dominant parasitic phenomena [Echter 13]. In addition, when solid (or solid-shell) elements are used to model the curved profile, curvature thickness (trapezoidal) locking may also be present. The structure is represented in Figure 7.4 for a single element mesh, along with the corresponding control lattice. The final mesh is obtained by performing an order elevation along the thickness (radial) and width directions, followed by knot insertion in circumferential direction. The structure has a radius, at the neutral surface, of $R = 10.0$ and a width $w = 1.0$. An elastic modulus of 1000.0 and a Poisson's ratio of 0.0 are considered. The load is given as a function of the thickness t, as $F = 0.1t^3$. From the Bernoulli beam theory, the radial displacement can be computed to be equal to 0.942 [Echter 13]. The problem is discretized using ten NURBS elements, with only one element through the thickness and width directions.

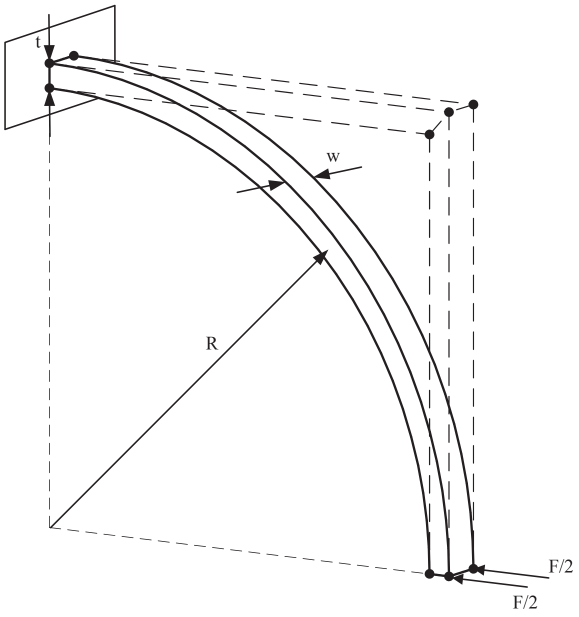


Figure 7.4: Scheme of the curved cantilever beam problem discretised with a single element and corresponding control lattice.

The results for the radial displacement as the beam slenderness R/t is increased are presented in Figure 7.5. It can be seen that, although the proposed formulation is not locking free, it is able to significantly improve the behaviour of the standard quadratic NURBS solid element. The performance of H2ANS is also superior to the quadratic Kirchhoff-Love shell element. In fact, H2 and KL2 formulations are seen to suffer from locking, even when considering a moderately thin shell. Cubic elements present a better overall performance, although not being completely locking-free.

In Figure 7.6, the proposed formulation is now also compared with the shell formulations presented in [Echter 13]. The results obtained by H2ANS are very close to those attained by the 5p-stand-DSG shell element. Echter and co-workers [Echter 13] justify the deterioration of the results obtained by the 5p-stand-DSG element through shear locking effects. However, as seen in the previous example, since the ANS methodology is able to alleviate shear locking effects, the decrease of the H2ANS performance as the slenderness of the beam increases may be related to curvature thickness locking. As observed in [Echter 13], in this case the 3p-DSG and 3p-HS formulations are instead completely locking-free.

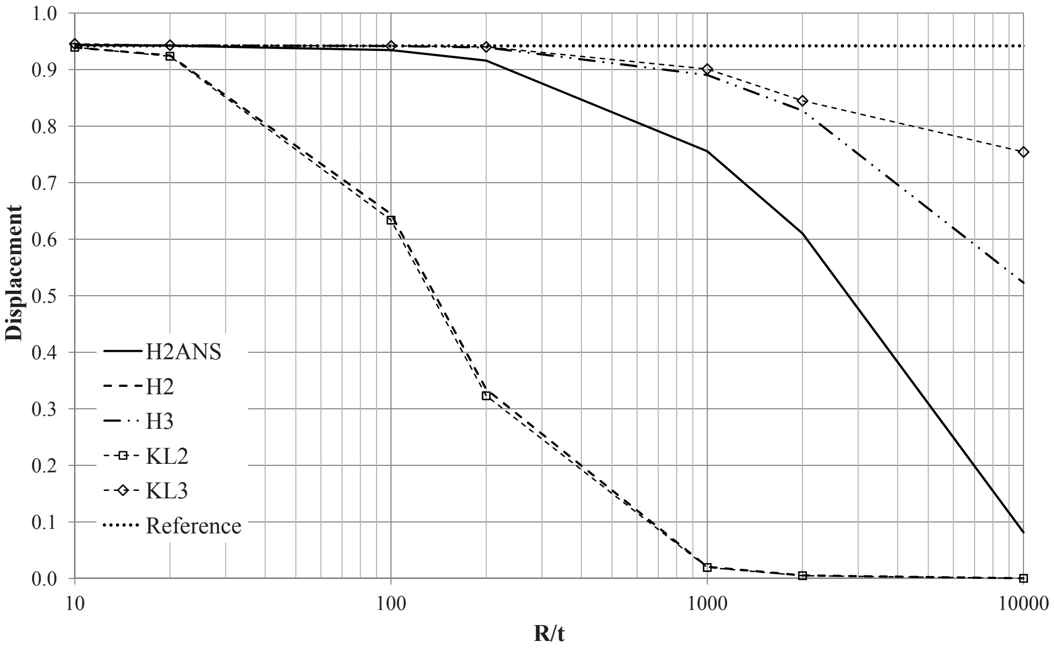


Figure 7.5: Displacement versus slenderness for the curved cantilever beam problem (1).

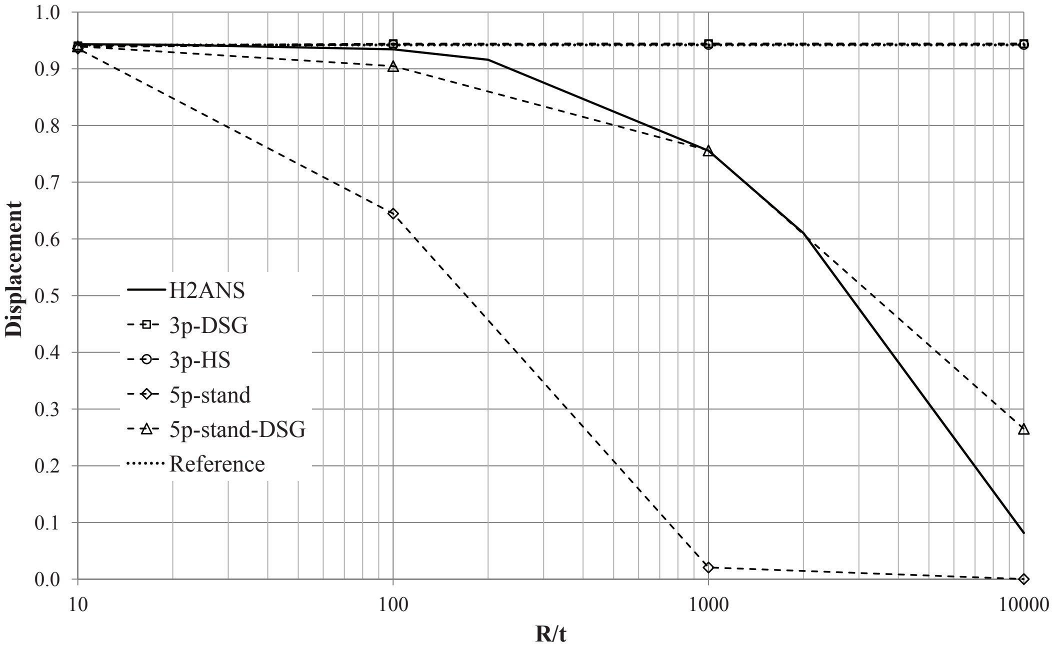


Figure 7.6: Displacement versus slenderness for the curved cantilever beam problem (2).

7.1.3 Shell Obstacle Course I: The Scordelis-Lo Roof

In this example, introduced by Scordelis and Lo [Scordelis 69], a cylindrical shell supported by rigid diaphragms in the curved edges is subjected to a volume force (self-weight). The geometry of the problem is presented in Figure 7.7 and the dimensions of the structure are: radius $R = 25.0$, length $L = 50.0$ and thickness $t = 0.25$. The magnitude of the volume force is given as $\rho g = 360$, where ρ is the density and g is the gravity acceleration constant, for a set of coherent unities. The elastic properties are given by $E = 4.32 \times 10^8$ and $\nu = 0.0$. Due to symmetry conditions, only a quarter of the structure is modelled.

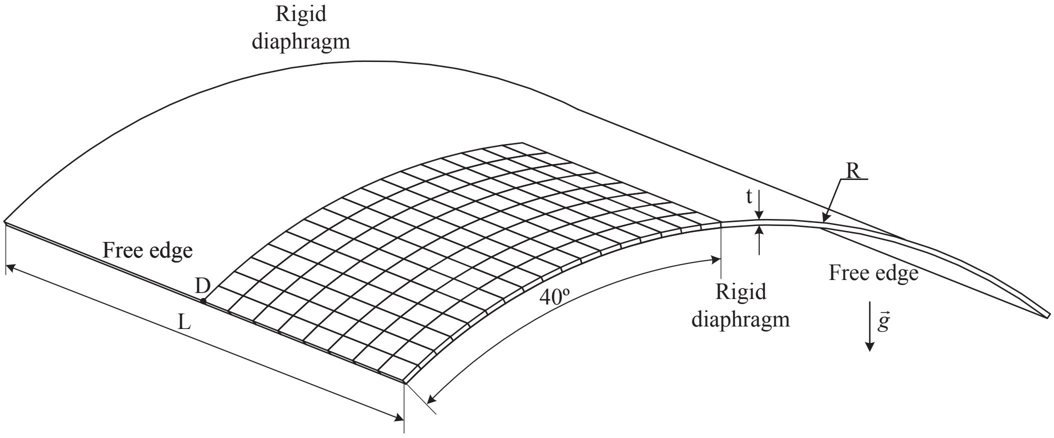


Figure 7.7: Schematic representation of the Scordelis-Lo roof problem.

The vertical displacement of the midpoint of the free edge (point D in Figure 7.7) is numerically computed and compared with the reference solution of 0.3024, with the results being presented in Figure 7.8. The proposed H2ANS formulation is able to obtain good results and a very fast convergence, significantly improving the behaviour of the conventional formulation (H2 element). In fact, it can be seen that the results from H2ANS are similar to those obtained by cubic solid and Kirchhoff-Love shell elements.

The results for the normalised displacements of point D for various Lagrangian-based formulations available in the literature are compared with the proposed methodology in Figure 7.9. As can be seen, the H2ANS element presents competitive results when compared to different solid and solid-shell formulations.

In the following, the NURBS-based solid elements H2 and H3 are compared with the proposed H2ANS solid-shell in terms of computational costs. The CPU time obtained by each formulation as a function of the number of control points is presented in Figure 7.10. The results are normalised by the CPU time obtained by the H3 formulation using a mesh composed of 4900 control points. It can be observed that the proposed NURBS-based solid-shell formulation presents a significantly lower computational cost when compared with the cubic solid element, while being able to obtain a similar prediction of the displacement of

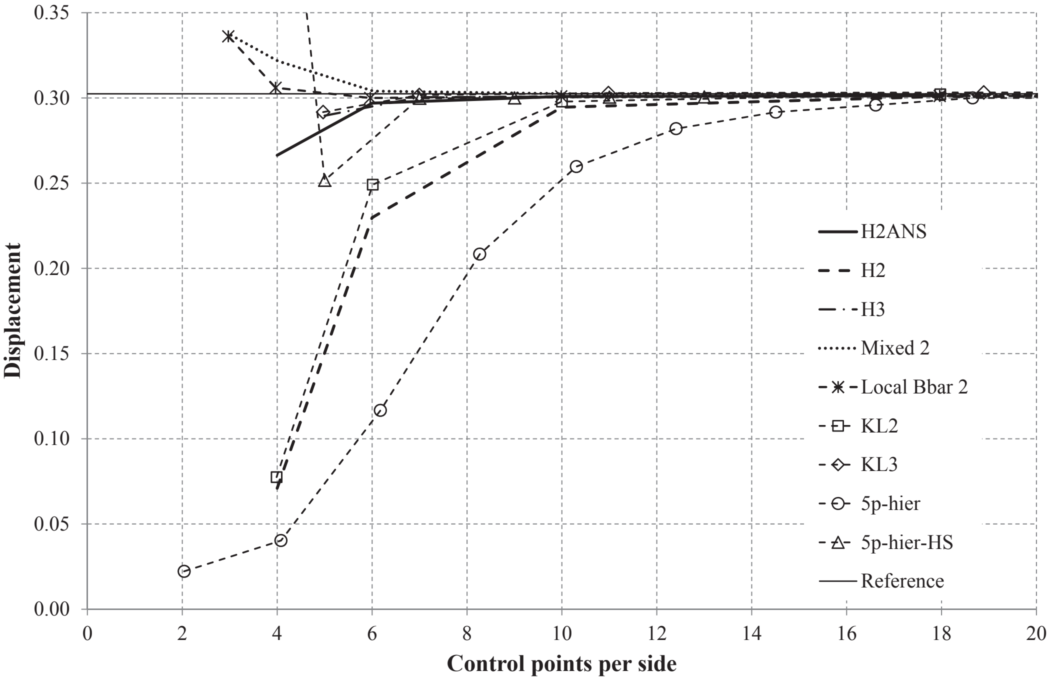


Figure 7.8: Displacement of the midpoint of the free edge for the Scordelis-Lo roof.

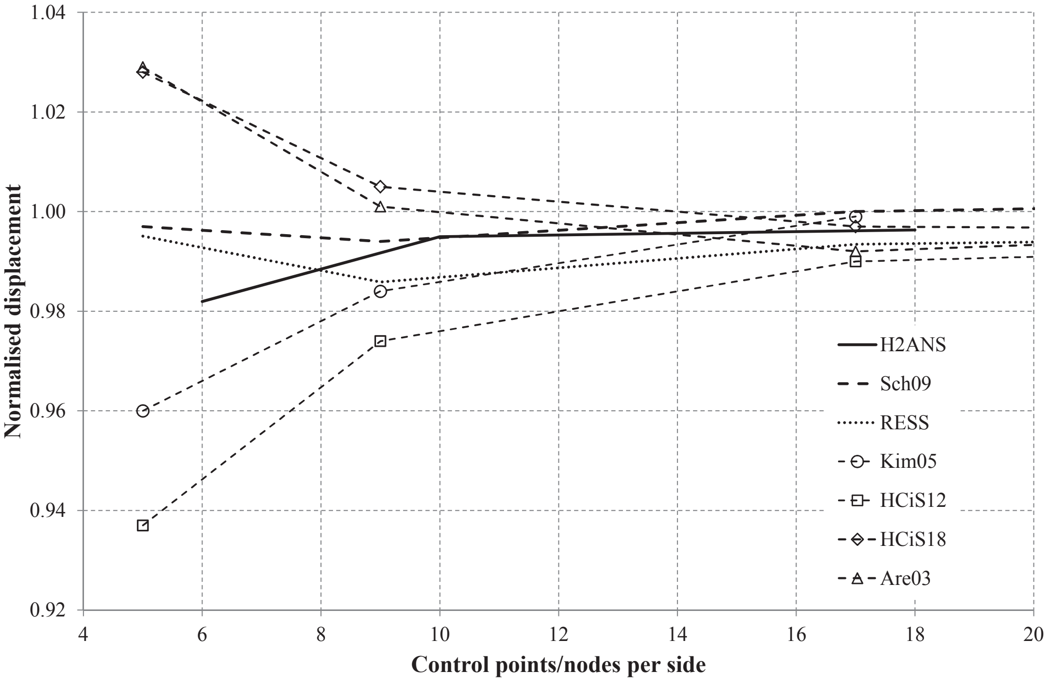


Figure 7.9: Displacement of the midpoint of the free edge for the Scordelis-Lo roof: comparison with Lagrangian-based Finite Element formulations.

point D, as seen in Figure 7.8.

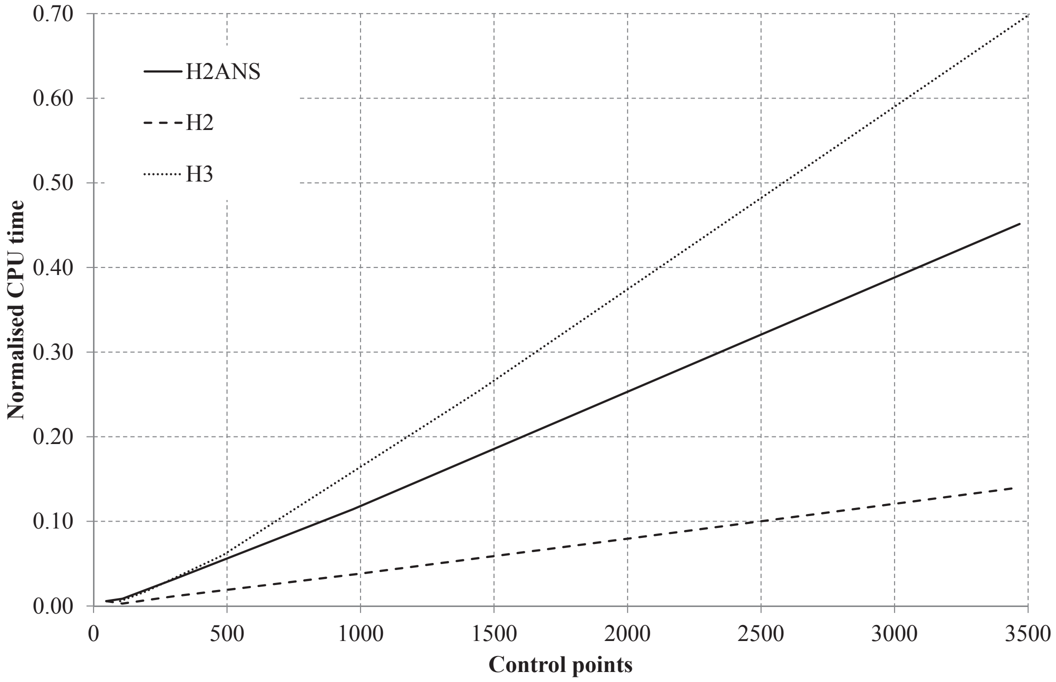


Figure 7.10: Scordelis-Lo roof: comparison of computational costs.

7.1.4 Shell Obstacle Course II: Full Hemispherical Shell

The full hemispherical shell schematically represented in Figure 7.11 is another well-known benchmark to assess the performance of shell (and solid-shell) elements. In this problem, a hemisphere of radius $R = 10.0$ and thickness $t = 0.04$ is subjected to a pair of opposite concentrated loads applied at antipodal points of the equator, while the equator edge is considered to be free. Due to symmetry conditions, only one quarter of the structure needs to be modelled, as seen in the figure. The magnitude of the load is $F = 1.0$, the material parameters are given as $E = 6.825 \times 10^7$ and $\nu = 0.3$, and the reference radial displacement at point A is $u = 0.0924$.

In Figure 7.12 the results for the radial displacement at point A versus the number of control points per side is presented. Once again, the proposed H2ANS formulation is able to obtain good results and convergence, being superior to quadratic solid and Kirchhoff-Love shell elements, and comparable to formulations accounting for higher order interpolations.

The normalised results obtained by the H2ANS element can also be compared to Lagrangian-based formulations, as shown in Figure 7.13. The results demonstrate that the proposed element is able to obtain a performance that is superior or similar to some of the solid and solid-shell Lagrangian formulations available in the literature. These results once

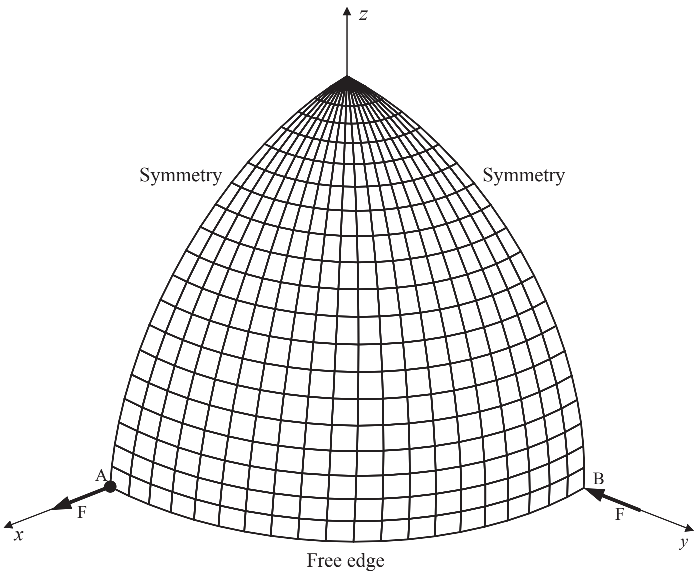


Figure 7.11: Full hemispherical shell problem setup (1/4 of the whole structure is shown).

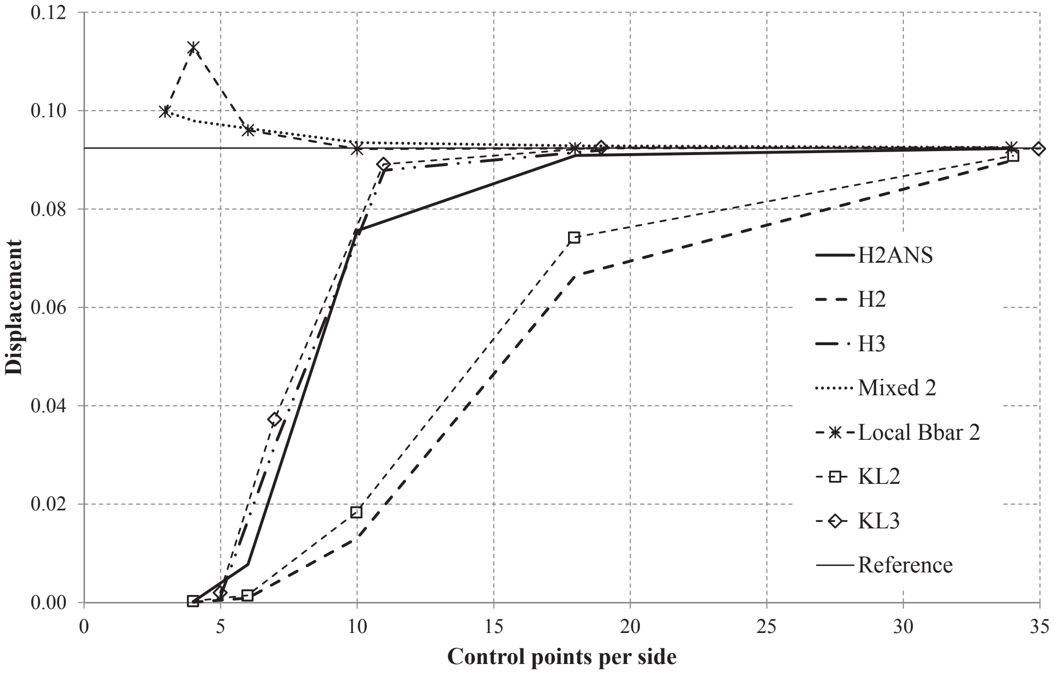


Figure 7.12: Radial displacement of point A for the full hemispherical shell problem.

again allow to infer that the proposed methodology can lead to a competitive formulation in terms of reliability of the provided numerical solution.

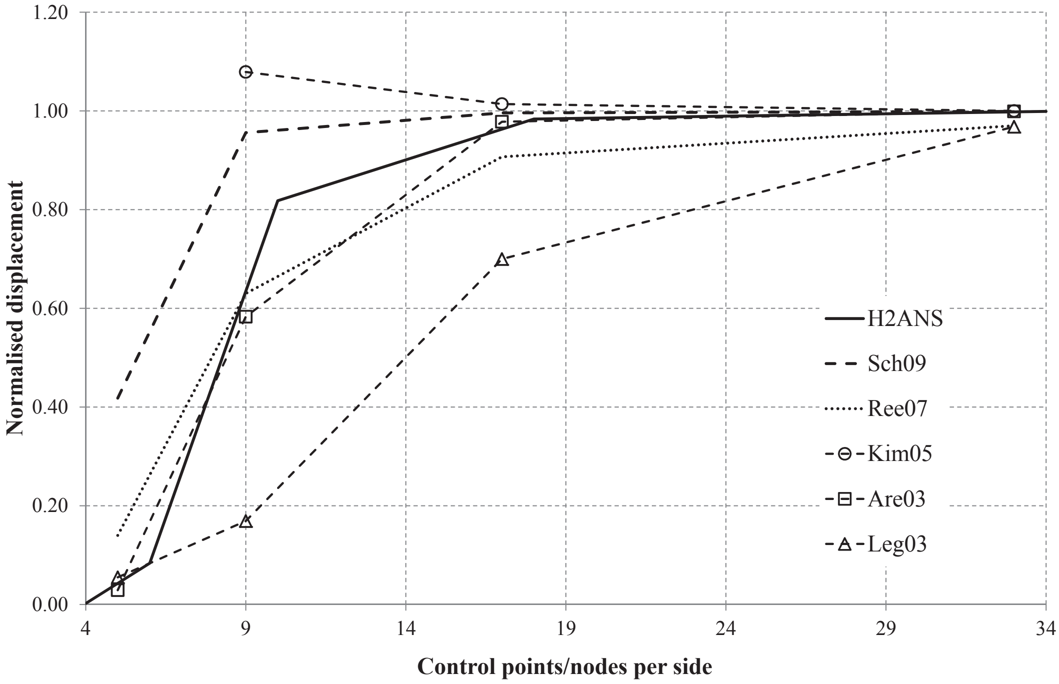


Figure 7.13: Radial displacement of point A for the full hemispherical shell problem: comparison with Lagrangian-based Finite Element formulations.

7.1.5 Shell Obstacle Course III: Pinched Cylinder

As a last example of this set of three shell obstacle course problems, the pinched cylinder with end diaphragms subjected to a pair of concentrated loads is presented. This is a rather demanding example due to the very localized strain state resulting from the application of the point load. The cylinder has radius $R = 300.0$, length $L = 600.0$ and thickness $t = 3.0$, as can be schematically seen in Figure 7.14. The concentrated loads have a magnitude of 1.0, for material properties given as $E = 3.0 \times 10^6$ and $\nu = 0.3$. Due to symmetry, only one eighth of the structure is modelled. The reference solution for the radial displacement at the loaded point is given as $u = 1.8248 \times 10^{-5}$.

The results for the different formulations are presented in Figure 7.15. The H2ANS NURBS-based element gives again better results than those coming from quadratic elements, even if, in this case, not as good as those obtained with cubic elements.

Finally, the radial displacement obtained for the H2ANS is normalised and compared with solid and solid-shell formulations available in the literature. The results are depicted in Figure 7.16. As can be seen, the results obtained by the proposed formulation are very similar to those from the HCiS12, HCiS18, Are03 and H1/ME9 elements.

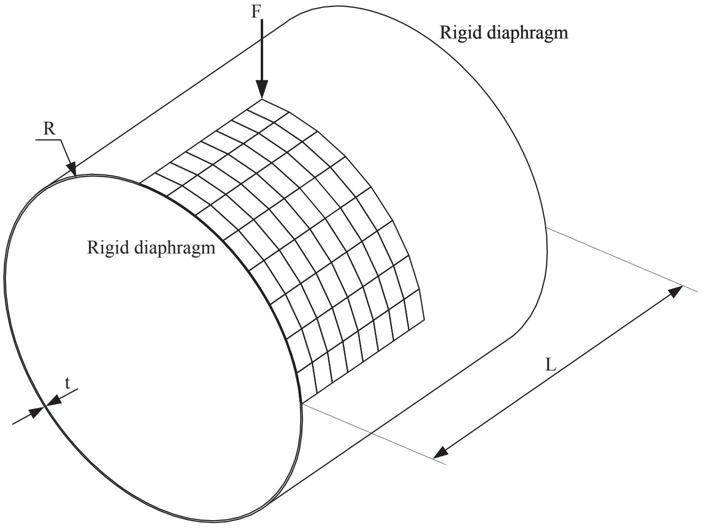


Figure 7.14: Schematic representation of the pinched cylinder problem.

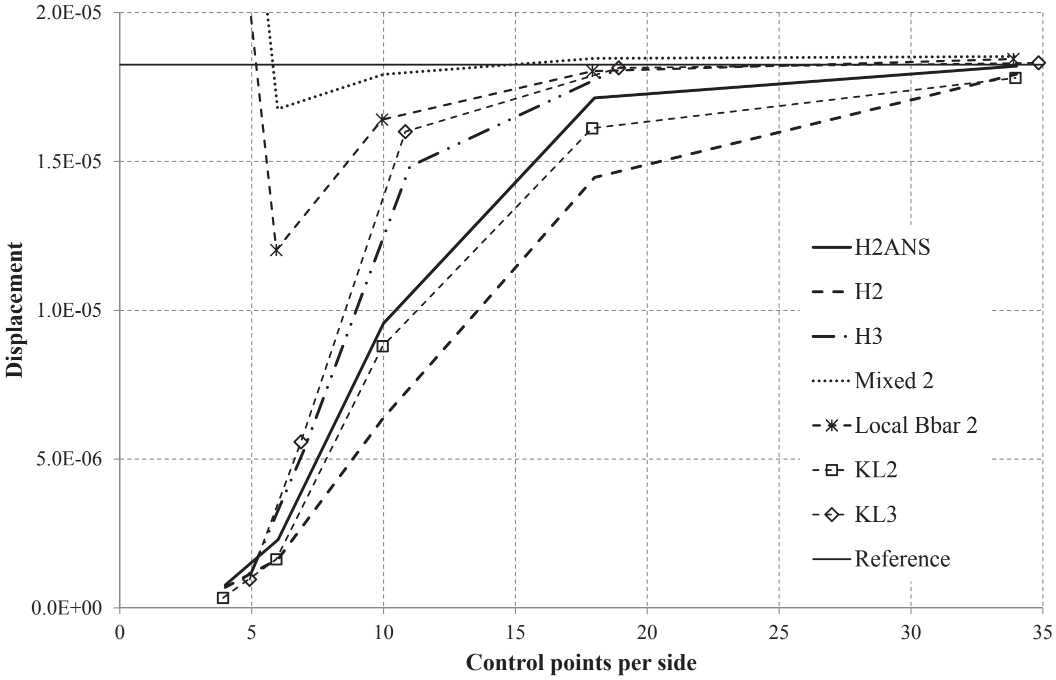


Figure 7.15: Radial displacement for the pinched cylinder problem.

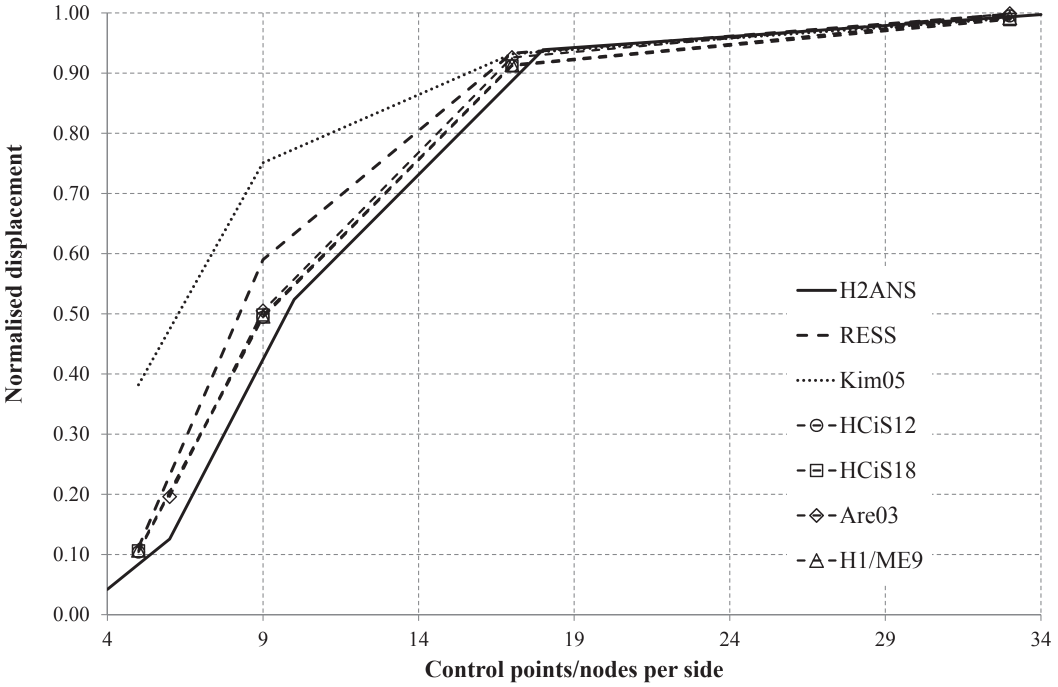


Figure 7.16: Radial displacement for the pinched cylinder problem: comparison with Lagrangian--based Finite Element formulations.

7.2 Nonlinear Problems

In the following, the proposed H2ANS NURBS-based element is applied for the analysis of shell-like structures in the geometric and material nonlinear regimes. The implementation of the formulation for this type of problems follows the methodologies presented in Sections 4.2 and 4.3.

The reference to results available in the literature is performed by using the first three letters of the first authors' name and the year of publishing.

7.2.1 Elastic Large Deflection Bending of a Beam

In this example, a beam is clamped in one end and subjected to an in-plane transverse force $F = 1000.0$ in its free end, as shown in Figure 7.17. Results in the context of FEM can be found in a number of references, such as [Simo 90a, Betsch 96, Miehe 98, Valente 04b]. The geometry of the beam is characterized by a length $L = 1.0$, width $w = 0.1$ and thickness $t = 0.1$, while the elastic properties are defined by the bulk modulus $\kappa = 83.33 \times 10^5$ and shear modulus $G = 38.46 \times 10^5$. When performing the numerical simulation, the load is applied in ten equally spaced increments.

The analysis of the problem is initiated by performing a mesh convergence study in order to compare the performance of the H2ANS formulations and its solid counterpart H2, in

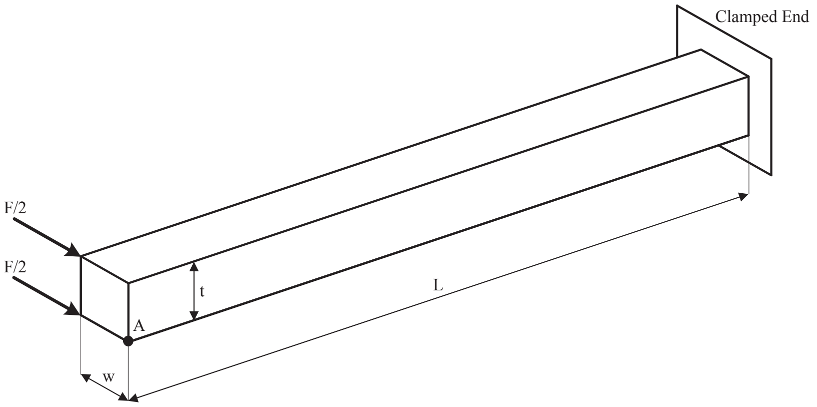


Figure 7.17: Schematic representation of the elastic large deflection membrane bending of a beam benchmark.

the context of a geometry nonlinear analysis. In this problem, the beam is discretised using a single element along the width and thickness directions. The load-displacement curves of point A for various mesh densities are presented in Figure 7.18 and compared with a reference solution coming from the work of Simo *et al.* [Simo 90a], for a mesh composed of 10 elements. The results show that the H2ANS element is able to provide a very good solution considering a mesh comprised of just 6 elements in the length direction. It can also be seen that the proposed solid-shell NURBS-base formulation has a better performance than the standard solid quadratic element, specially when considering coarse meshes.

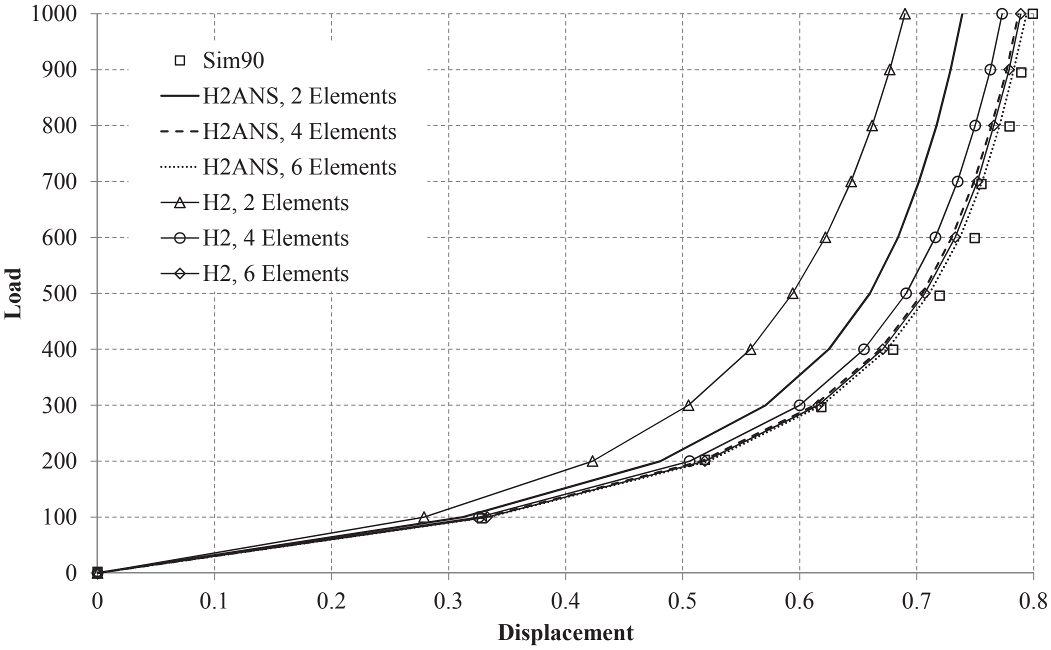


Figure 7.18: Elastic large deflection bending of a beam: displacement of point A versus load.

Employing the same benchmark problem, the sensitivity to mesh distortion is assessed.

To that end, two types of distortion are considered and represented in Figure 7.19 for a 6 element mesh. For each case, the distortion parameter d varies within the range $[0.0, 3.0]$. The results for the displacements of point A for the distorted meshes of type I (DTI) and type II (DTII) are presented in Figures 7.20 and 7.21, respectively. It can be seen that, in the DTI case, when considering distortion parameters up to $d = 0.2$, the solution is not affected in a significant manner. It is also demonstrated that the DTII represents a more difficult situation, leading to a higher mesh sensitivity for both the H2ANS and H2 elements, although this sensitivity is more significant for a distortion parameter $d > 0.2$. In conclusion, for the range of distortion levels considered, the proposed solid-shell presents a similar or slightly superior performance when compared to the standard solid element. For some higher distortion levels, especially in case DTII, the H2ANS tends however to present convergence difficulties, in this example.

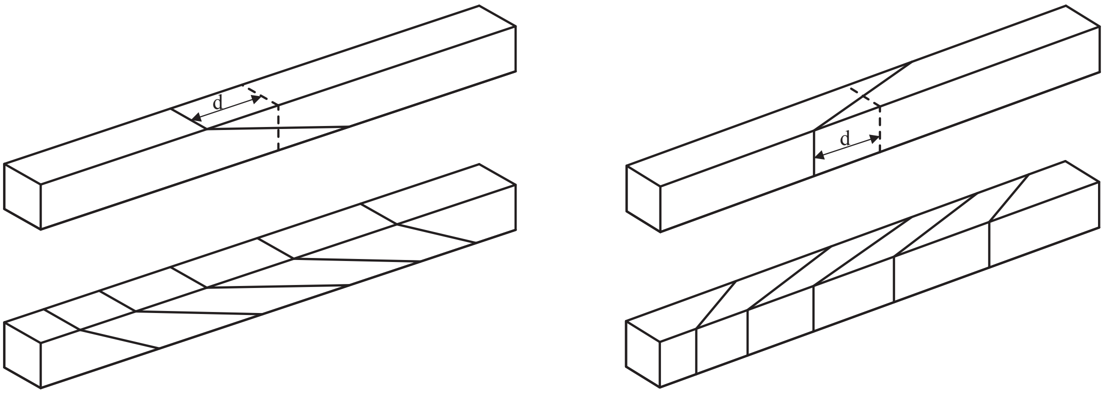


Figure 7.19: Elastic large deflection bending of a beam: definition of distortion parameter d and NURBS mesh for (left) distortion type I and (right) distortion type II.

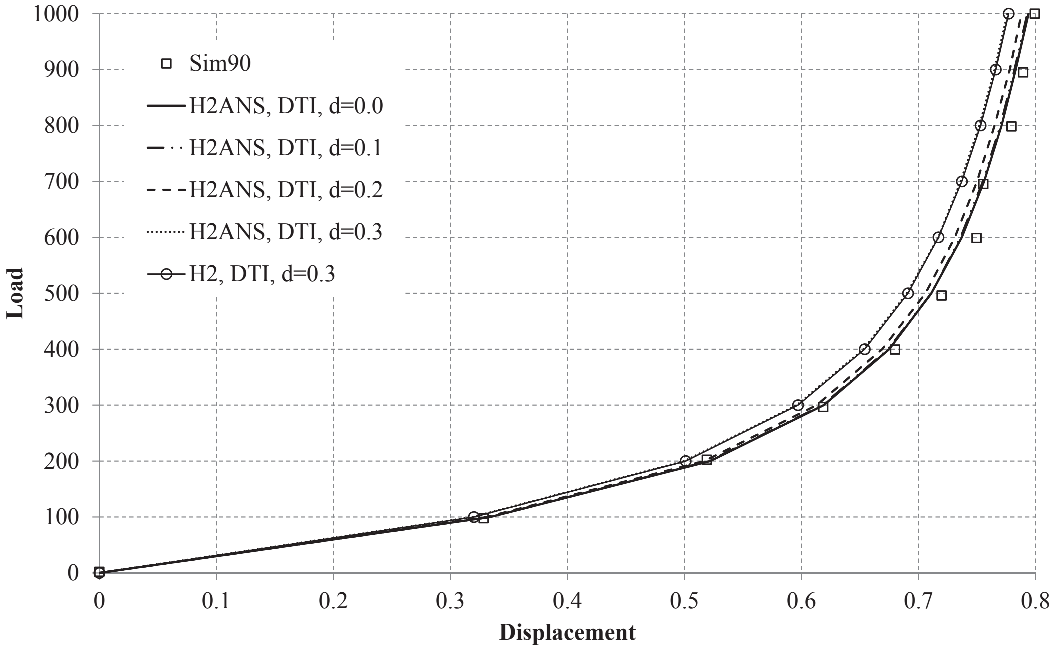


Figure 7.20: Elastic large deflection bending of a beam: displacement of point A versus load for distorted mesh of type I.

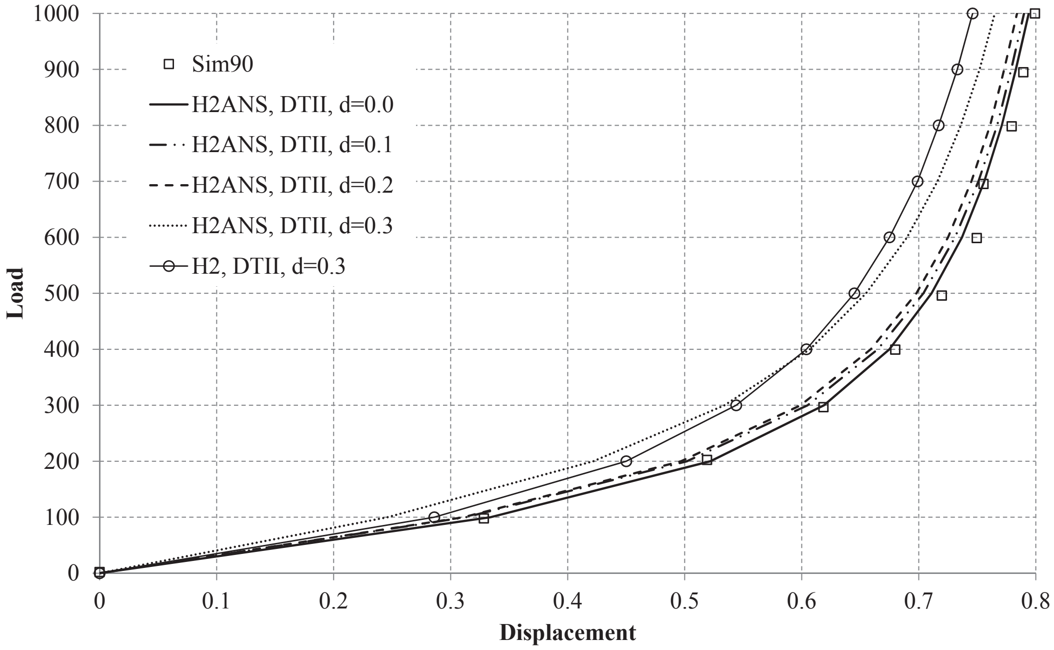


Figure 7.21: Elastic large deflection bending of a beam: displacement of point A versus load for distorted mesh of type II.

7.2.2 Large Deflection of an Elastic and Elasto-Plastic Straight Cantilever Beam

In this example the flat cantilever beam previously seen in the linear elastic range (Section 7.1.1), is now considered in the nonlinear regime. This example assesses the out-of-plane bending behaviour of the formulations and has been analysed by a wide range of authors considering both elastic [Simo 90c, Parisch 95, Miehe 98, El-Abbasi 00, Valente 04b, Reese 07, Schwarze 11] and elasto-plastic [Dvorkin 95, Eberlein 99, Valente 04a] constitutive relations.

In the first case, a geometric nonlinear regime combined with linear elastic material is considered. The elastic modulus is defined as $E = 1 \times 10^7$ and the Poisson's coefficient as $\nu = 0.3$. Following the above mentioned authors, the load is considered to be constant and with a total magnitude given by $F = 40 \times \lambda$, where λ is a load factor ranging from 0.0 to 1.0. The load is applied in 10 equal steps and the solutions for the vertical tip displacements are compared with theoretical values coming from the literature [Frisch-Fay 62]. In Figure 7.22, the results obtained by the proposed H2ANS solid-shell element and the standard quadratic NURBS-based solid element H2 are compared by means of a convergence study. A single element is considered along the width and thickness directions. As can be seen, the H2ANS formulation is able to significantly improve the behaviour of the H2 element, specially when a coarse mesh is employed. For a mesh consisting of 16 elements, the H2ANS element is able to attain the reference solution.

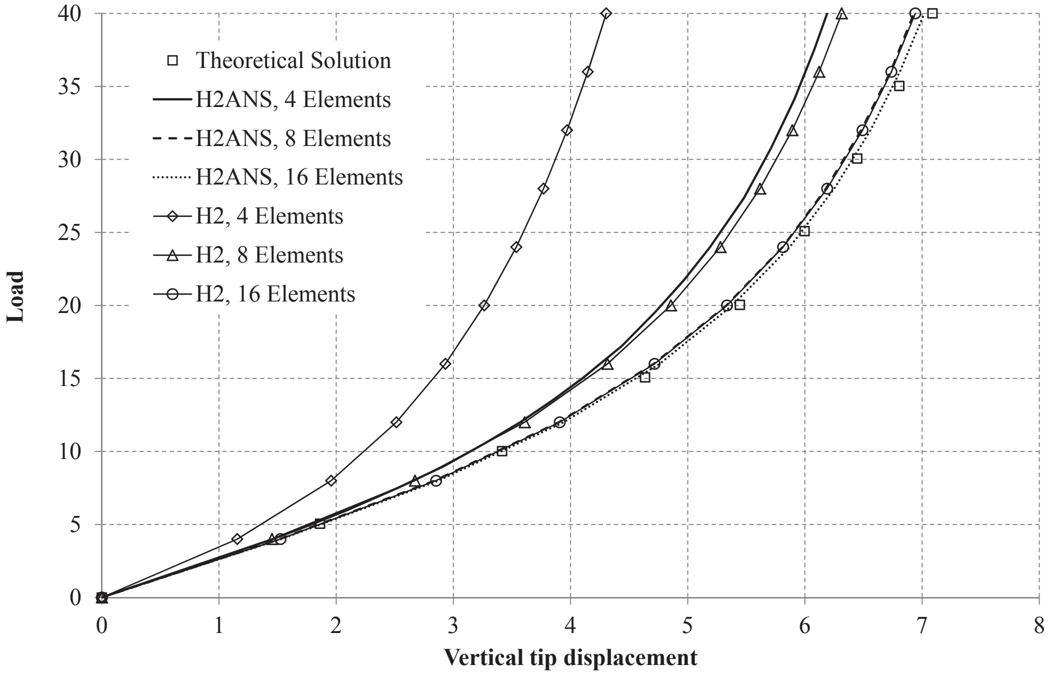


Figure 7.22: Large deflection of an elastic straight cantilever beam: mesh convergence study.

In Figure 7.23, the results obtained are now compared with Lagrangian-based formulations from the works of Valente *et al.* [Valente 04b], Reese [Reese 07] and Schwarze and Reese [Schwarze 11]. The results demonstrate that the proposed H2ANS formulation for a given mesh density can be competitive when compared to classic solid and solid-shell formulations available. However, it is important to take into account that the results coming from the literature are, in some cases, referred to linear elements based on reduced integration schemes, making them particularly efficient in terms of computational costs to the expense of, in some cases, introducing numerical instabilities into the solution. The final shape of the beam can be seen in Figure 7.24, along with the control lattice of the deformed configuration.

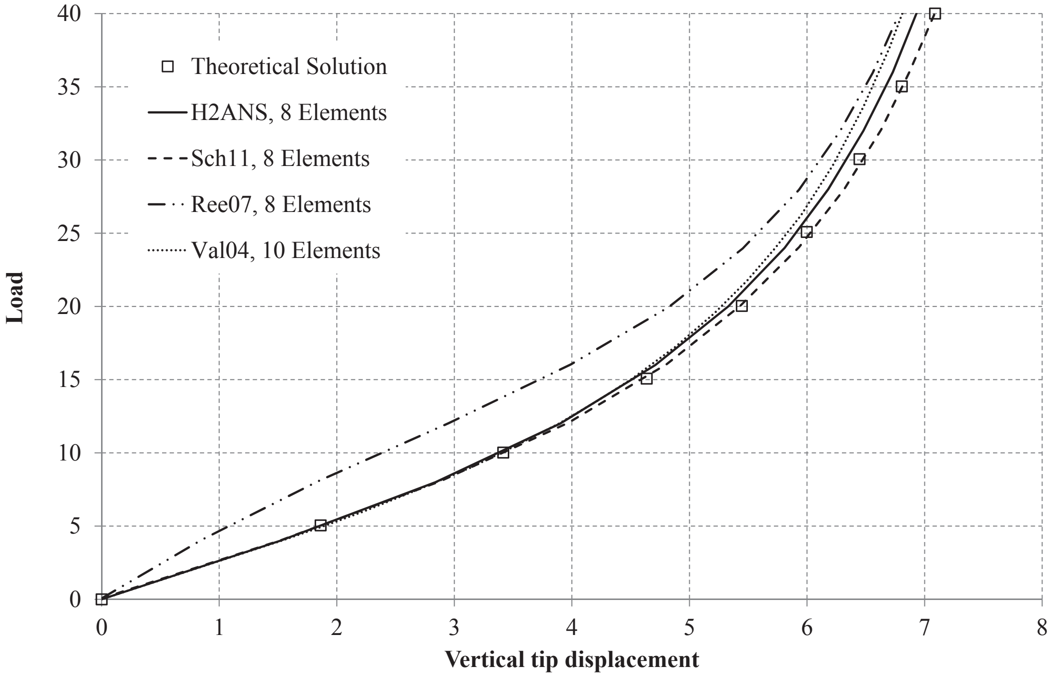


Figure 7.23: Large deflection of an elastic straight cantilever beam: comparison with available finite element formulations.

In the following, the present example is used to validate the multipatch implementation in the commercial software package Abaqus, as described in Section 3.4.4. To that end, the beam is divided into two patches midway through the length direction. The comparison with the single patch model is presented in Figure 7.25 for the H2ANS NURBS-based element. It is possible to observe that for the coarser mesh the models present some variation in the load-displacement curves, although this variation is no longer significant after successive mesh refinements. The H2 formulation presents a similar behaviour.

The same problem in also analysed considering both geometric and material nonlinear conditions. The geometry remains unaltered but the elasto-plastic constitutive relations are now defined by the elastic modulus $E = 1.2 \times 10^7$ and Poisson's coefficient $\nu = 0.3$, while

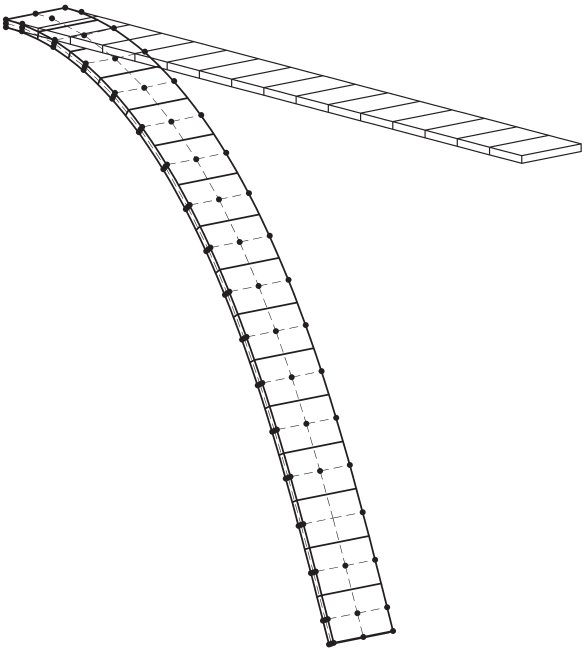


Figure 7.24: Large deflection of an elastic straight cantilever beam: initial geometry and final deformed shape with control lattice.

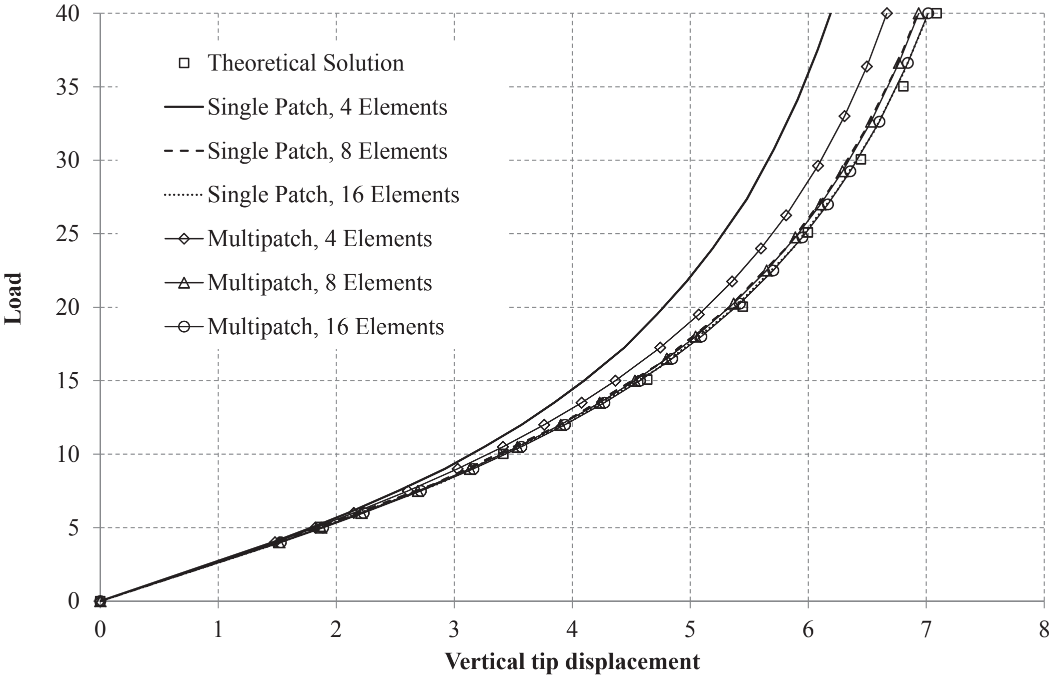


Figure 7.25: Large deflection of an elastic straight cantilever beam: comparison between the single patch and multipatch models.

the yield stress is described by means of the equivalent plastic strain $\varepsilon^{\mathrm{pl}}$ as

$$\sigma_y\left(\varepsilon^{\mathrm{pl}}\right) = \sigma_0 + H\varepsilon^{\mathrm{pl}}, \tag{7.2}$$

in which the initial yield stress is $\sigma_0 = 2.4 \times 10^4$ and the linear isotropic hardening coefficient is equal to $H = 1.2 \times 10^5$. The results for the tip displacement are given in Figure 7.26 for two mesh densities. In the same plot, results from the works of Dvorkin *et al.* [Dvorkin 95] and Eberlein and Wriggers [Eberlein 99] are also presented for comparison purposes, both employing references employ a mesh consisting of 20 elements with a single element in the width and thickness directions. The results demonstrate that the H2ANS NURBS-based solid-shell element formulation is able to more accurately represent the behaviour of the beam, when compared to the standard quadratic solid NURBS-based element, which presents a stiffer behaviour. Also, the results of the proposed solid-shell are in good accordance with the ones coming from the literature, specially for the 20 element mesh.

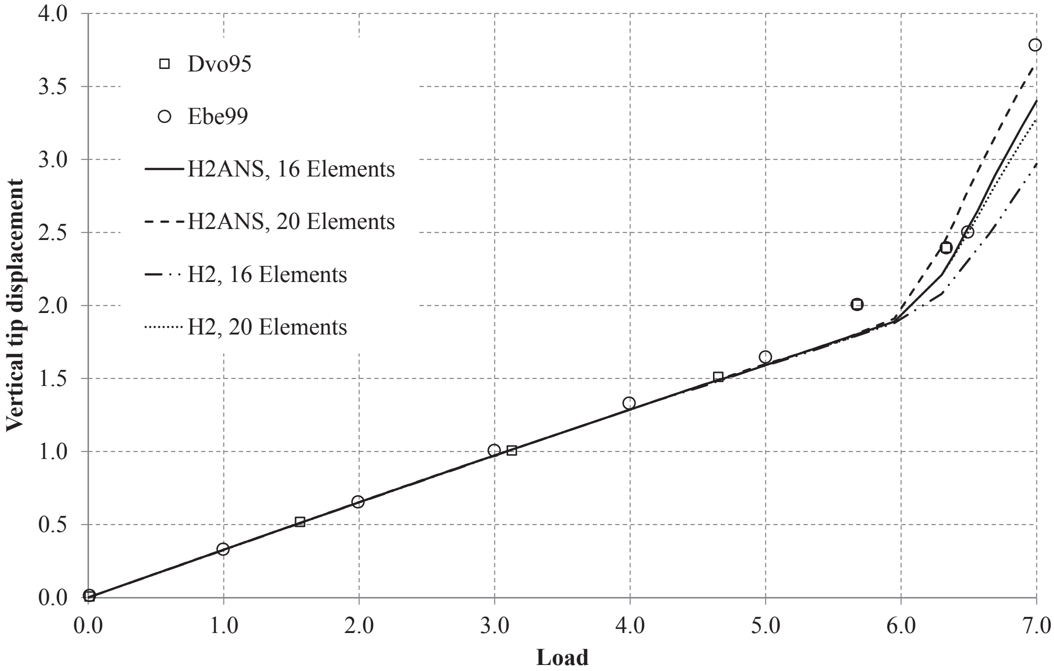


Figure 7.26: Large deflection of an elasto-plastic straight cantilever beam: mesh convergence study.

7.2.3 Geometric Nonlinear Pinching of a Clamped Cylinder

In this example, a cylindrical shell is fully clamped in one end and subjected to a pair of point loads with opposite directions in its free end [Parisch 91, Brank 95, Valente 03, Valente 04b, Alves de Sousa 06b]. The schematic representation of the problem can be seen in Figure 7.27. The elastic constitutive parameters are given by $E = 2.0685 \times 10^7$

and $\nu = 0.3$, while the geometry is defined by the length $L = 3.048$, radius $R = 1.016$ and thickness $t = 0.03$. The maximum imposed inwards load has a magnitude of $F = 1600.0 \times \lambda$, with $\lambda \in [0.0, 1.0]$. Due to symmetry conditions, only a quarter of the structure is modelled.

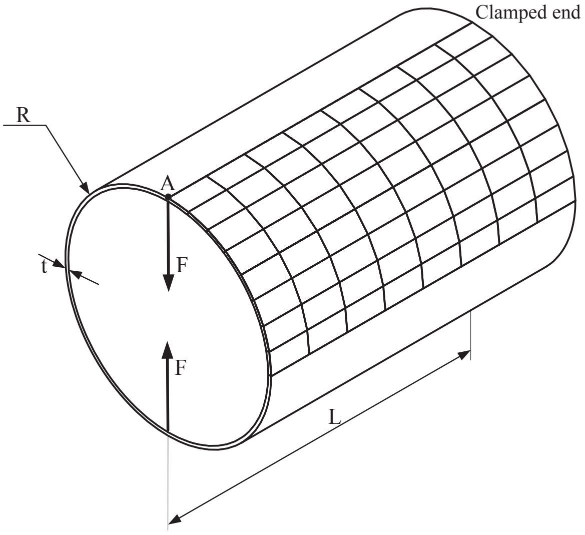


Figure 7.27: Schematic view of the clamped cylinder benchmark.

The load-displacement curves with respect to point A are presented in Figure 7.28 for the H2 and H2ANS NURBS-based elements, considering different mesh densities. The results are compared with a reference solution coming from the work of Brank *et al.* [Brank 95]. As can be seen, the H2ANS solid-shell element presents an improved performance when compared with its solid counterpart, being more noticeable in coarse meshes. The results for the 16 × 16 mesh are now compared with those coming from the literature for solid-shell elements, as for instance proposed by Alves de Sousa *et al.* [Alves de Sousa 06b] and Valente *et al.* [Valente 04b] for the same mesh density (Figure 7.29). Once again, the results attained by the H2ANS formulation are in good agreement with the ones presented by the references. In Figure 7.30 the deformed configurations at different load stages are depicted, for the upper half of the cylinder. It is worth noting that the loaded points go beyond the highest physical displacement possible (the radius of the shell).

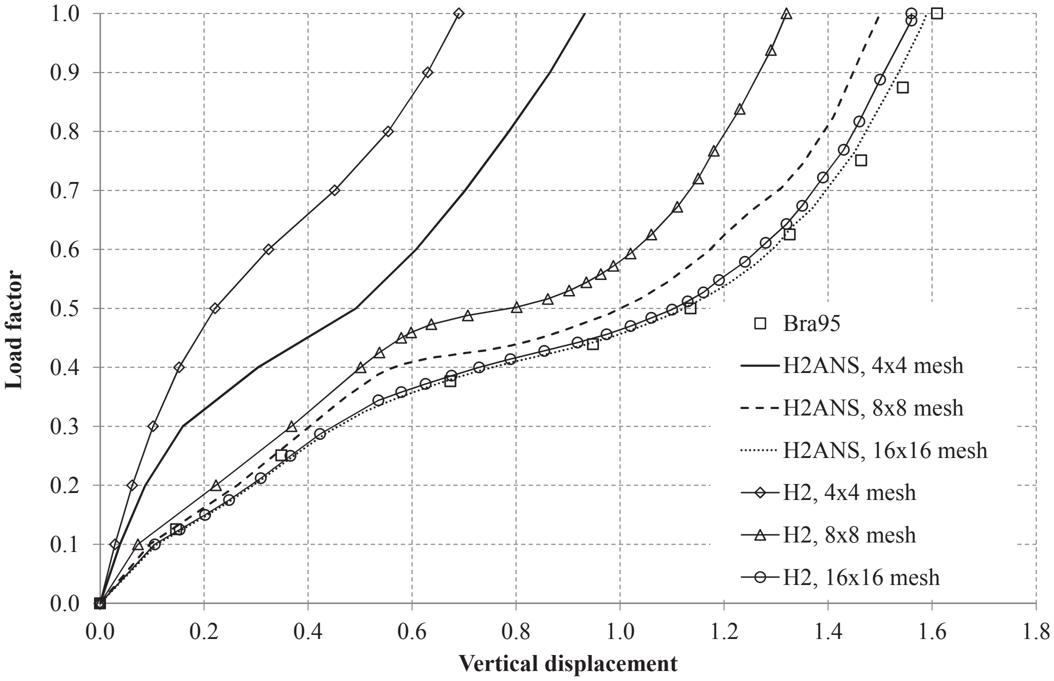


Figure 7.28: Pinching of a clamped cylinder: convergence study.

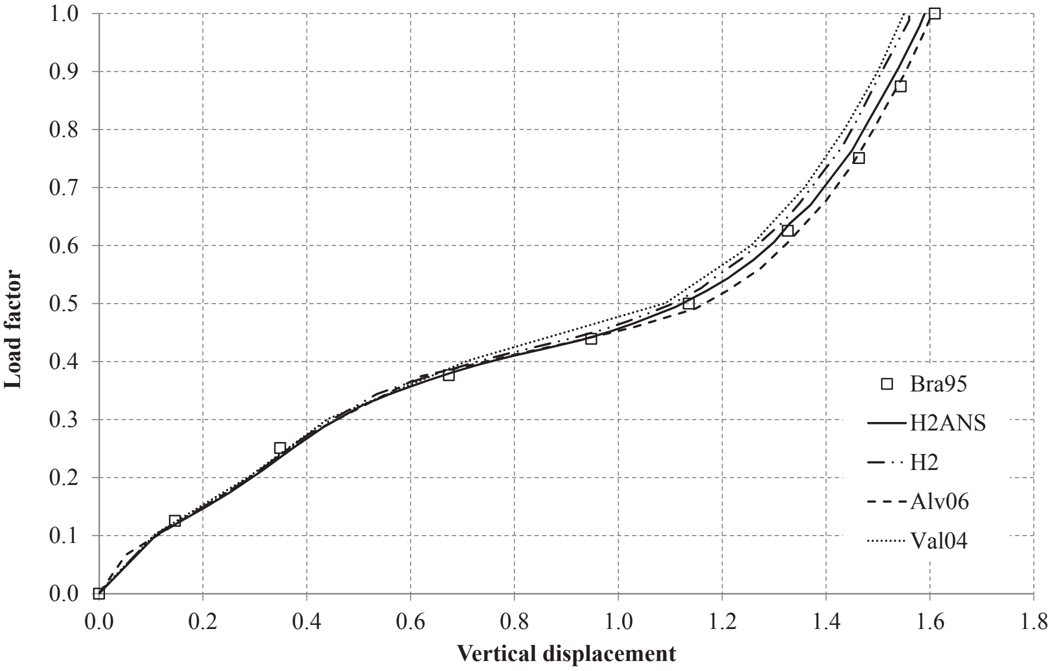


Figure 7.29: Pinching of a clamped cylinder: comparison with formulations available in the literature.

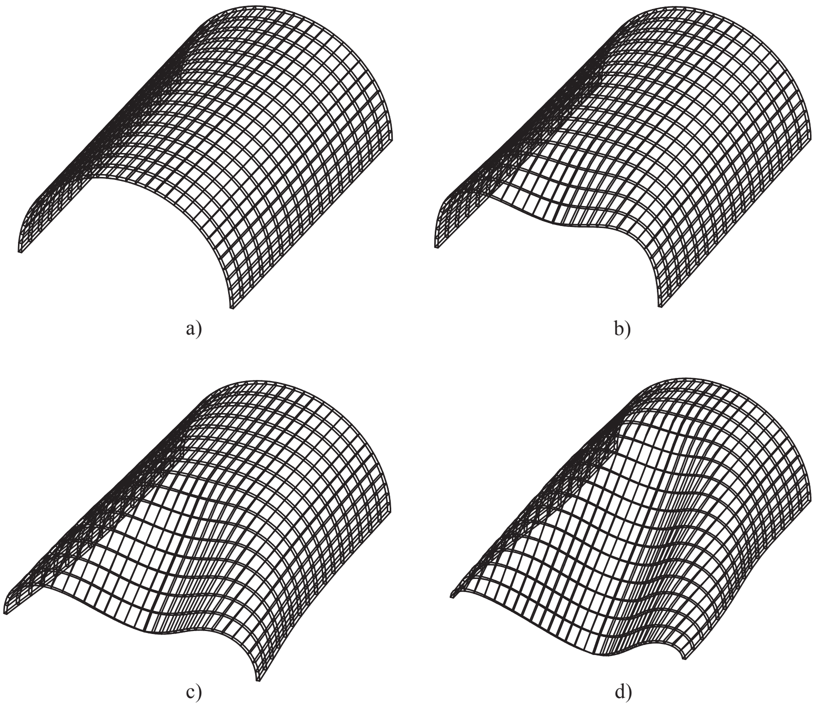


Figure 7.30: Pinching of a clamped cylinder: configuration for a) $\lambda = 0.0$, b) $\lambda = 0.33$, c) $\lambda = 0.42$ and d) $\lambda = 1.0$.

7.2.4 Channel-Section Beam

The present benchmark deals with a U-shaped channel-section beam with one clamped end and subjected to a concentrated load in its free end. The geometry of the beam, schematically represented in Figure 7.31, is defined by a length $L = 36.0$, height $h = 6.0$, width $w = 2.0$ and thickness $t = 0.05$, according to references [Chróścielewski 92, Ibrahimbegović 94, Betsch 96, Eberlein 99, Li 00, Valente 04a]. The elastic properties relate to an elastic modulus $E = 1 \times 10^7$ and a Poisson's coefficient of $\nu = 0.333$. A mesh comprised of $25 \times 36 \times 1$ elements is used, following the work of Valente [Valente 04a].

The load-displacement curve for the H2 and H2ANS formulations are presented in Figure 7.32 and compared with the results coming from the work of Eberlein and Wriggers [Eberlein 99] and Li and Zhan [Li 00]. It can be seen that the solid-shell formulation is able to predict a behaviour which is in good accordance with the reference solutions. The H2 formulation presents a stiffer solution, leading to a slight overestimation of the critical point before the buckling of the beam occurs. The final deformed configuration obtained by the H2ANS element it is presented in Figure 7.33, in which is possible to observe the buckling of the upper flange near the clamped edge and the twisting in the free end. This behaviour is in accordance with the results coming from the literature [Ibrahimbegović 94, Eberlein 99, Li 00].

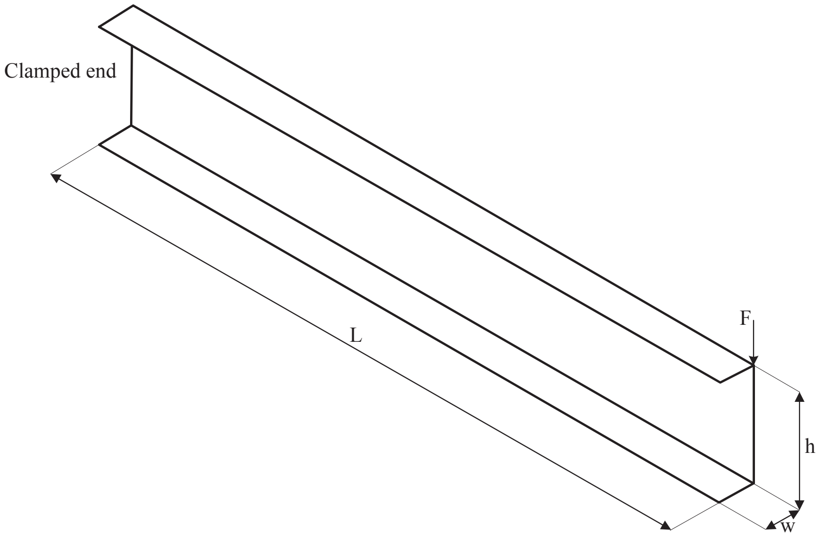


Figure 7.31: Schematic representation of the channel-section beam.

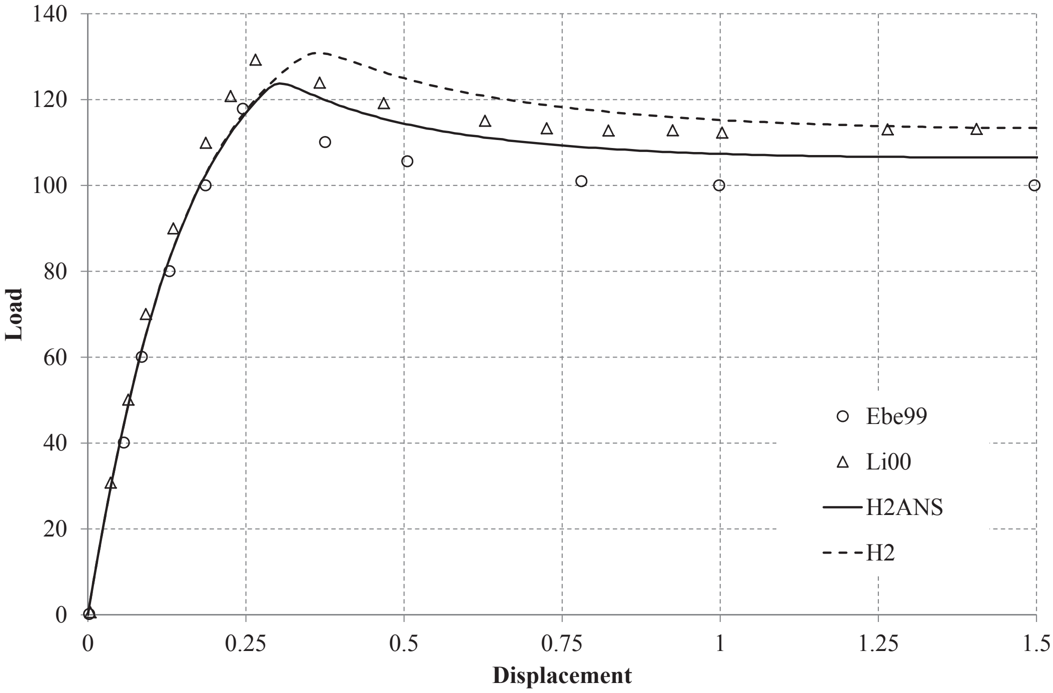


Figure 7.32: Channel-section beam: load-displacement curves.

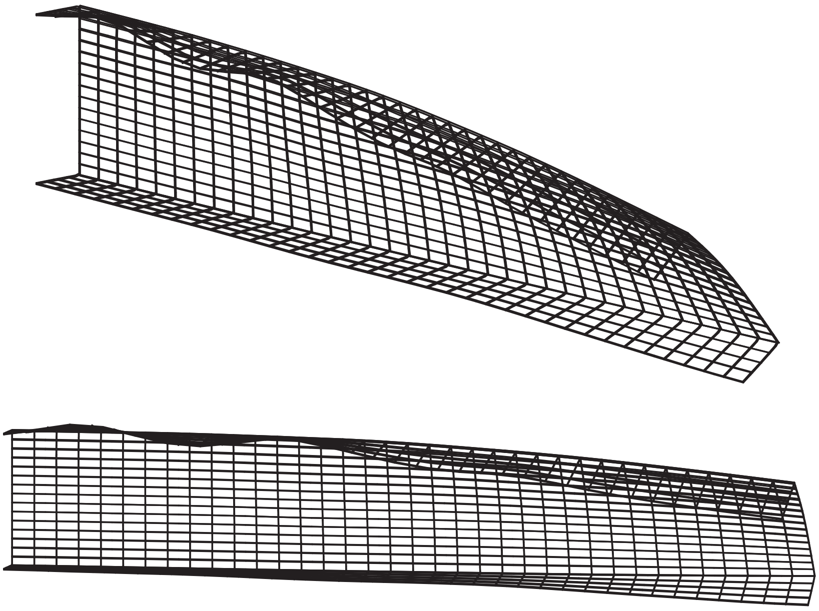


Figure 7.33: Channel-section beam: final deformed configuration.

7.2.5 Cantilever Ring Plate

In this example, a cantilever ring plate is clamped in one end and subjected to a vertical line load on its free end, as shown in Figure 7.34. According to Basar *et al.* [Basar 92], this benchmark is very sensitive in the evaluation of models which involve large rigid body rotations and displacements and has been studied in various works [Buechter 92, Wriggers 93, Brank 95, Sansour 98, Li 00, Valente 03]. The geometry of the model is characterised by an internal $R_i = 6.0$ and external $R_o = 10.0$ radii with a constant thickness of $t = 0.03$. The distributed load has a nominal value of $F = 100.0 \times \lambda$ per unit length, where λ is a load factor. The line load transforms the plane structure into a doubly curved one. The elastic constitutive parameters are the elastic modulus $E = 2.1 \times 10^{10}$ and Poisson's coefficient $\nu = 0.0$.

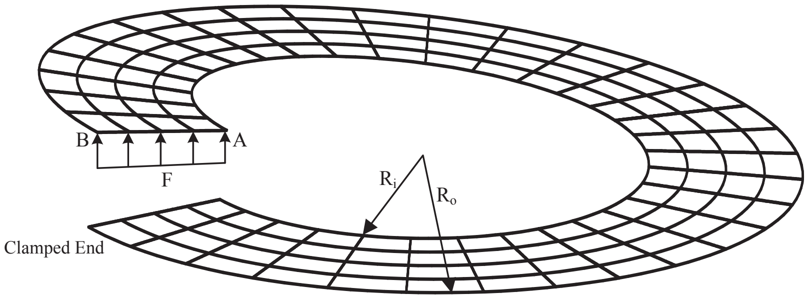


Figure 7.34: Schematic representation of the cantilever ring plate benchmark.

For the solution of the problem two mesh densities are considered: a coarser mesh with

$16 \times 2 \times 1$ elements and a finer mesh with $32 \times 4 \times 1$ (this one seen in Figure 7.34). In Figure 7.35, the displacement along the out-of-plane direction of points A and B as obtained for the H2 and H2ANS formulations are plotted and compared with those obtained by Simo and Rifai [Simo 88] (as reproduced in Basar *et al.* [Basar 92]) and Valente [Valente 03], for a maximum load factor of $\lambda = 2.0$. This load factor is enough for comparison purposes since it corresponds to the load zone where the shape variations are more drastic [Basar 92, Valente 04a]. It can be seen that for the finer mesh the proposed H2ANS formulation is able to obtain results that are in good accordance with the reference solutions. The H2 solid element is not able to correctly reproduce the desired behaviour, leading to an underestimation of the displacement field. In Figure 7.36, the deformed mesh (along with the corresponding control lattice) is depicted for a loading factor $\lambda = 20.0$, qualitatively demonstrating the good performance of the proposed formulation in the presence of large rotations and displacements.

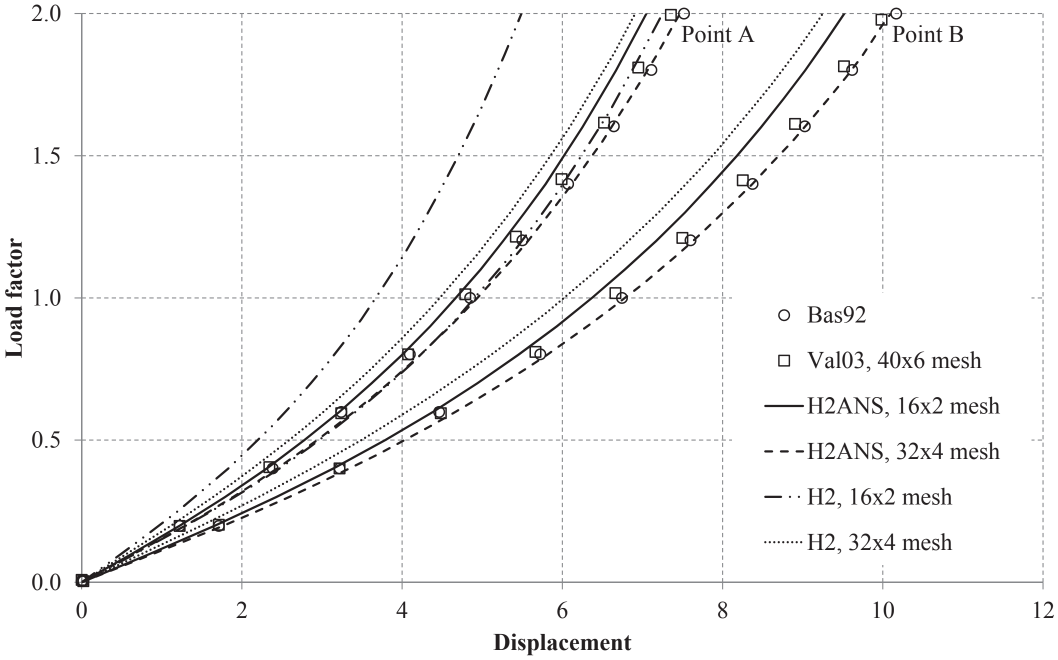


Figure 7.35: Cantilever ring plate: evolution of the displacement of points A and B for a load factor $\lambda = 2.0$.

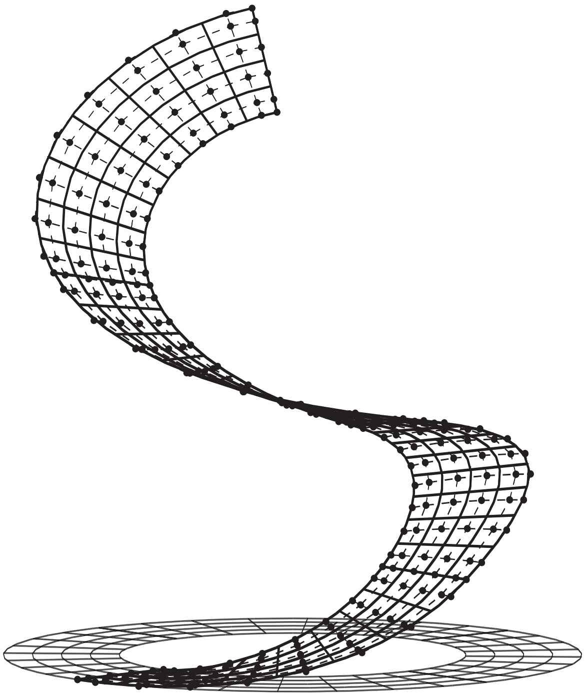


Figure 7.36: Cantilever ring plate: deformed mesh and control lattice for a load factor $\lambda = 20.0$.

7.2.6 Snap-Through Behaviour of a Shallow Roof Structure

In this example, the snap-through and snap-back load-displacement path of a cylindrical structure is analysed. This is a standard benchmark problem used to assess the performance of shell and solid-shell formulations [Horrigmoe 78, Crisfield 81, Cho 98, Eriksson 02, Valente 03, Valente 04b, Alves de Sousa 06b, Schwarze 11], and a schematic representation of the structure can be seen in Figure 7.37. Due to symmetry conditions, only a quarter of the structure is modelled. Following references [Valente 04b, Alves de Sousa 06b], the geometry of the model is defined by the parameters $L_1 = 508.0, L_2 = 507.15$, radius $R = 2540$ and thickness $t = 6.35$. The material is defined by the elastic modulus $E = 3102.75$ and the Poisson's coefficient $\nu = 0.3$. The load applied at the centre of the structure (point A) has a magnitude of $F = 1000.0$.

The load-displacement curves for points A and B are presented in Figure 7.38 and compared with solutions coming from the the works of Horrigmoe and Bergan [Horrigmoe 78] and Schwarze and Reese [Schwarze 11]. In the current example two mesh densities are considered: a coarse mesh composed of $2 \times 2 \times 1$ elements and a finer one with $5 \times 5 \times 1$ elements (shown in Figure 7.37). It can be seen that the H2ANS formulation is able to reproduce the behaviour of the structure, even when considering the coarse mesh. These results are in good agreement with the reference solutions, leading to a correct prediction of the snap-through and snap-back effects. On the other hand, the conventional H2 formulation presents a very stiff response when the coarse mesh is employed, resulting in an overestimation of the critical load for which the snap-trough behaviour of the structure occurs. This overestimation is alleviated when a finer mesh is considered.

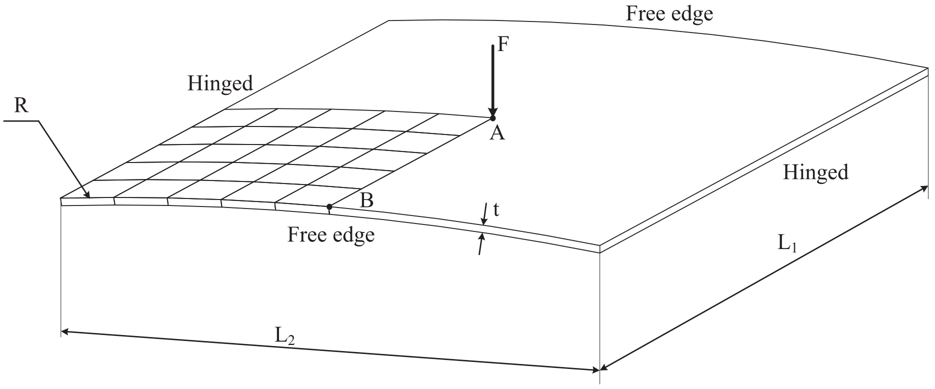


Figure 7.37: Schematic representation of the shallow roof structure.

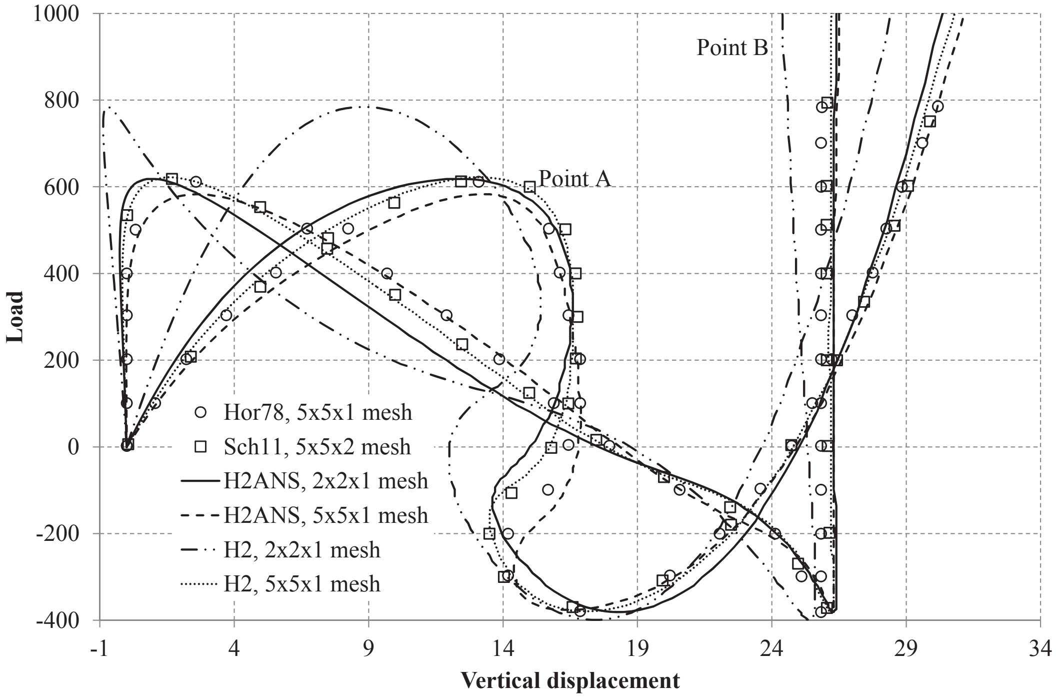


Figure 7.38: Shallow roof structure: load-displacement curves for Points A and B.

7.2.7 Elastic and Elasto-Plastic Stretch of a Cylinder with Free Edges

In this example, a cylindrical shell with free edges, depicted in Figure 7.39, is deformed under the action of two opposite pulling loads, inducing large rotations and displacements. This popular benchmark has been analysed in the literature considering both elastic [Sansour 92, Brank 95, Masud 00b, Valente 03, Valente 04b, Sze 04, Schwarze 11, Hosseini 13] and elasto-plastic [Masud 00a, Valente 04b] constitutive relations. The geometry of the problem is defined by the length $L = 10.35$, radius $R = 4.953$ and thickness $t = 0.094$. The structure is subjected to a pair of concentrated loads with magnitude $F = 40000 \times \lambda$, where λ is a load factor ranging from 0.0 to 1.0. Due to symmetry conditions, only one eighth of the structure needs to be modelled. The material properties are given by an elastic modulus $E = 10.5 \times 10^6$ and Poisson's coefficient $\nu = 0.3125$. The elasto-plastic behaviour is defined by the yield stress $\sigma_0 = 1.05 \times 10^5$ and a linear isotropic hardening coefficient of $H = 10.5 \times 10^5$. A mesh consisting of $16 \times 8 \times 1$ is employed in both cases, as shown in Figure 7.39.

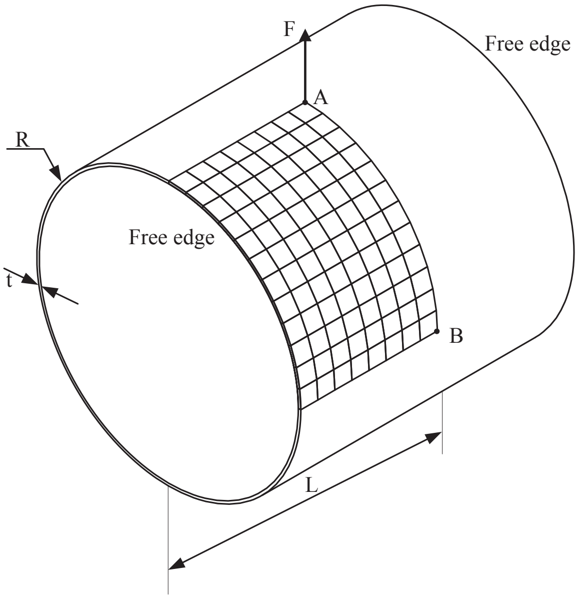


Figure 7.39: Schematic representation of the stretch of a cylinder benchmark.

The obtained load-displacement curves obtained for points A and B for the elastic case can be seen in Figure 7.40, while the results for the elasto-plastic counterpart version are presented in Figure 7.41. In these figures, reference values taken from the works of Hosseini *et al.* [Hosseini 13], Schwarze and Reese [Schwarze 11], Valente *et al.* [Valente 04b], Sze *et al.* [Sze 04], Masud *et al.* [Masud 00b] and Masud and Tham [Masud 00a] are also provided. As can be seen, in both the elastic and elasto-plastic regime, the H2ANS element follows the results from Sze *et al.* [Sze 04] and Masud and Tham [Masud 00a], respectively. It is worth noting that in the latter case, the H2 NURBS-based solid element presents convergence difficulties after the application of 95% of the total load, being observable the excessive displacement obtained for point A. The final deformed configurations obtained by the H2ANS NURBS-based element can be seen in Figure 7.42.

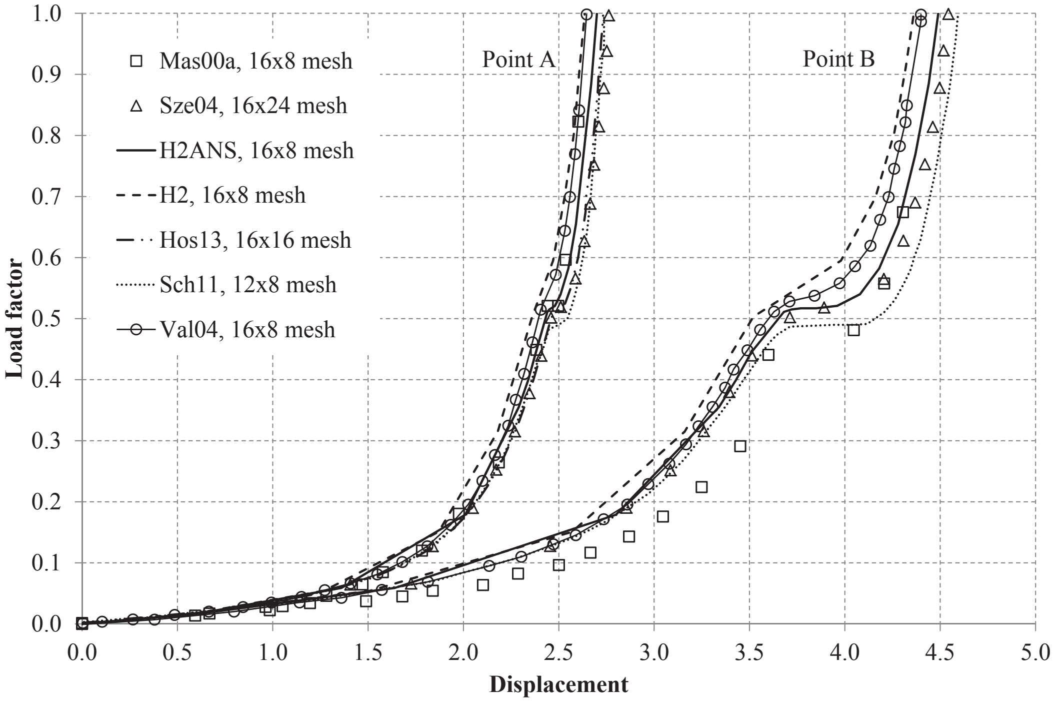


Figure 7.40: Elastic stretch of a cylinder: load-displacement curves for points A and B.

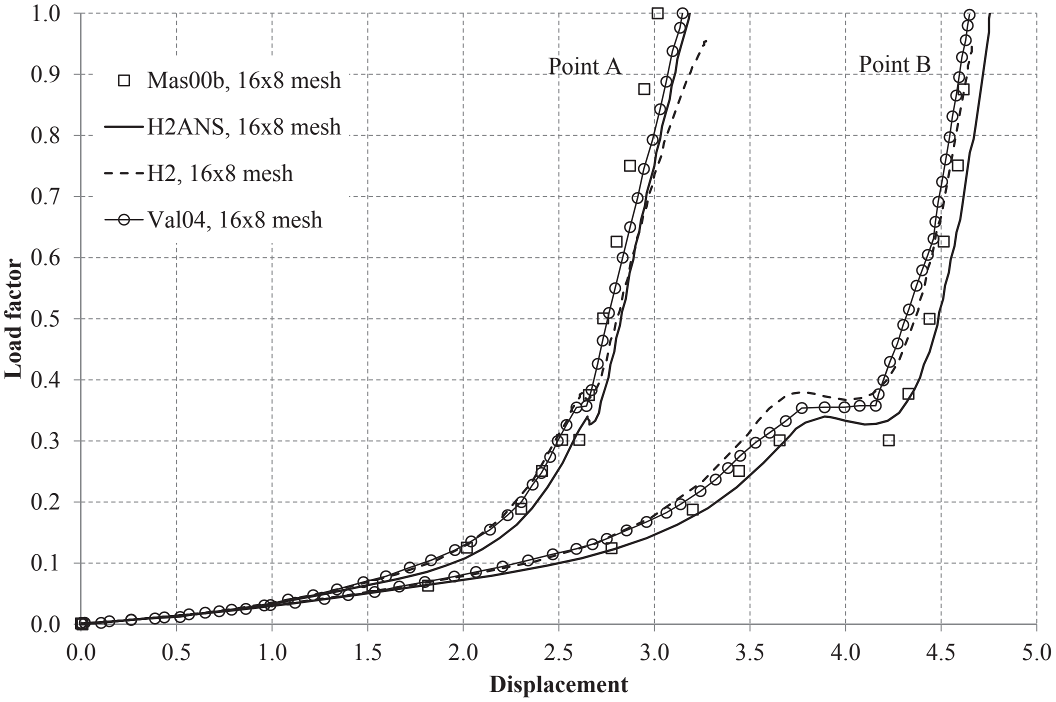


Figure 7.41: Elasto-plastic stretch of a cylinder: load-displacement curves for points A and B.

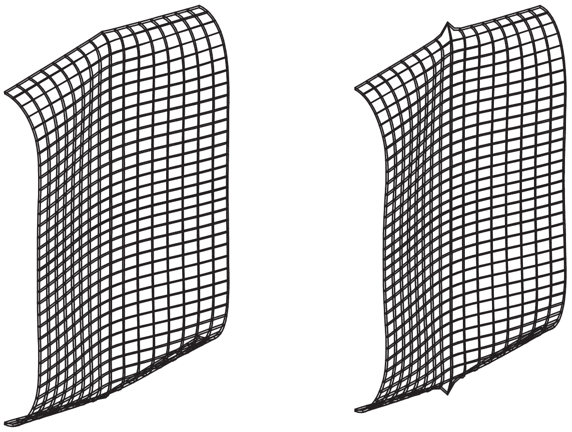


Figure 7.42: Elasto-plastic stretch of a cylinder: deformed configuration considering (left) elastic and (right) elasto-plastic constitutive relations.

7.2.8 Elastic and Elasto-Plastic Analysis of a Hemispherical Shell with 18° Hole

This example deals with a doubled curved shell with a 18° hole subjected to a pair of concentrated loads applied at antipodal points, leading to a problem dominated by large rotations. A schematic representation of one quarter of the structure can be seen in Figure 7.43 where the equator plane represents a free edge. The geometry of the problem is defined by a radius $R = 10.0$ and thickness $t = 0.04$. In the following, and as done for the previous example, this benchmark problem is analysed considering both elastic [Simo 90a, Liu 98, Sansour 00, Masud 00b, Sze 02, Kim 05, Schwarze 11] and elasto-plastic [Masud 00a, Valente 04b] constitutive relations.

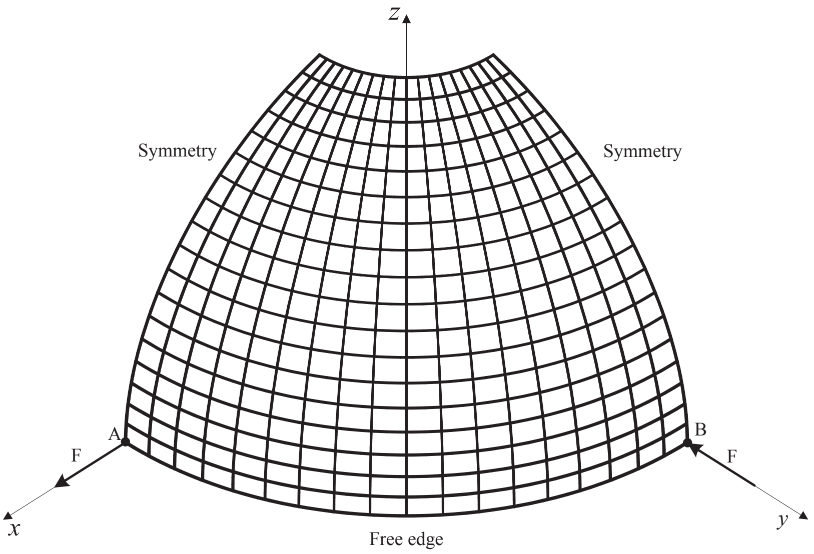


Figure 7.43: Schematic representation of one quarter of the hemispherical shell with 18° hole.

Focusing first on the elastic case (but accounting large deformations), the constitutive parameters relate to an elastic modulus $E = 6.825 \times 10^7$ and Poisson's coefficient $\nu = 0.3$. The load is given by $F = 1.0 \times \lambda$, where the load factor is set to $\lambda = 100.0$, and the total load is applied in 10 equal steps. The displacements of points A and B for different mesh densities are presented in Figure 7.44 and 7.45, respectively, and compared with reference results coming from the works of Simo *et al.* [Simo 90a] and Sansour and Kollmann [Sansour 00], both considering a 16×16 mesh. It can be seen that for both the 16×16 and 18×18 meshes, the results coming from the H2ANS solid-shell formulation are in good agreement with the reference solutions coming from the literature. On the contrary, the second-order NURBS-based solid element (H2) presents a very stiff behaviour due to locking effects, leading to an underestimation of the displacements of points A and B, even when considering the refined mesh.

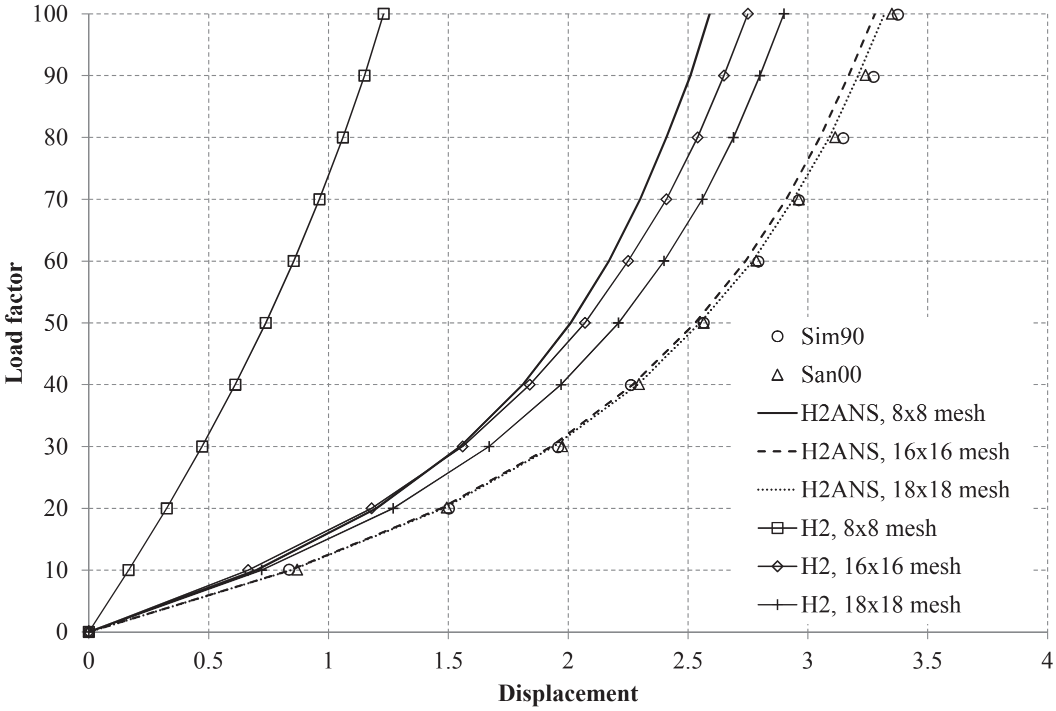


Figure 7.44: Hemispherical shell with 18° hole: displacement for point A for the elastic case.

In the following, the same problem is analysed within the material nonlinear range. To this end, an initial yield stress $\sigma_0 = 6.825 \times 10^5$ and a linear isotropic hardening coefficient of $H = 6.825 \times 10^6$ are introduced in the model, the load now being given by $F = 0.5 \times \lambda$, for a load factor $\lambda = 400.0$. The displacement of points A and B are presented in Figure 7.46, along with reference solutions from the works of Masud and Tham [Masud 00a] and Valente *et al.* [Valente 04b]. The H2ANS formulation is able to provide results with a good agreement with the reference solution presented by Masud and Tham [Masud 00a] when considering a mesh composed of $18 \times 18 \times 1$ elements. The final deformed shape for the

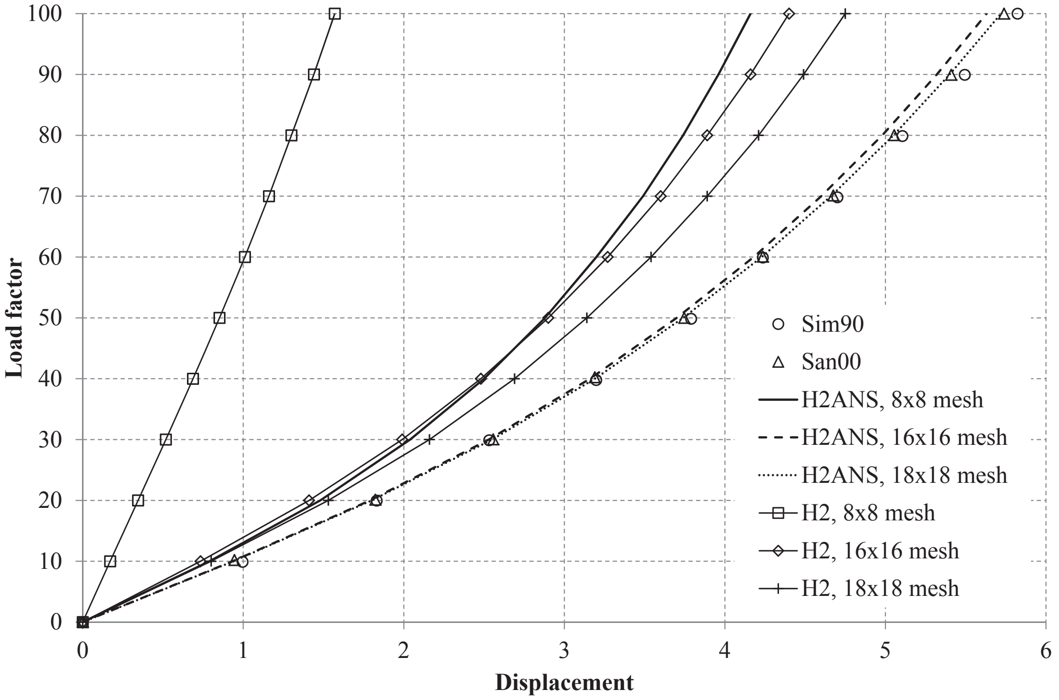


Figure 7.45: Hemispherical shell with 18° hole: displacement for point B for the elastic case.

finer mesh is presented in Figure 7.47.

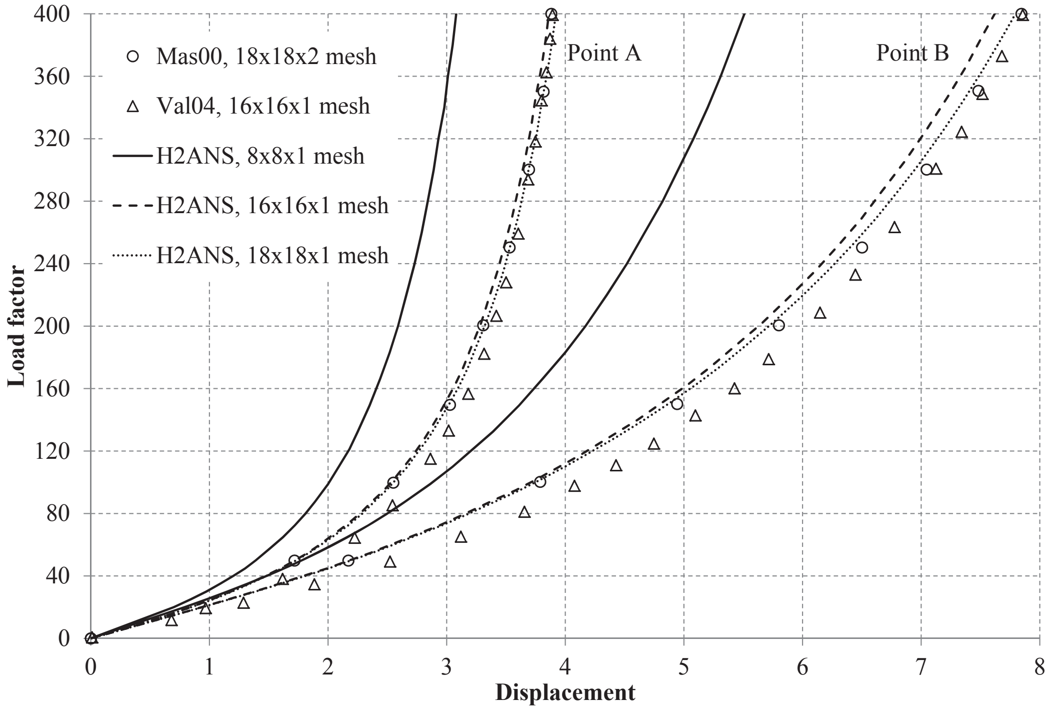


Figure 7.46: Hemispherical shell with 18° hole: displacement for points A and B for the elasto-plastic case.

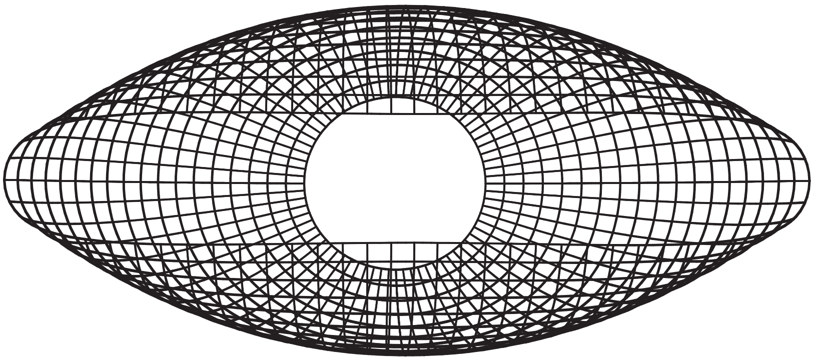


Figure 7.47: Elasto-plastic hemispherical shell with 18° hole: deformed configuration.

7.2.9 Geometry and Material Nonlinear Analysis of a Pinched Cylinder

In this example, the pinched cylinder problem presented in the fully linear context in Section 7.1.5 is now analysed considering both geometric and material nonlinear effects. This is a classical test to assess the performance of a formulation in the presence of localized plasticity and significant shape modifications [Wriggers 96, Hauptmann 98, Miehe 98, Eberlein 99, Valente 04b]. As in its linear counterpart, the geometry of the problem is defined by a length $L = 600.0$, radius $R = 300.0$ and thickness $t = 3.0$ (Figure 7.14). Both ends of the cylinder are constrained in order to maintain their circular shape, but allowing for a longitudinal displacement. The elasto-plastic constitutive relation is defined by a bulk modulus $\kappa = 2500.0$, shear modulus $G = 1154.0$, initial yield stress $\sigma_0 = 24.3$ and a linear isotropic hardening coefficient $H = 300.0$. A total load of $F = 2.0 \times \lambda$ is applied, where λ ranges from 0.0 to 5000.0. Two mesh densities are considered: a coarse mesh comprised of $16 \times 16 \times 1$ elements and a more refined one with $32 \times 32 \times 1$ elements, over one eighth of the total structure.

The results for the displacement of the loaded point are plotted in the Figure 7.48, along with reference solutions from Wriggers *et al.* [Wriggers 96], Miehe [Miehe 98] and Eberlein and Wriggers [Eberlein 99]. In can be seen that the load-deflection path for the two mesh densities obtained by the present formulation are in good agreement with the numerical results from the works of Miehe [Miehe 98] and Eberlein and Wriggers [Eberlein 99], specially for the refined mesh. It can also be seen that the H2ANS formulation does not present non-smooth curves which are representative of a snap-through like behaviour usually present in this benchmark when a coarse mesh is employed [Hauptmann 98]. The standard H2 solid element presents high convergence difficulties when dealing with this benchmark and was not able to complete the numerical simulation, even for the finer mesh. For this reason, the results for the H2 element are not presented in the plots. In Figure 7.49, the deformed shapes corresponding to a maximum vertical tip displacement up to approximately

300.0 consistent units (the same value of the radius of the structure) are depicted for the $16 \times 16 \times 1$ mesh.

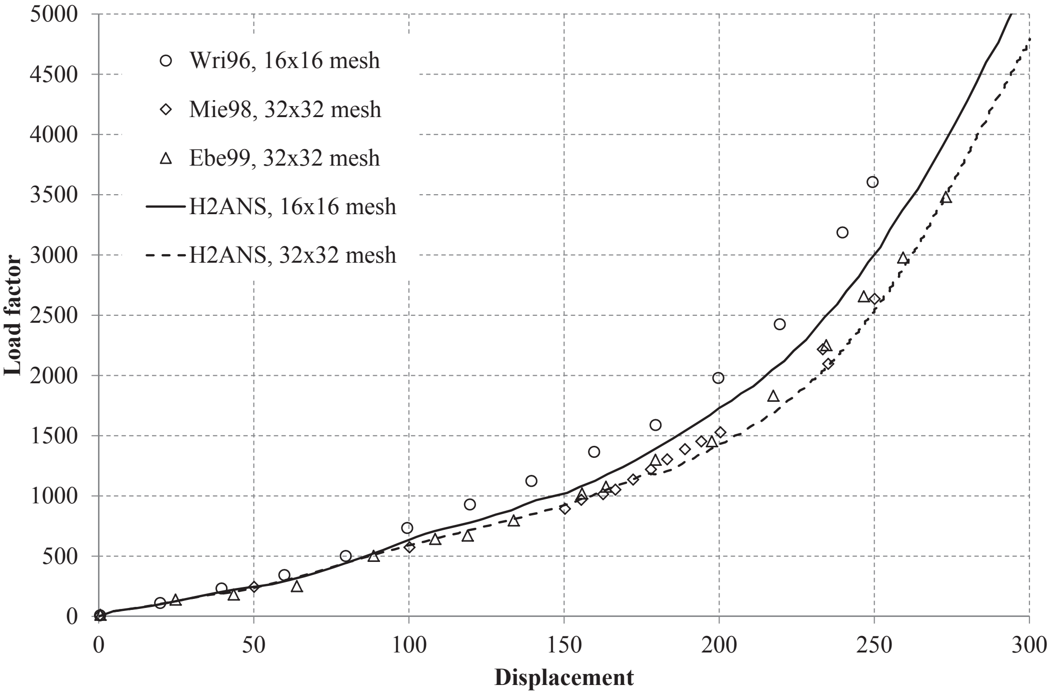


Figure 7.48: Geometry and material nonlinear analysis of a pinched cylinder: displacement curve for different mesh densities.

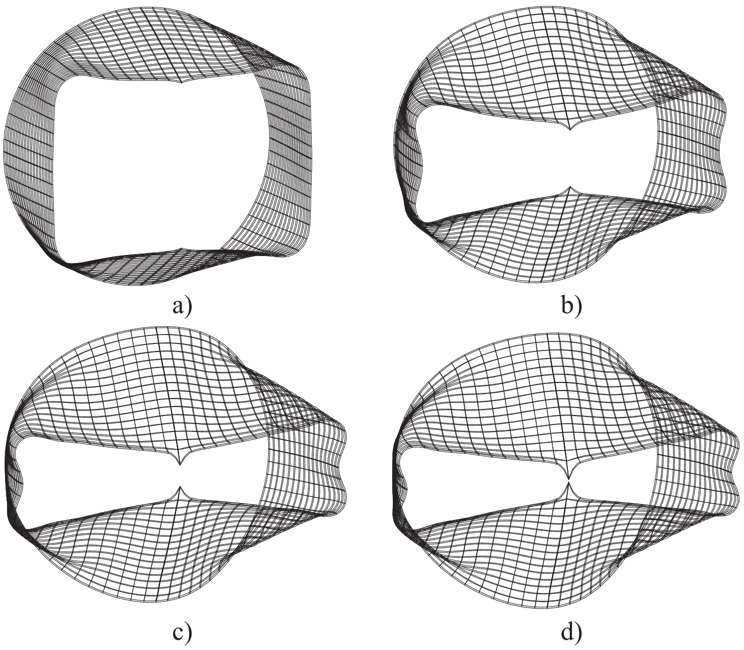


Figure 7.49: Geometry and material nonlinear analysis of a pinched cylinder: deformed mesh for tip displacement of a) $w \approx 120.0$, b) $w \approx 240.0$, c) $w \approx 275.0$ and d) $w \approx 300.0$.

7.2.10 Elasto-Plastic Full Hemispherical Shell

As a last example, the full hemispherical shell analysed within the linear regime in Section 7.1.4 is now modelled considering both geometric and material nonlinearities. The geometry of the structure is defined by a radius $R = 10.0$ and thickness $t = 0.5$. Following references [Eberlein 99, Klinkel 06, Schwarze 11], the elastic material properties are defined by an elastic modulus $E = 10.0$, and a Poisson's coefficient $\nu = 0.2$, while the plastic behaviour of the material is defined by the yield stress $\sigma_0 = 0.2$ and an isotropic hardening coefficient $H = 9.0$. The load magnitude is given as $F = 0.04$.

The results for the load-displacement curves for points A and B located in the inner and outer surfaces are presented in Figures 7.50 and 7.51, respectively. These curves are compared with those coming from the works of Schwarze and Reese [Schwarze 11], Klinkel *et al.* [Klinkel 06] and Eberlein and Wriggers [Eberlein 99]. As can be seen, load-displacement curves obtained by the proposed H2ANS formulations are in good agreement with the results from Klinkel *et al.* [Klinkel 06]. The H2 formulation, however, is affected by locking and, consequently, presents a stiffer behaviour.

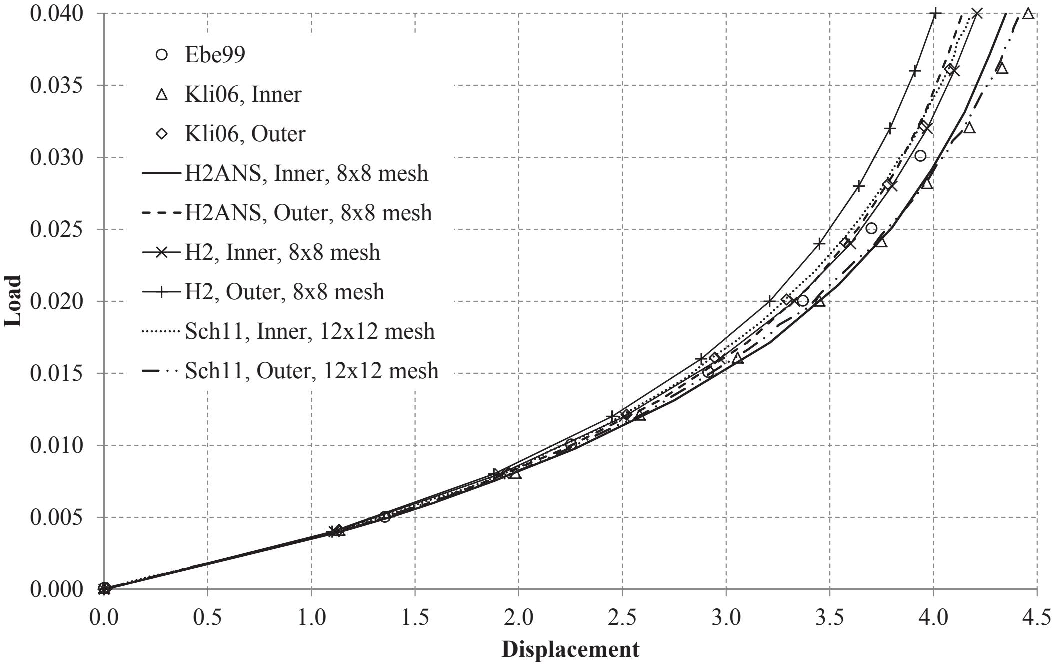


Figure 7.50: Full hemispherical shell problem: load-displacement curves for point A.

Finally, a comparison between the NURBS-based cubic solid element (H3) and the H2ANS solid-shell element is performed. The load-displacement curves for points A and B obtained by both formulations using a $8 \times 8 \times 1$ mesh are presented in Figure 7.52, showing that both elements present a very similar performance. The CPU times obtained by the formulations is depicted in Figure 7.53, where the results from the quadratic NURBS-based

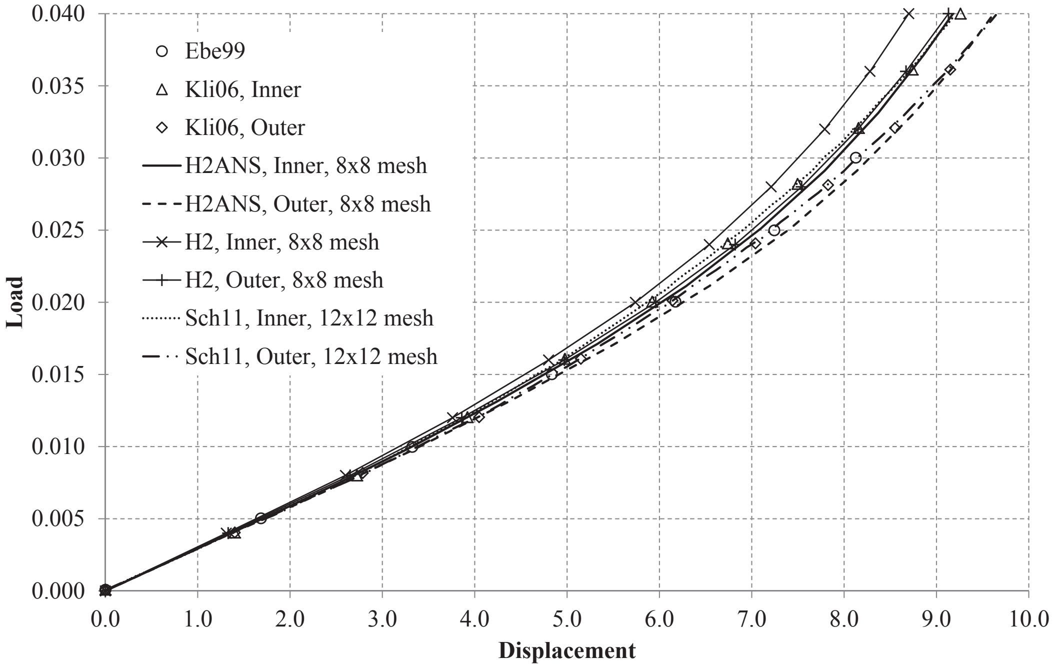


Figure 7.51: Full hemispherical shell problem: load-displacement curves for point B.

elements are also presented for comparison purposes. The results are normalised using the CPU time obtained by the H3 element with a mesh composed of 1444 control points, corresponding to $16 \times 16 \times 1$ elements. As can be seen, the H2ANS element presents a significantly lower CPU time when compared with the cubic solid NURBS-based element, while maintaining a very similar prediction of the load-displacement curves of the analysed points, making the H2ANS a more efficient choice for the numerical simulation of this problem.

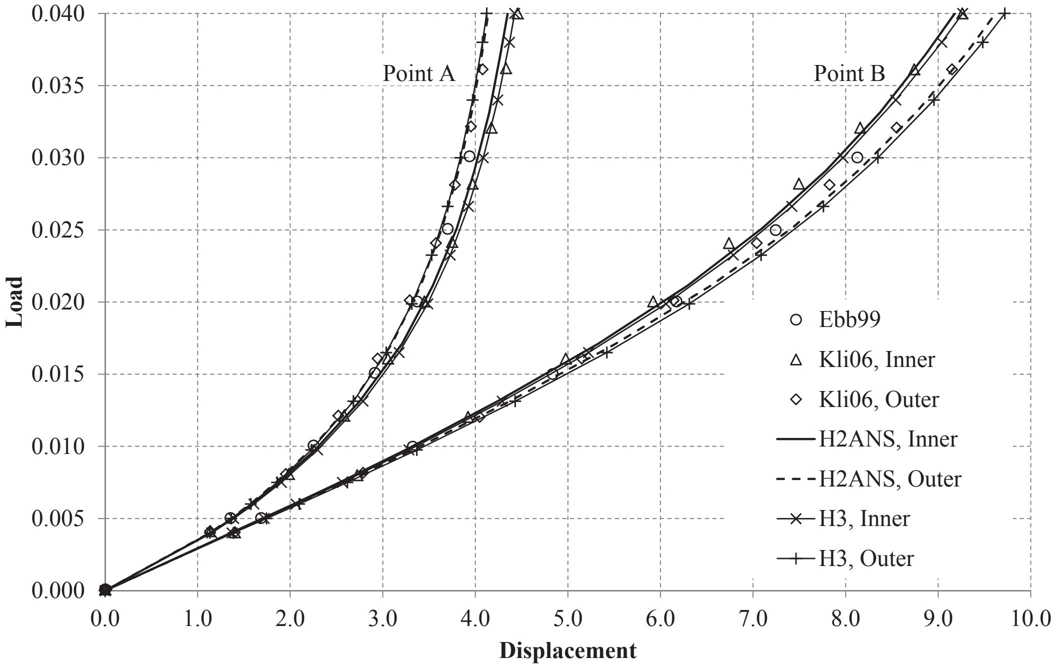


Figure 7.52: Full hemispherical shell problem: comparison between the H3 and H2ANS elements.

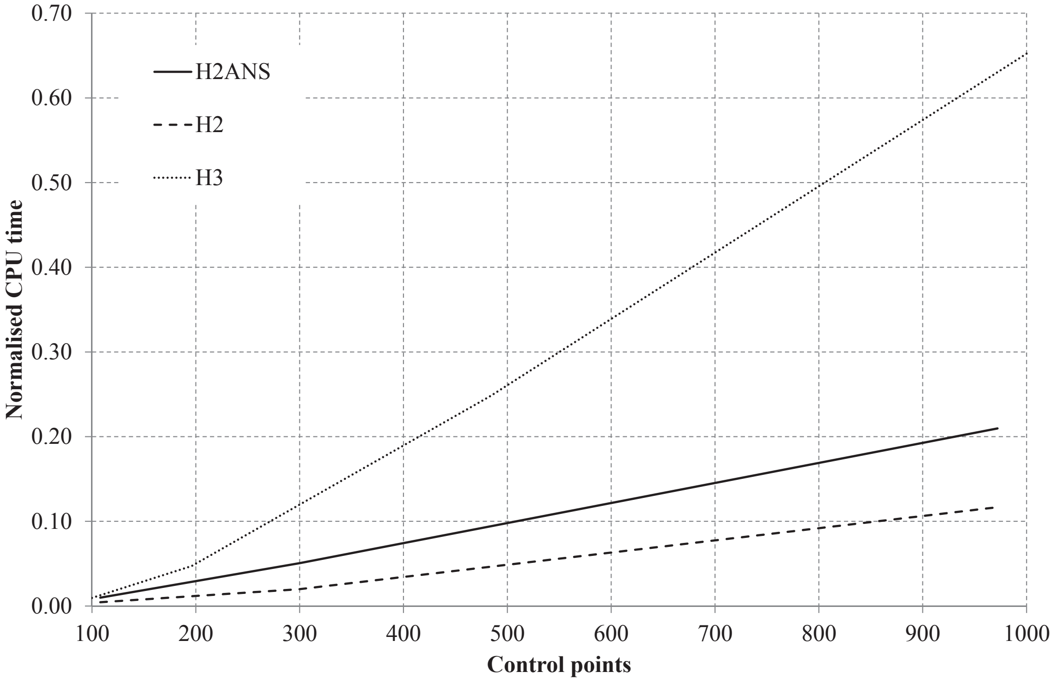


Figure 7.53: Full hemispherical shell problem: comparison of CPU time.

7.3 Problems Dealing With Volumetric Locking

The following numerical examples will be used to assess the performance of the NURBS-based formulation proposed in Section 5.6 (denoted as H2PV) when volumetric locking effects are present.

7.3.1 The Curved Beam

The first example deals with a curved beam which is subjected to a uniform radial displacement on its free edge, as can be seen in Figure 7.54. The geometry of the beam if defined by an outer and inner radii given by $R_\text{o} = 10.0$ and $R_\text{i} = 5.0$, respectively. The radial displacement is set to be equal to $u_0 = 0.1$, while the boundary conditions are defined as $u(0,y) = v(0,R_\text{i}) = 0$. The material properties are given by the elastic modulus $E = 9600.0$ and Poisson's coefficient $\nu = 0.4995$, and plane strain conditions are assumed [Taylor 11].

The strain energy error of the numerical solution versus the element size is presented in Figure 7.55 for the H2, H2ANS and H2PV NURBS-based formulations. It can be seen that the H2PV element is able to significantly improve the behaviour of the standard quadratic solid element. For the finer mesh, composed of 16×16 elements, the error in the strain energy obtained by the H2PV element is more than one order of magnitude lower than the one obtained by the H2.

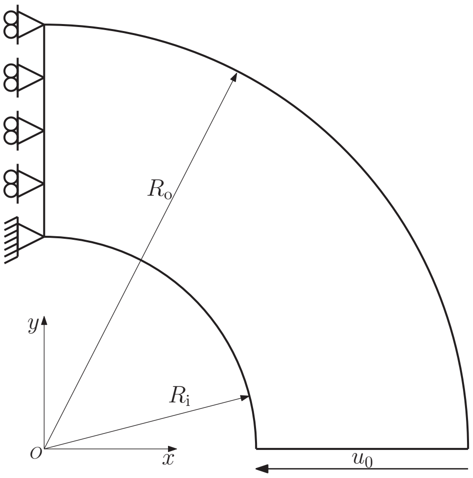


Figure 7.54: Schematic representation of the curved beam.

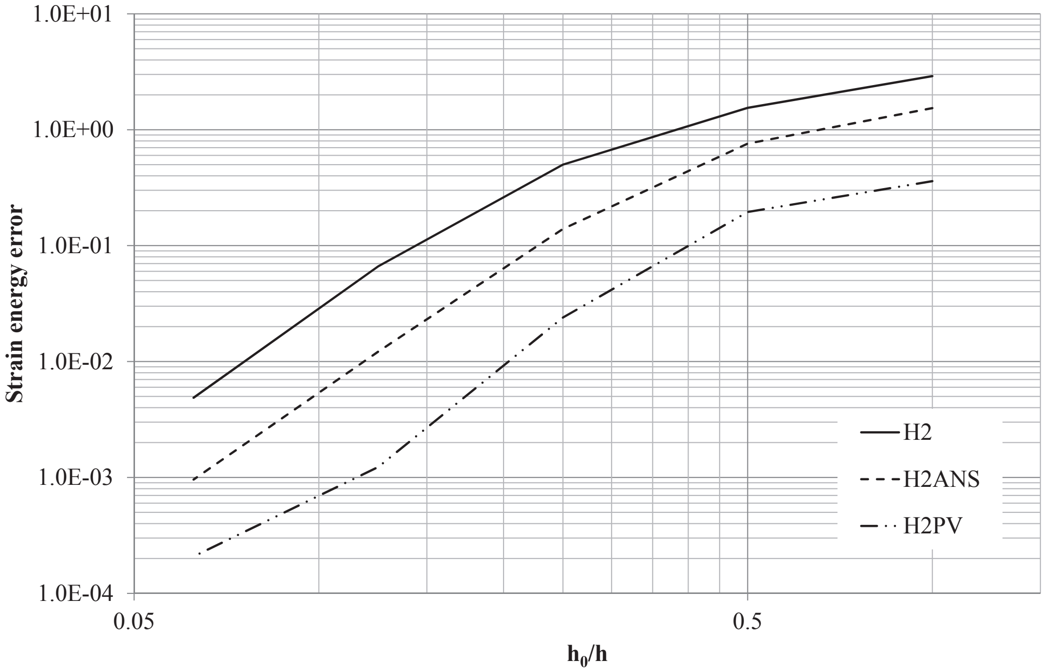


Figure 7.55: Curved beam: strain energy error versus element size.

7.3.2 The Cook's Membrane

The Cook's membrane is a well-known benchmark test to assess the performance of a given formulation in the near incompressible case [Simo 90b, Ibrahimbegović 90, César de Sá 99, Kasper 00]. This problem consists of a skew plate which is clamped on one side and subjected to a shear load $F = 100.0$ on the opposite edge. The geometry of the problem is given in Figure 7.56. The constitutive parameters are taken as $E = 240.565$ and $\nu = 0.4999$. Once again, plane strain conditions are considered.

The vertical displacements of point A obtained by different NURBS-based formulations are presented in Figure 7.57. For comparison purposes, the results obtained by Elguedj *et al.* [Elguedj 08] for a quadratic $\overline{\mathbf{B}}$ patch are also presented. The H2PV element is able to significantly improve the behaviour of the standard quadratic solid element, showing a performance that is closer to the one of the cubic element. The H2ANS formulation presents a performance similar to the one of the H2 element, demonstrating that the ANS methodology is not adequate to tackle volumetric locking. The second-order $\overline{\mathbf{B}}$ NURBS-based element presents the best overall performance. This was to be expected since this methodology is applied at the patch level, while the method proposed herein operates at the element level. However, and particularly to this element-wise approach, the proposed methodology may prove to be easier to implement into available finite element codes by means of user subroutines.

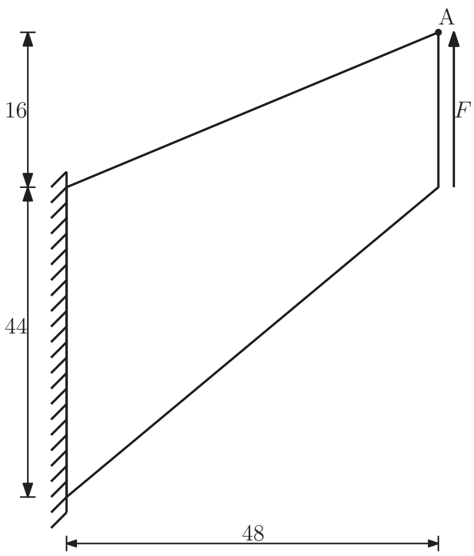


Figure 7.56: Cook's membrane problem setup.

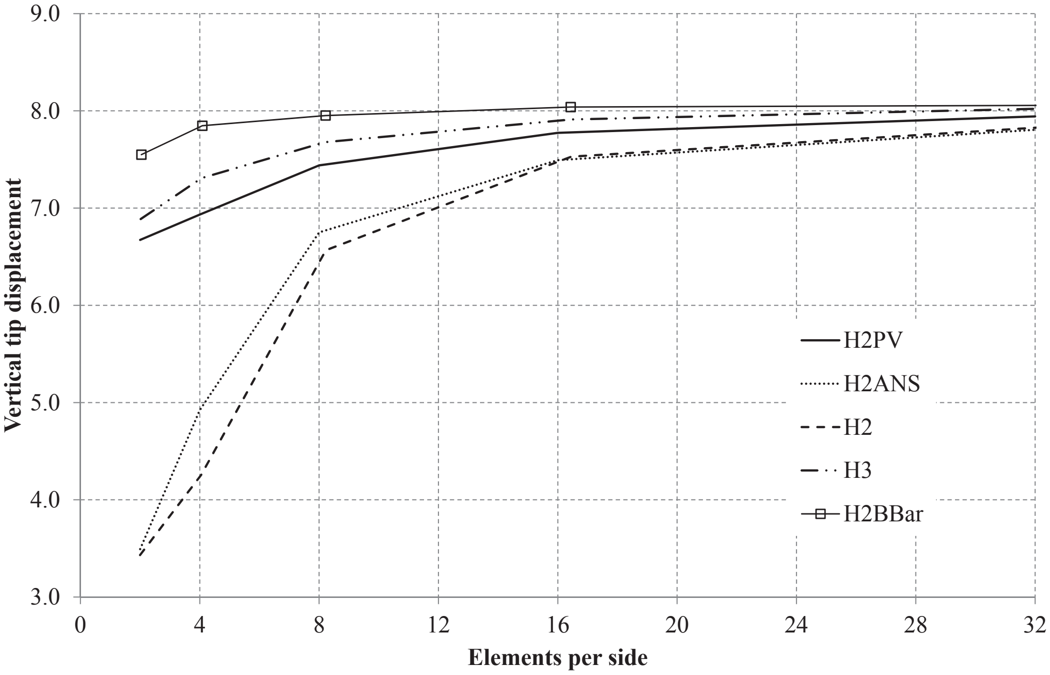


Figure 7.57: Cook's membrane: vertical tip displacement (point A).

7.4 Contact Benchmark Problems

In this section, a set of numerical examples are used to validate the implementation and assess the accuracy of the Point-to-Segment formulation described in Chapter 6. All the examples presented herein are referred to the linear elastic range.

7.4.1 The Contact Patch Test

The contact patch test, introduced in [Taylor 91], is a well-known benchmark problem to assess the ability of the contact formulation to exactly transmit constant normal tractions over two different bodies. If the patch test is not satisfied, the algorithms introduce errors at the contacting surfaces that do not necessarily decrease with mesh refinement [Crisfield 00, El-Abbasi 01, Zavarise 09a]. In [Taylor 91], it was demonstrated that the classical Node-to-Segment algorithms do not pass the patch test. This can be remedied by employing a two-pass algorithm [Zavarise 09a].

In this work, the setup proposed by Crisfield [Crisfield 00] and presented in Figure 7.58 is considered. Instead of a distributed load, the top surface of the upper body is subjected to a prescribed displacement Δu. The exact solution of the problem's stress field is given by

$$\sigma_{xx} = \tau_{xy} = 0, \tag{7.3}$$

$$\sigma_{yy} = \frac{E}{1-\nu^2}\Delta u, \tag{7.4}$$

$$\sigma_{zz} = \nu\sigma_{yy}. \tag{7.5}$$

The Elastic modulus is taken as $E = 1000.0$, the Poisson's coefficient as $\nu = 0.0$ and $\Delta u = 0.001$. The upper body is considered to be the slave while the lower body is the master. The Finite Element discretisation adopted can be seen in Figure 7.58, where the contact collocation points are represented by diamond symbols.

The contact stresses at the interface for the PTS algorithm is presented in Figure 7.59, along with the curves obtained for the NTS algorithm using fully integrated linear (CPE4) and quadratic (CPE8) Lagrangian-based elements available in the commercial Finite Element software Abaqus. It can be seen that the PTS contact formulation is not able to exactly satisfy the contact patch test. However, it is worth mentioning that the maximum and minimum stress along the horizontal direction at the integration points are $\sigma_{yy}^{\max} \approx -0.997$ and $\sigma_{yy}^{\min} \approx -1.002$, respectively. Taking into account that the reference solution is $\sigma_{yy} = -1.0$, it can be stated that the PTS methodology offers a significant improvement over the classical NTS algorithm typically employed in the context of Finite Element Analysis, since the quadratic Lagrangian-based elements fail the contact patch test analysed herein and introduce quite substantial errors [Crisfield 00]. As can be seen, the PTS algorithm presents oscillations

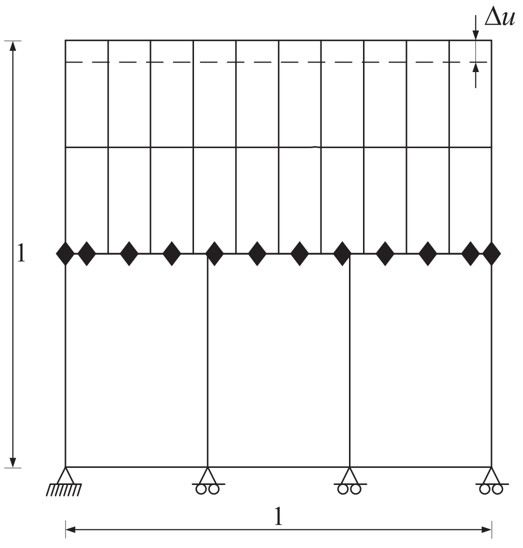


Figure 7.58: Contact patch test problem setup (diamond symbols represent contact collocation points).

of much lower magnitude about the reference solution, when compared to the algorithm of Abaqus, even for higher-order approximations.

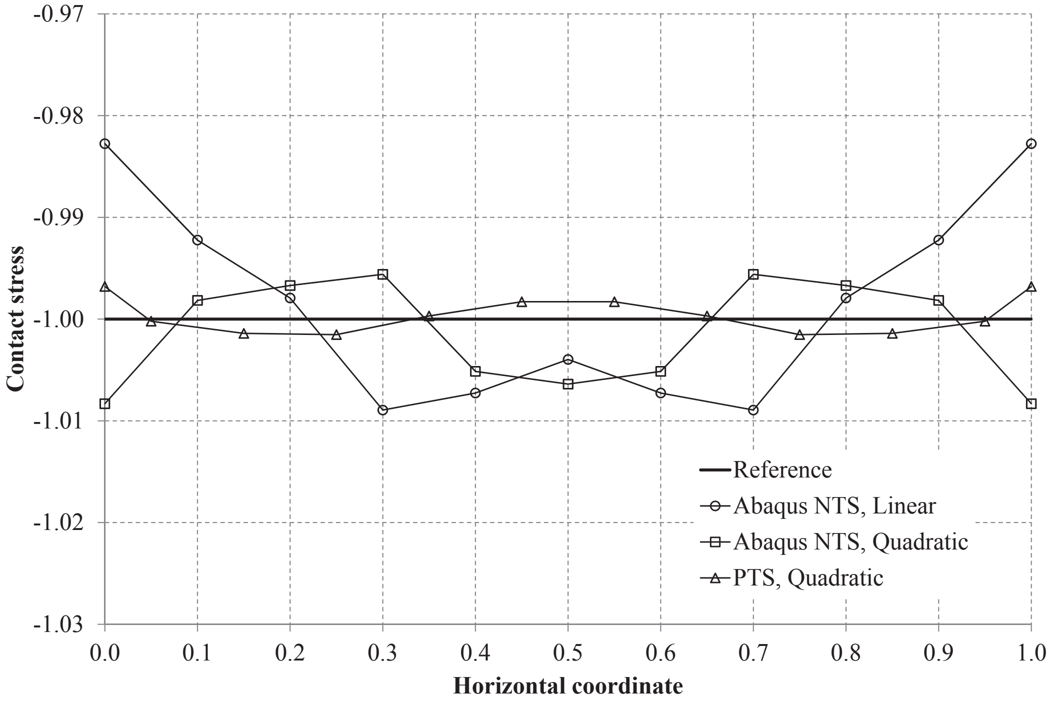


Figure 7.59: Contact patch test: contact stress at the interface.

7.4.2 Indentation of an Elastic Block by a Circular Rigid Punch

This example deals with the indentation of an elastic foundation by a circular rigid punch, as represented in Figure 7.60. The material constitutive parameters are referred to an elastic modulus $E = 1000.0$ and Poisson's coefficient $\nu = 0.3$, and plane strain conditions are considered. The radius of the tool is given as $R = 8.0$ and the depth of indentation is set to $d = 0.6$, where the foundation is defined by a length $L = 16.0$ and height $h = 4.0$ [Kikuchi 88]. Due to symmetry conditions, only half of the model is considered.

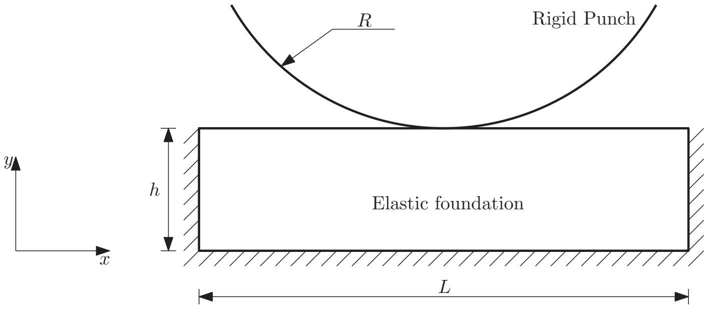


Figure 7.60: Setup of the indentation of an elastic block by a circular rigid punch.

The values of the strain energy for different mesh densities are presented in Table 7.1 for a quadratic NURBS-based formulation. Since no analytical solution is available, a reference solution of 109.513 is obtained using the commercial software Abaqus employing a mesh consisting of 125×125 quadratic quadrilateral elements. It can be seen that, although some small oscillations are obtained, the PTS algorithm is able to converge to the reference solution quite rapidly. The final deformed configuration can be seen in Figure 7.61, for two mesh densities.

Table 7.1: Normalised strain energy for the indentation of an elastic block by a circular rigid punch.

Number of Elements	Number of dof's	Normalised Strain Energy
4×4	36	0.984
5×5	49	0.970
6×6	64	0.998
10×10	144	1.005
14×14	256	0.999

In the second part of the current example, the performance of the methodology presented in Section 6.3.4 to compute the contact stresses is assessed. The results for 3 mesh densities, as well as the reference solution coming from Abaqus, are presented in Figure 7.62. The coarser mesh comprised of 6×6 elements is able to correctly approximate the contact stress at the centre of the elastic foundation. However, since only 3 collocation points are

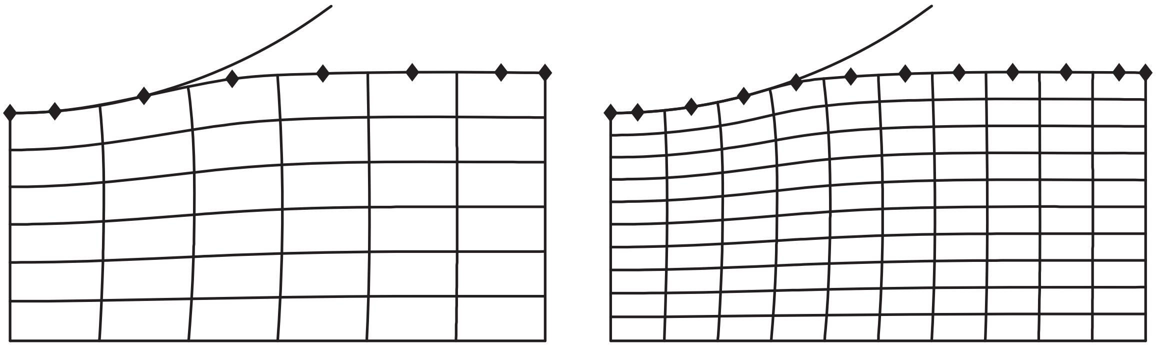


Figure 7.61: Indentation of an elastic block by a circular rigid punch: deformed configuration for (left) 6 × 6 and (right) 10 × 10 meshes (contact collocation points represented by diamond symbols).

considered to be active, the predicted contact stresses in the remainder of the structure are underestimated. Nevertheless, when considering more refined meshes (10 × 10 and 14 × 14 elements) it can be seen that the presented methodology is able to predict the reference contact stresses with a good accuracy.

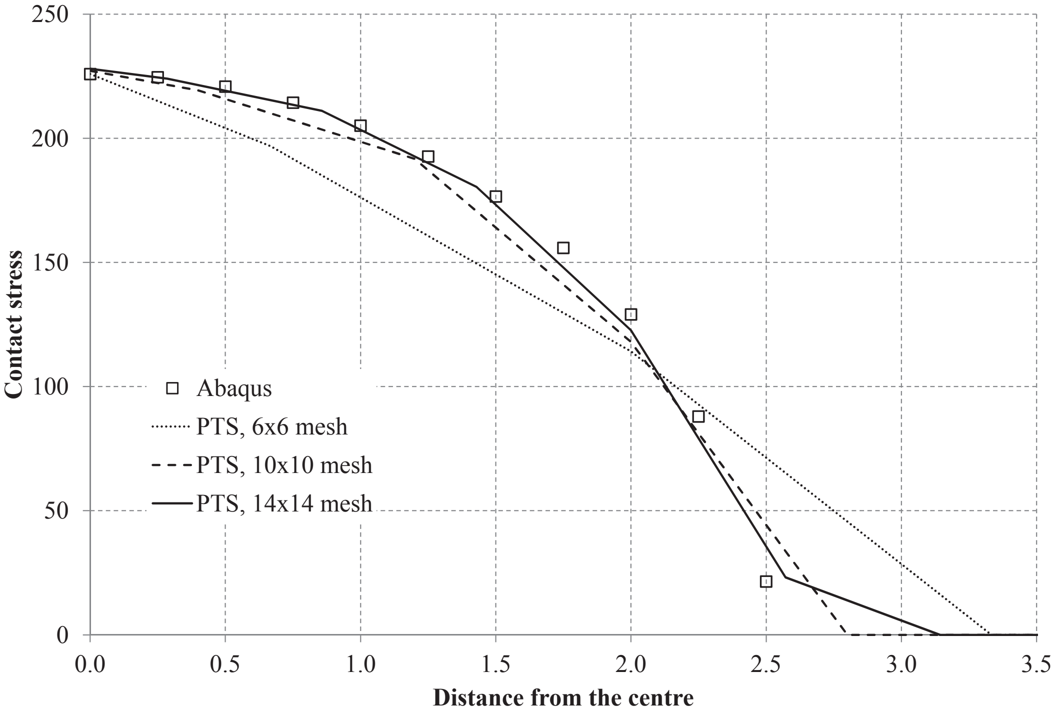


Figure 7.62: Indentation of an elastic block by a circular rigid punch: contact stress for different mesh densities.

7.4.3 Indentation of an Elastic Block by a Flat Rigid Punch

The current example deals with the indentation of an elastic foundation by a flat rigid punch, for which a schematic representation can be seen in Figure 7.63. The elastic foundation is defined by the length $L = 3.0$ and height $h = 1.0$, with the material properties given by an elastic modulus $E = 200.0 \times 10^9$ and Poisson's coefficient $\nu = 0.3$, and plane strain conditions are assumed. Due to symmetry conditions and as in the previous example, only half of the problem is modelled. The rigid flat punch, with a width $a = 0.5$, is subjected to a vertical prescribed displacement of 0.01.

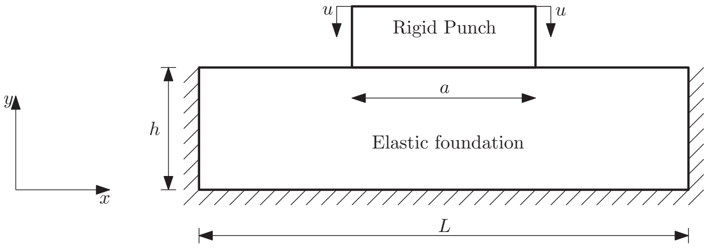


Figure 7.63: Setup of the indentation of an elastic block by a flat rigid punch.

The theoretical contact stress distribution at the punch-block interface is given by [Johnson 85]

$$\sigma^{\mathrm{C}}(x) = \frac{F}{\pi\sqrt{a^2 - x^2}}, \tag{7.6}$$

where F is the total applied force at the punch and x is the distance from the centre of contact. It can be seen from the previous equation that the contact stress will tend to infinity at the sharp corner of the punch.

In this problem, three different mesh configurations consisting of 16×16 elements are considered. In the first configuration (mesh I), uniform knot vectors are taken into account in both directions. In the other two configurations (mesh II and mesh III), the knot vectors were defined in such a way that a higher element density is present in the vicinity of the sharp corner of the rigid punch. These different mesh configurations can be seen in Figure 7.64.

The results for the contact stress distribution for the three considered configurations are plotted in Figure 7.65, along with the theoretical results and a solution coming from the commercial software Abaqus. Due to the low number of collocation points in the contact zone, the model with mesh I is not able to correctly reproduce the contact stress distribution, leading to higher contact stresses than those predicted by the theoretical model. On the other hand, the mesh with the finer refinement near the sharp corner of the punch is able to predict the contact stress distribution very accurately.

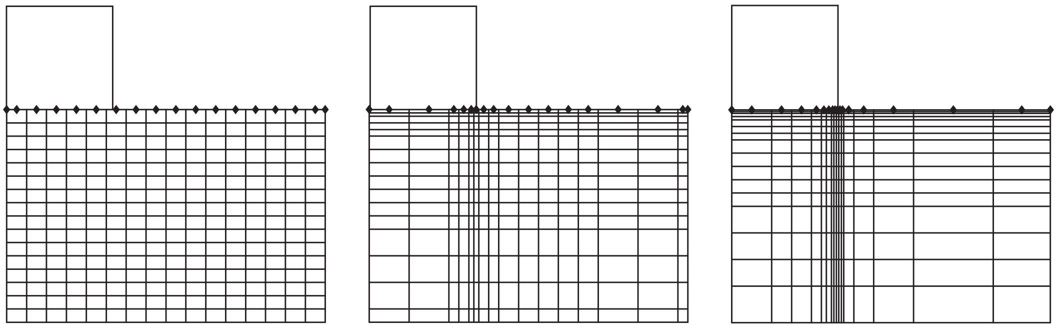


Figure 7.64: Indentation of an elastic block by a flat rigid punch: mesh configurations (from left to right) I, II and III.

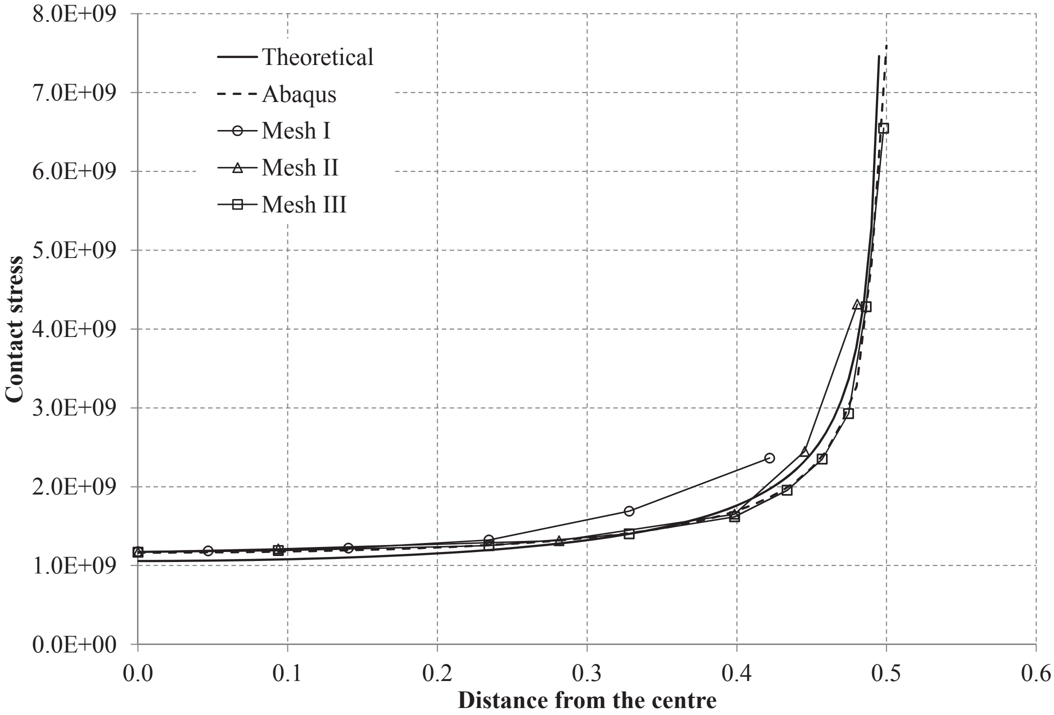


Figure 7.65: Indentation of an elastic block by a flat rigid punch problem: contact stress for the different mesh configurations.

7.4.4 Hertz Contact Problem

In this problem, an infinitely long elastic cylinder is pressed between two rigid surfaces, as can be seen in Figure 7.66. The cylinder is defined by a radius $R = 4.0$ and the constitutive parameters are given by the elastic modulus $E = 1.0 \times 10^6$ and Poisson's coefficient $\nu = 0.3$. Due to symmetry conditions, only a quarter of the cylinder is modelled, being the upper symmetry plane subjected to a total prescribed displacement of 0.15. The resulting model is then discretised using a 16×16 mesh. Different knot vectors were considered in order to obtain different numbers of collocation points in the contact zone, leading to the mesh configurations I, II and III, shown in Figure 7.67.

The analytical solution for the contact stress distribution along the contact surface can be computed as [Timoshenko 51, Kikuchi 88]

$$\sigma^{\mathrm{C}} = \frac{2F}{\pi b^2}\sqrt{b^2 - x^2}, \tag{7.7}$$

where F is the equivalent force and b is the half-width of the contact surface, defined as

$$b = 2\sqrt{\frac{FR(1-\nu^2)}{E\pi}}. \tag{7.8}$$

The solution is obtained by considering that the cylinder is affected by small displacements and small strains only.

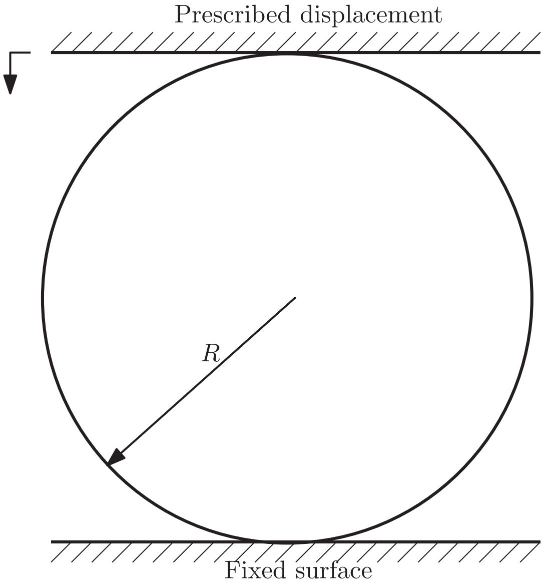


Figure 7.66: Hertz contact problem setup.

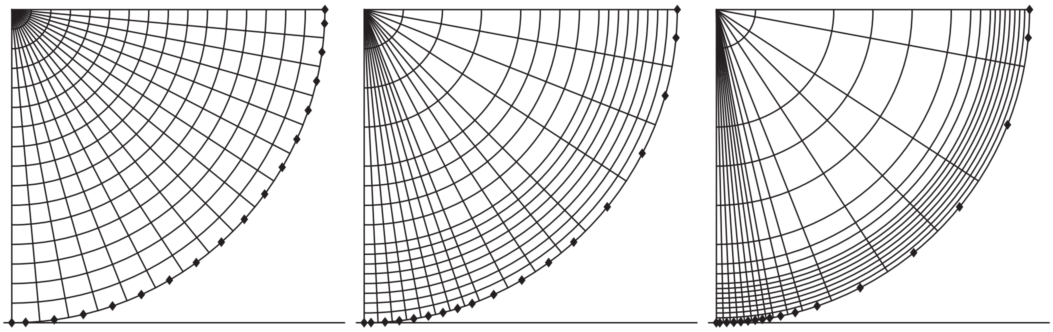


Figure 7.67: Hertz contact problem: mesh configurations (from left to right) I, II and III.

The resulting curves for the normal contact stress versus the distance from the centre, for each mesh configuration, are presented in Figure 7.68. The results are compared with the analytical solution and with those coming from a consistent segment algorithm proposed by Baig [Baig 06]. It can be seen that, when considering the mesh configuration III, the contact stress distribution follows the same tendency of the analytical solution. Similarly to the previous example, the mesh configuration III is able to reproduce the contact stresses in the slave surface more accurately due to the higher number of collocation points in the contact zone, although all three configurations are able to predict similar values for the maximum normal contact stress. These predicted maximum contact stresses are seen to be lower to the values coming from the analytical solution (about 9%), being however very similar to those obtained by the consistent segment procedure of Baig [Baig 06].

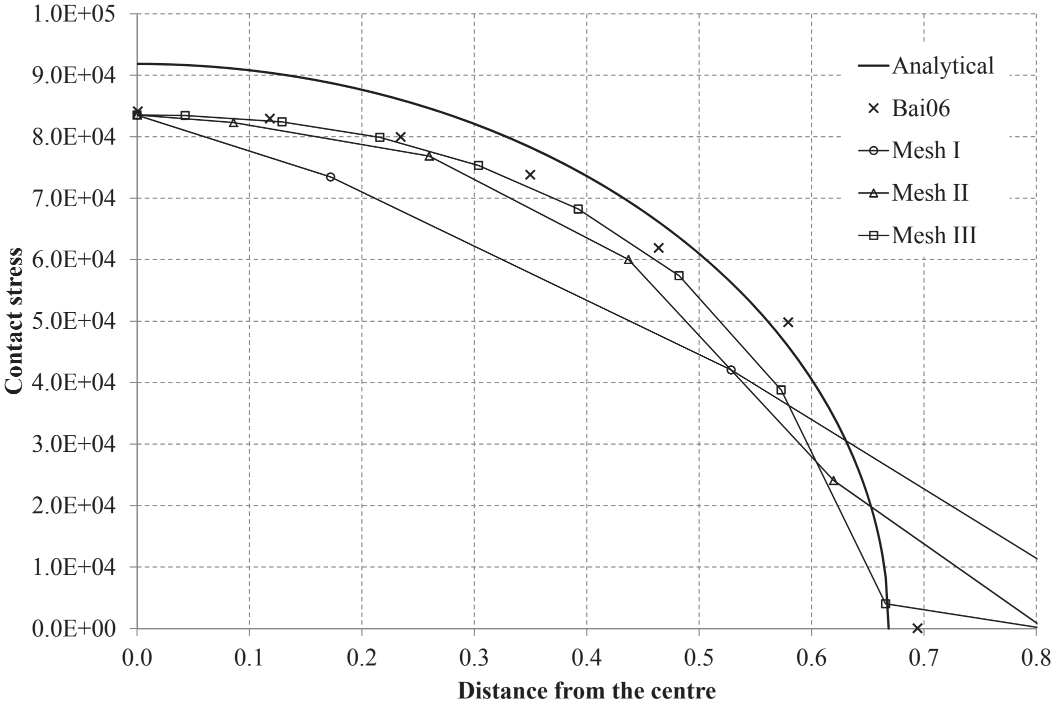


Figure 7.68: Hertz contact problem: contact stress for the different mesh configurations.

Chapter 8

Conclusions and Future Works

The main conclusions of the work are presented, along with some suggestions for future developments.

The present research work is mainly devoted to the development of robust tools for the numerical simulation of solid mechanics problems in the context of Isogeometric Analysis. With this goal, a NURBS-based solid-shell Finite Element formulation was proposed, based on the Assumed Natural Strain method. In addition, an introductory study of contact mechanics in Isogeometric Analysis, focused in the Point-to-Segment algorithm, was also performed. All these developments were implemented in a set of in-house developed tools.

In Chapter 2 the classical Finite Element Method was introduced and detailed, along with the procedures necessary to implement a Lagrangian displacement-based formulation. This chapter served to introduce the nomenclature employed, since the Finite Element Method and Isogeometric Analysis share many characteristics. In Chapter 3, the B-Spline basis functions were introduced, along with the procedure to define a curve, surface or volume using B-Splines and NURBS entities. The integration of the B-Spline basis with Finite Element Analysis is detailed and the numerical tools developed throughout the present work were presented. Chapter 4 dealt with the inclusion of geometric and material nonlinearities in the numerical models. A state-of-the-art review on the treatment of locking pathologies in the context of both the Finite Element Method and Isogeometric Analysis was given in Chapter 5. The extension of the Enhanced Assumed Strain and Assumed Natural Strain methods to Isogeometric Analysis was also studied.

From the research carried out, it was concluded that a high-order solid element based uniquely in the EAS method requires a high number of enhancing parameters to solve volumetric and shear locking pathologies. This would inevitably lead to high computational costs, since a square matrix with size equal to the number of enhancing parameters must be inverted which turns the EAS + IGA approach prohibited and not practical for complex problems. In addition, adding the enhancing variables to NURBS-based formulations, in the same way as is done with Lagrangian elements, may lead to numerical instabilities.

Afterwards, the Assumed Natural Strain method for Isogeometric Analysis was proposed. The method relies in the definition of a local space, in addition to the patch parametric space present in IGA (which is referred to as the global space). This local space was then employed to compute a set of local basis functions that interpolate the strain fields of the tying points, which replaced the ones from the standard Gauss points. The presented methodology led to a NURBS-based solid-shell formulation suitable for the analysis of thin structures, which was denoted as H2ANS. The main advantages of this methodology proved to be its simplicity and easiness of implementation, also into available finite element codes.

The methodology's performance was assessed by means of a wide range of numerical benchmark problems. The results demonstrate that the H2ANS formulation is able to alleviate locking pathologies (such as shear and membrane locking) leading to a significantly superior performance when compared with the standard quadratic solid element. In fact, in some of the presented examples, the H2ANS solid-shell element was able to obtain a performance which was very similar to the one of the standard cubic NURBS-based element. However, the proposed methodology presents significantly lower computational cost, specially when considering analyses in the nonlinear regime. In addition, since a full integration scheme consisting of $(p+1)\times(q+1)\times(r+1)$ was employed, it was not necessary to develop stabilisation techniques which are frequently used to avoid numerical instabilities in reduced integration formulations in the context of Lagrangian-based Finite Element formulations. The same methodology was employed to devise a NURBS-based quadratic element suitable for the analysis of problems in the linear elastic range which are affected by volumetric locking pathologies, which was denoted H2PV. This formulation, which can be seen as a type of local $\overline{\mathbf{B}}$ method, was able to improve the numerical solutions obtained by the standard quadratic NURBS-based solid element. It is worth mentioning that the methodology inherent to the H2ANS and H2PV formulations can be easily extended to higher-order interpolations, which is an added advantage over the EAS approach.

Chapter 6 was dedicated to the use of NURBS-based formulations in contact mechanics. The use of NURBS-based formulations for this type of problems is an attractive alternative to classical methodologies due to the high inter-element continuity and better approximation of contact stresses. In this Thesis, the Point-to-Segment algorithm (which can be seen as the Isogeometric Analysis counterpart of the Node-to-Segment algorithm widely used in Finite Element Analysis) was studied. The implementation for the analysis of linear elastic problems was validated using a set of well-known benchmark problems. This study represents the first step in the field of computational contact mechanics within the research group in which the author is inserted. Thus, it is intended to serve as the foundation on which future researchers can build upon in computational contact mechanics' field.

During the time period in which the present research work was performed, a set of numerical tools were created. These tools include an in-house developed Isogeometric COde (ICO) and several user subroutines for the commercial Finite Element software Abaqus. Focusing in the ICO, the code was written in a modular fashion, allowing other researchers to more easily implement and test new methodologies.

The author of this Thesis believes that an interesting research topic to be pursued in the future would be the extension of the Enhanced Assumed Strain methodology to Isogeometric Analysis. This is an attractive strategy to avoid different kinds of locking effects (volumetric, shear, etc.), but requires special considerations when employed in the context of IGA, since it was proved that using the same paradigm as adopted with the Finite Element Method renders a not effective approach, from the point of view of the computational costs.

Another interesting application of the proposed solid-shell ANS $+$ IGA approach would be in the numerical simulation of biomedical structures (tissues and ligaments, for instance), where 3D modelling capabilities along with locking-free formulations are welcomed.

From the point of view of the numerical integration, in this work a Gaussian integration procedure was adopted, as in the great majority of works in the literature. However, as was highlighted previously, this choice of integration points is not optimal for the IGA context and, recently, a significant amount of research effort has been dedicated to this subject. Therefore, extending the ANS $+$ IGA formulation proposed in this work to alternative sets of integration (and tying) points would be a valuable contribution.

Finally, and related to contact mechanics, the use of IGA for the numerical simulation of contact problems is particularly attractive for the analysis of, for instance, sheet and bulk metal forming operations where large sliding contact between the different components is present. In this category, the Point-to-Segment algorithm is an interesting methodology due to its simplicity. Thus, its extension to three-dimensional analysis and friction problems is also a valuable topic for future research, as well as the comparison with other methodologies (such as the mortar method).

Appendix A

Isogeometric COde User’s Manual

The Isogeometric COde (ICO) is an in-house code fully written in Fortran 90 for solving solid mechanics problems using the Isogeometric Analysis (IGA) concept. The code can be separated into two sub-codes: one for two-dimensional analysis and a second one for three-dimensional problems. In its current version, the ICO code supports:

.: single and multipatch (compatible discretisation) analysis;

.: linear isotropic elastic problems;

.: small strains plasticity with isotropic hardening;

.: geometric nonlinear analysis;

.: contact using the Point-to-Segment algorithm (only for two-dimensional analyses).

The present document is intended to provide a general description of the code and the required steps to perform an analysis of a solid mechanics problem employing the Isogeometric Analysis concept.

The general flowchart of the code can be seen in Figure A.1. In order to perform the numerical simulation, the user must create an input file containing all the data of the model (organised using specific keywords), which will then be read by the code.

A.1 The Input File

The input file contains all the necessary information to perform the numerical simulation. The file must be given a name which will then called by the code. In order to correctly perform an Isogeometric Analysis, the data must be entered in the input file by using specific keywords. After opening the file, the code will read it, searching for predetermined keywords. Once a keyword is found, the code reads all the information related to the keyword and stores it in the appropriate array.

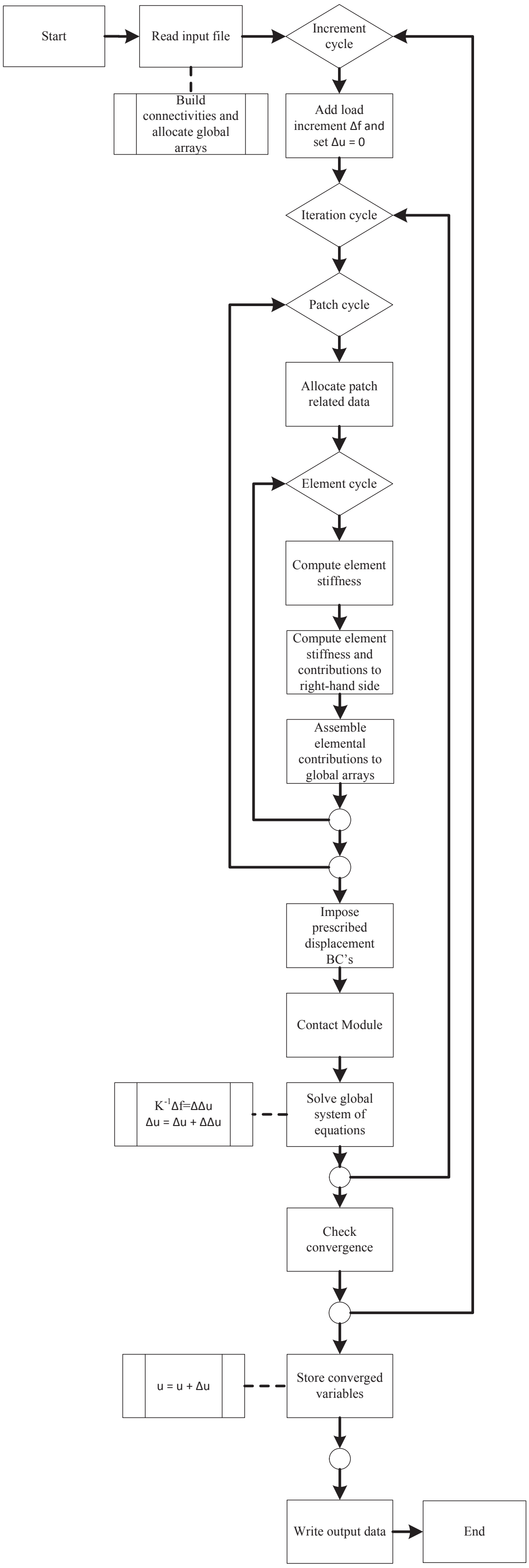

Start
Read input file
Build connectivities and allocate global arrays
Increment cycle
Add load increment Δf and set Δu = 0
Iteration cycle
Patch cycle
Allocate patch related data
Element cycle
Compute element stiffness
Compute element stiffness and contributions to right-hand side
Assemble elemental contributions to global arrays
Impose prescribed displacement BC's
Contact Module
$K^{-1}\Delta f = \Delta\Delta u$
$\Delta u = \Delta u + \Delta\Delta u$
Solve global system of equations
Check convergence
u = u + Δu
Store converged variables
Write output data
End

Figure A.1: Flowchart of the multipatch Isogeometric COde for two-dimensional analysis.

The start of an input file data is given by the keyword <`*begin`> and the end is defined by the keyword <`*end`>. Everything that is written outside of these boundaries will not be considered for the analysis. A general input file contains the following sections:

.: header, where the general dimensions of the problem are given;

.: mesh, in which the knot vectors, control points and element type are defined;

.: material properties;

.: analysis parameters, in which the number of iterations and increments are defined, as well as the type of analysis (optional);

.: boundary conditions.

In the input file the header must always be defined first in order to allocate the required variables. The general structure of the header is given as

```
*begin
nds
p, q, w
ncpx, ncpy, ncpz
closed_u, closed_v, closed_w
```

where `nds` is the number of degrees of freedom of each control point, `p`, `q` and `w` the order chosen along the ξ, η and ζ-coordinate directions, respectively. The number of control points along each direction is given by `ncpx`, `ncpy` and `ncpz`. The last line defines if the boundaries are open or closed (1: Closed, 0: Open). The keyword <`*begin`> defines the beginning of the model data input for a single patch analysis.

It is then necessary to define the control points and the elements to be used in the analysis. This section will have the following aspect

```
*knots
U
V
W
*element
X
*bnet
x1 y1 z1 w1
x2 y2 z2 w2
... ...
```

where `U`, `V` and `W` are the knot vectors along each coordinate direction. The element definition is given by the string `X`, according to the element library presented in Table A.1. Note that Gauss integration is employed in all the formulations. The keyword <`*bnet`> sets the beginning of the input of the NURBS control points, where each line represents a different control point. In each line the points are defined by the `x`, `y` and `z` coordinates, followed by the weight of the control point. The complete lattice is read using three cycles. The inner cycle for the *z*-direction, the middle cycle for the *y*-direction and the outer cycle for *x*-direction. The user must take special care in this definition in order to avoid an incorrect interpretation of the control lattice.

Table A.1: ICO Element Library

Element Tag	**Description**
Hex8	Linear hexahedral element with full integration
Hex27	Quadratic hexahedral element with full integration
Hex27ANS	Fully integrated second-order solid-shell hexahedral element enhanced with the Assumed Natural Strain method
Hex64	Cubic hexahedral element with full integration
Quad4S/E	Linear quadrilateral element for plane stress/strain with full integration
Quad9S/E	Quadratic quadrilateral element for plane stress/strain with full integration
Quad16S/E	Cubic quadrilateral element for plane stress/strain with full integration

The next step is to define the material properties associated with the patch. These properties are defined in the input file as

```
*material
iprops
props(1), props(2),...., props(iprops)
```

where `iprops` is the dimension of the array `props`. Each position of the array `props` is associated with a given material property, as shown in Table A.2. In the current version of the code, only linear elastic and isotropic hardening small strain plasticity models are implemented.

The next step consists in defining the number of increments and maximum iterations in the analysis. This section of the input file is defined as

```
*NLGeom
*Increments
```

Table A.2: ICO properties array index definition

Props index	Material Property
1	Elastic modulus
2	Poisson's Coefficient
3	Density
4	Element thickness (for plane stress analyses)
5	Yield stress
6 to -	Hardening data

```
incmax
*Iterations
itermax
```

where `incmax` and `itermax` are the total number of increments and the maximum number of iterations per increment, respectively. The keyword <`*NLGeom`> should only be used if a geometric nonlinear analysis is being considered. If this section is omitted in the input file, the parameters will take the default values of `incmax`=1 and `itermax`=25 and a geometric linear analysis will be considered.

Finally, the boundary conditions are defined. In the current version of ICO, homogeneous and inhomogeneous Dirichlet boundary conditions as well as external point loads can only be applied directly into the control points. The user must take into account that inhomogeneous Dirichlet boundary conditions and external point loads should only be applied in interpolatory control points. The imposition of these types of constraints in other positions will be addressed in future versions of the code. Additionally, the user may also apply pressure loads in the faces of the elements, as well as gravitic loads (self-weight). For the particular case of a two-dimensional analysis, in this boundary conditions section it is also possible to define the slave and master segments which will be used in the Point-to-Segment contact methodology.

If a multipatch analysis is to be conducted, the user replaces the keywords <`*begin`> by <`*begin_MP`>. In addition, the global connectivity (numbering) of the patches must be defined as

```
*MP_conn
i, conn
...
```

where `i` is the global element number and `conn` the connectivity. This second line must be repeated for each element of the model. All the other properties are defined as described above in a sequential way for each patch.

As an example, the ICO input file for the Scordelis-Lo roof problem discretised using a mesh consisting of four quadratic elements from Section 7.1.3 is given below.

```
*begin
3           !problem dimension
2, 2, 2     !degree of each dimension
4, 4, 3     !control points in each direction
0, 0, 0     !open(0) or closed(1) knot vector
---------------------------------------------------------
*knots    !knot spans
0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0
0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0
0.0, 0.0, 0.0, 1.0, 1.0, 1.0
---------------------------------------------------------
*element
Hex27
*bnet
4.3861336E+00 1.8750000E+01 2.4875000E+01 9.6984631E-01
4.4081745E+00 1.8750000E+01 2.5000000E+01 9.6984631E-01
4.4302154E+00 1.8750000E+01 2.5125000E+01 9.6984631E-01
4.3861336E+00 2.5000000E+01 2.4875000E+01 9.6984631E-01
4.4081745E+00 2.5000000E+01 2.5000000E+01 9.6984631E-01
4.4302154E+00 2.5000000E+01 2.5125000E+01 9.6984631E-01
1.2629368E+01 0.0000000E+00 2.1874708E+01 9.6984631E-01
1.2692833E+01 0.0000000E+00 2.1984631E+01 9.6984631E-01
1.2756297E+01 0.0000000E+00 2.2094554E+01 9.6984631E-01
1.2629368E+01 6.2500000E+00 2.1874708E+01 9.6984631E-01
1.2692833E+01 6.2500000E+00 2.1984631E+01 9.6984631E-01
1.2756297E+01 6.2500000E+00 2.2094554E+01 9.6984631E-01
1.2629368E+01 1.8750000E+01 2.1874708E+01 9.6984631E-01
1.2692833E+01 1.8750000E+01 2.1984631E+01 9.6984631E-01
1.2756297E+01 1.8750000E+01 2.2094554E+01 9.6984631E-01
1.2629368E+01 2.5000000E+01 2.1874708E+01 9.6984631E-01
1.2692833E+01 2.5000000E+01 2.1984631E+01 9.6984631E-01
1.2756297E+01 2.5000000E+01 2.2094554E+01 9.6984631E-01
1.5989342E+01 0.0000000E+00 1.9055356E+01 1.0000000E+00
1.6069690E+01 0.0000000E+00 1.9151111E+01 1.0000000E+00
1.6150039E+01 0.0000000E+00 1.9246867E+01 1.0000000E+00
1.5989342E+01 6.2500000E+00 1.9055356E+01 1.0000000E+00
1.6069690E+01 6.2500000E+00 1.9151111E+01 1.0000000E+00
1.6150039E+01 6.2500000E+00 1.9246867E+01 1.0000000E+00
1.5989342E+01 1.8750000E+01 1.9055356E+01 1.0000000E+00
1.6069690E+01 1.8750000E+01 1.9151111E+01 1.0000000E+00
1.6150039E+01 1.8750000E+01 1.9246867E+01 1.0000000E+00
1.5989342E+01 2.5000000E+01 1.9055356E+01 1.0000000E+00
```

```
1.6069690E+01 2.5000000E+01 1.9151111E+01 1.0000000E+00
1.6150039E+01 2.5000000E+01 1.9246867E+01 1.0000000E+00
-------------------------------------------------------
*material
5
4.32e8, 0.0, 360.0, 1.0, 1e25
-------------------------------------------------------
*bcdof
31
1, 1
5, 1
9, 1
13, 1
17, 1
21, 1
25, 1
29, 1
33, 1
37, 1
41, 1
45, 1
13, 2
14, 2
15, 2
16, 2
29, 2
30, 2
31, 2
32, 2
45, 2
46, 2
47, 2
48, 2
1, 3
2, 3
3, 3
4, 3
2, 1
3, 1
4, 1
*gravity
1.0, 3
*end
```

A.2 Keywords

In the following, the keywords available for defining the input file for ICO will be described in detail. All the keywords are case sensitive.

<*bcdof>

The keyword `<*bcdof>` is used to define homogeneous Dirichlet boundary conditions directly into the control points. This keyword uses the global numbering of the control points to define the boundary condition. The data following the `<*bcdof>` keyword is:

.: first line: total number of Dirichlet boundary conditions;

.: additional line for each boundary condition: integer defining global control point number, integer defining which degree of freedom is being restricted.

Note that a boundary condition cannot be repeated or the system of reduced equations will not be dimensioned correctly, leading to an allocation error. In alternative, the user may employ the keyword `<*boundary>` as a way to impose Dirichlet boundary conditions.

<*begin>

The keyword `<*begin>` defines the start of the data for the input file. Before using this keyword, any comments may be added to the input file. The data following the `<*begin>` keyword is:

.: first line: integer defining the dimension of the problem;

.: second line: integers defining the order of the basis used in the *xx*-, *yy*- and *zz*-directions;

.: third line: integers defining the number of control used in the *xx*-, *yy*- and *zz*-directions;

.: fourth line: integers defining if the knot vectors are open or closed (0 for open and 1 for closed).

The keyword `<*begin>` must always be used in the input file or otherwise the code will exit with and I/O error.

<*begin_MP>

The keyword `<*begin_MP>` defines the start of the data for the input file for a multipatch analysis. Before using this keyword, any comments may be added to the input file. The data following the `<*begin_MP>` keyword is:

- first line: integer defining the number of patches;
- second line: integer defining the dimension of the problem;
- one additional line for each patch containing integers defining the order of the basis used in the *xx*-, *yy*- and *zz*-directions;
- one additional line for each patch containing integers defining the number of control used in the *xx*-, *yy*- and *zz*-directions;
- one additional line for each patch containing integers defining if the knot vectors are open or closed (0 for open and 1 for closed).

The keyword `<*begin_MP>` must always be used in the input file or otherwise the code will exit with and I/O error. All the remaining keywords can be used as described herein. However all the lines in each keyword must be repeated once for each patch.

<*bnet>

The keyword `<*bnet>` is used to define the control lattice of the problem. This data is read using three DO-cycles. The outer cycle (i) for direction *xx*, the middle cycle (j) for direction *yy* and the inner cycle (k) for direction *zz*. The data is stored in matrix $\mathrm{B}_{\mathrm{net}}(i,j,k,l)$, where $l = 1,2,3,4$ contains the *xx*-coordinate, *yy*-coordinate, *zz*-coordinate and weight of the control point. The data following the `<*bnet>` keyword is:

- one line for each control point: *xx*-coordinate, *yy*-coordinate, *zz*-coordinate, weight.

The users must take special attention to the way the control lattice in read in the code in order to avoid an incorrect interpretation of the geometry.

<*boundary>

The keyword `<*boundary>` is used to define homogeneous Dirichlet boundary conditions directly into the control points. This keyword uses the indexes from the control lattice to defined the control point subjected to the boundary condition. The data following the `<*boundary>` keyword is:

.: first line: total number of Dirichlet boundary conditions;

.: additional line for each boundary condition: i, j, k, l where i, j and k defines the position in the control lattice and l defined which degree of freedom is being restricted.

Note that a boundary condition cannot be repeated or the reduced system of equations will not be dimensioned correctly, leading to an allocation error. In alternative, the user may employ the keyword <`*bcdof`> as a way to impose Dirichlet boundary conditions.

<*ContactPTS>

This keyword is used in order to activate the Point-to-Segment algorithm for dealing with contact between two bodies. The definition of the master and slave segments (using the keywords <`*Master`> and <`*Slave`>, respectively) must be performed after the <`*ContactPTS`> keyword. The Point-to-Segment algorithm is available for two-dimensional analyses only.

<*dispdof>

The keyword <`*dispdof`> is used to apply prescribed displacements directly into the control points. This keyword uses the global numbering of the control points to define the boundary condition. The data following the <`*dispdof`> keyword is:

.: first line: total number of prescribed displacement boundary conditions;

.: additional line for each prescribed displacement boundary condition: integer defining global control point number, integer defining which degree of freedom is considered, real number defining the the displacement.

Note that this keyword applies the prescribed displacement directly into the control points. Therefore, if the control point does not belong to the physical geometry of the problem, this command should not be employed. The user may also employ the keyword <`*displacement`> as a way to apply displacement boundary conditions directly into the control points.

<*displacement>

The keyword <`*displacement`> is used to apply prescribed displacements directly into the control points. This keyword uses the indexes from the control lattice to define the boundary condition. The data following the <`*displacement`> keyword is:

.: first line: total number of prescribed displacement boundary conditions;

.: additional line for each boundary condition: i, j, k, l, d where i, j and k defines the position in the control lattice and l defines which degree of freedom is being considered and d is the displacement being imposed.

Note that this keyword applies the prescribed displacement directly into the control points. Therefore, if the control point does not belong to the physical geometry of the problem, this command should not be employed. The user may also employ the keyword `<*dispdof>` as a way to apply displacement boundary conditions directly into the control points.

<*element>

The keyword `<*element>` is used to defined the element type that will be used to solve the numerical problem. The data following the `<*element>` keyword is:

.: first and only line: string defining the element (see Table A.1).

In all the formulations, standard Gaussian quadrature is employed. The term full integration of an element of order p stands for $(p+1)$ integration points used in each direction, while reduced integration consists of p integration points along each direction. The user must choose an element formulation which is consistent with the inputed data. For example, errors or inaccurate results may occur if the Hex8 element is used in a mesh containing data for a quadratic formulation.

<*end>

The keyword `<*end>` defines the end of the input file. No data written after this keyword will be read by the code. This `<*end>` keyword must always be present in the input file or otherwise the code will exit with and I/O error.

<*gravity>

The keyword `<*gravity>` is used to apply gravity loads (self-weight) to the numerical model. The data following the `<*gravity>` keyword is:

.: first and only line: real number defining the acceleration constant, integer defining the direction of the gravity (1, 2 or 3, corresponding to the xx, yy or zz axis, respectively).

Note that in order to use the keyword `<*gravity>` to define gravity loads, the density of the material must be defined and different from zero.

<*Iterations>

The keyword `<*Iterations>` defines the maximum number of iteration allowed in each increment. The data following the `<*Iterations>` keyword is:

.: first and only line: integer defining the maximum number of iterations per increment.

If the keyword `<*Iterations>` is omitted, the code will use the default value of 25 iterations. If the code is unable to converge within the defined number of iterations, the simulation will terminate and a warning message will be printed to the screen.

<*Increments>

The keyword `<*Increments>` defines the number of increments in which the analysis is divided. The data following the `<*Increments>` keyword is:

.: first and only line: integer defining the number of increments.

If the keyword `<*Increments>` is omitted, the code will use the default value of 1 increment.

<*knots>

The keyword `<*knots>` is used to input the knot vectors to be used in the analysis. The data following the `<*knots>` keyword is:

.: first line: knot vector along the *xx*-direction;

.: second line: knot vector along the *yy*-direction;

.: third line: knot vector along the *zz*-direction.

<*load>

The keyword `<*load>` is used to apply external forces directly into the control points. This keyword uses the indexes from the control lattice to define the control point subjected to the load. The data following the `<*load>` keyword is:

.: first line: total number of external forces;

.: additional line for each boundary condition: i, j, k, l, F where i, j and k defines the position in the control lattice, l defines which degree of freedom is being restricted and F is the magnitude of the load.

Note that this keyword applies the load directly into the control points. Therefore, if the control point does not belong to the physical geometry of the problem, this command should not be employed. The user may also employ the keyword <*loaddof> as a way to apply external forces directly into the control points.

<*loaddof>

The keyword <*loaddof> is used to apply external forces directly into the control points. This keyword uses the global numbering of the control points to define the boundary condition. The data following the <*loaddof> keyword is:

.: first line: total number of external forces;

.: additional line for each boundary condition: integer defining global control point number, integer defining which degree of freedom is considered, real number defining the load.

Note that this keyword applies the load directly into the control points. Therefore, if the control point does not belong to the physical geometry of the problem, this command should not be employed. The user may also employ the keyword <*load> as a way to apply external forces directly into the control points.

<*Master>

Keyword used to define the master segment for a two-dimensional contact analysis. The data following the <*Master> keyword is:

.: first line: integer defining the order of the master segment;

.: second line: number of control points which define the master segment;

.: third line: knot vector of the master segment;

.: fourth line: list of control point that define the master segment.

<*material>

The keyword <*material> is used to define the material properties to be used in the analysis. The data following the <*material> keyword is:

.: first line: integer *iprops* defining the total number of properties

.: second line: *props*(1), *props*(2),...., *props*(*iprops*), where *props*(*i*) is a real number defining property *i*, accordingly to Table A.2.

To define an hardening curve, the data must be inputed in tabular form. The position 6 of the array props must contain the yield stress and position 7 the corresponding plastic strain. Additional points of the hardening curve must be added in the same manner: the yield stress followed by the corresponding plastic strain. The higher the number of data points, the better the approximation to the hardening curve will be. The minimum number of points that must be inserted is 2. Note that in the current version of the code, only linear elastic and small strain elastoplasticity with isotropic hardening constitutive models are implemented.

<*MP_conn>

Keyword used to define the global connectivity in a multipatch analysis. The data following the <`*MP_conn`> keyword is:

.: one line for each element: integer defining the global element number followed by integers that define the control points that belong to the connectivity of the element.

<*NLGeom>

The keyword <`*NLGeom`> is used when a geometric nonlinear analysis is considered. If this keyword is omitted in the input file, a geometric linear analysis is considered. No additional information is required.

<*pressure>

The keyword <`*pressure`> is used to apply pressure loads in a surface of a three-dimensional element. The data following the <`*pressure`> keyword is:

.: first line: number of pressure loads applied in the model;

.: additional line for each pressure load: number of the element, string defining the element surface, real number defining the magnitude of the load.

The pressure load is applied in the inward direction following the normal of the face. To apply an outward pressure, the load magnitude must have a negative sign. The strings to define the surface are S1-S6, accordingly to Figure A.2.

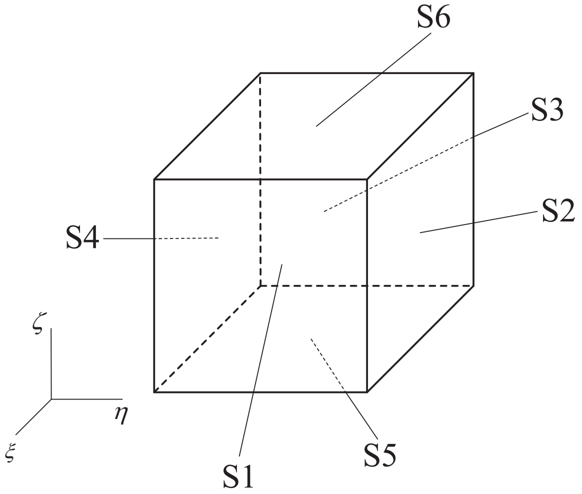


Figure A.2: Surface definition.

<*Slave>

Keyword used to define the slave segment for a two-dimensional contact analysis. The data following the `<*Slave>` keyword is:

.: first line: integer defining the order of the slave segment;

.: second line: number of control points which define the slave segment;

.: third line: knot vector of the slave segment;

.: fourth line: list of control point that define the slave segment.

Appendix B

User Element Subroutine for Abaqus

A detailed description of the implementation of NURBS-based Finite Element formulation into the commercial Finite Element code Abaqus by means of a User Element subroutine is presented in the following. To that end, a conceptual implementation of the main subroutines is given, together with the necessary input files.

At the beginning of each increment, before computing the element-related variables, it is necessary to input the NURBS-related data, which is accomplished by means of the `UEXTERNALDB` subroutine which, in turn, will store the information in the global module `ModVariables`. Afterwards, it is possible to perform the computation of the elemental stiffness and internal force vector, along with the necessary state variables (stress and strain fields, local axes, *etc.*).

B.1 NURBS Data Input File

The NURBS data input file contains all the information necessary for the computation of the NURBS-based element variables that cannot be included directly in the Abaqus input file. This file will then include the knot vector and control points weights which are required to compute the NURBS basis functions. In the current implementation, the different variables are separated by keywords, as can be seen in the following example.

```
*begin
3           !problem dimension
2, 2, 2     !degree of each dimension
4, 4, 3     !control points in each direction
-------------------------------------------------------
*knots    !knot spans
0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0
0.0, 0.0, 0.0, 0.5, 1.0, 1.0, 1.0
0.0, 0.0, 0.0, 1.0, 1.0, 1.0
-------------------------------------------------------
```

```
*bnet     !nurbs control points
0.0000000000E+00 0.0000000000E+00 2.4875000000E+01 1.0000000000E+00
0.0000000000E+00 0.0000000000E+00 2.5000000000E+01 1.0000000000E+00
0.0000000000E+00 0.0000000000E+00 2.5125000000E+01 1.0000000000E+00
0.0000000000E+00 6.2500000000E+00 2.4875000000E+01 1.0000000000E+00
...
*end
```

B.2 Subroutine UEXTERNALDB

As mentioned before, the UEXTERNALDB subroutine is responsible for reading the user-defined input file (defined in the previous section) which contains all the information necessary for the computation of NURBS basis functions. The coding presented herein is specific to the H2ANS element proposed in Section 5.5, as the **M** matrix is precomputed in this step and stored in the global module. It is also worth mentioning that the NURBS-based element connectivity is also computed in order to be used directly in the calculation of the basis functions. The subroutine can be conceptually written in Fortran programming language as follows.

```
subroutine UEXTERNALDB(LOP,LRESTART,TIME,DTIME,KSTEP,KINC)
use ModVariables
include 'ABA_PARAM.INC'
dimension TIME(2)
!Variable definition -----
character*256::FileName
character*256::Line
...
!Define path to NURBS-data input file -----
FileName = 'C:/.../SLo_2el_p2q2w2.txt'
open(unit=1,file=FileName)
!Read input file -----
do while(Line .ne. '*end')
  read(1,*)Line ...
end do
!Generate element connectivity for NURBS basis functions
call gen_ien_inn
!Compute ANS related matrices
allocate(Nb(p+1),Mb(q+1),dNbdxi(p+1),dMbdeta(q+1))
allocate(Nbr(p),Mbr(q),dNbrdxi(p),dMbrdeta(q))
!Tying Points Coordinates -----
CA = dsqrt(1.0d0/3.0d0)
CB = dsqrt(3.0d0/5.0d0)
TyPt = 0.0d0
```

```
TyPt(1,1)= CA; TyPt(1,2)= CB;
TyPt(2,1)=-CA; TyPt(2,2)= CB;
...
TyPt(16,1)=-CA; TyPt(16,2)=-CA
!Bezier knot vectors -----
ub = 1.0d0
...
do k1=1,p+1
  ub(k1) = 0.0d0
end do
...
!ANS M Matrix for Tying Point cycle 1 -----
MTP1 = 0.0d0
do k1=1,6
  uk = (TyPt(k1,1) + 1.0d0)/2.0d0
  vk = (TyPt(k1,2) + 1.0d0)/2.0d0
  Nbr = 0.0d0
  dNbrdxi = 0.0d0
  !Compute basis functions and derivatives for local space -----
  call BSplineBasisAndDeriv(2,p-1,uk,ubr,Nbr,dNbrdxi)
  Mb = 0.0d0
  dMbdeta = 0.0d0
  call BSplineBasisAndDeriv(3,q  ,vk, vb, Mb,dMbdeta)
  count = 0
  !Build ANS M matrix for tying point cycle 1 -----
  do k2=0,q
    do k3=0,p-1
      count = count + 1
      MTP1(k1,count) = Nbr(p-k3)*Mb(q+1-k2)
    end do
  end do
end do
!Inverse of the ANS M Matrix for Tying Point cycle 1 -----
temp6 = 0.0d0
call gaussj(MTP1,6,temp6,ilixo)
!ANS M Matrix for Tying Point cycle 2 -----
...
!ANS M Matrix for Tying Point cycle 3 -----
...
return
end
```

B.3 Abaqus Input File

In the Abaqus input file the user defines the control points coordinates, the type of user element and its connectivities and all the boundary conditions. In addition, the type of analysis and the output variables are also defined for each step of the simulation. A sample of an input file for Abaqus in the context of Isogeometric analysis using quadratic elements is given in the following.

```
*Heading
Abaqus-IGA sample input file
***
*** Control points coordinates
***
*Node
1, 0.0000000000000000E+000, 0.0000000000000000E+000, 2.4875000000000000E+001
2, 4.3861336451230644E+000, 0.0000000000000000E+000, 2.4875000000000004E+001
3, 1.2629368485328952E+001, 0.0000000000000000E+000, 2.1874707884098953E+001
4, 1.5989341790952663E+001, 0.0000000000000000E+000, 1.9055355522584584E+001
...
***
*** User element definition
***
*User element, Type=U1, Coordinates=3, Var=596, Nodes=27, Properties=6
1,2,3
*Element, Type=U1, Elset=uelement
1,43,42,41,39,38,37,35,34,33,27,26,25,23,22,21,19,18,17,11,10,9,7,6,5,3,2,1
2,44,43,42,40,39,38,36,35,34,28,27,26,24,23,22,20,19,18,12,11,10,8,7,6,4,3,2
3,47,46,45,43,42,41,39,38,37,31,30,29,27,26,25,23,22,21,15,14,13,11,10,9,7,6,5
...
*Uel property, Elset=uelement
4.32e8, 0.0, 360.0, 1.0, 1e15, 10.0
***
*** Auxiliary node sets
***
*nset,nset=symmx, generate
 1, 13, 4
17, 29, 4
33, 45, 4
*nset, nset=symmy, generate
1, 4, 1
17, 20, 1
33, 36, 1
*nset, nset=symmxz, generate
13, 16, 1
```

```
***
*** Step definition
***
*Step, Nlgeom=No
*Static
1.0, 1., 1e-05, 1.0
*Boundary
symmx, 1, 1
symmy, 2, 2
symmxz, 1, 1
symmxz, 3, 3
*Node print
U
*End step
```

B.4 Subroutine UEL

After reading the NURBS-related data (knot vectors, weights, *etc.*) and storing it in the global module, it is possible to compute the element related variables necessary for the numerical simulation. In the context of Isogeometric Analysis, the implementation of a formulation is very similar to the one from the classic Finite Element Method. Abaqus provides as an input all the necessary variables for the computations, such as material properties, control points displacements, type of analysis, *etc*. The NURBS data necessary to the computation of the basis functions is recovered from the global module and as detailed in Section B.2. The user must then provide as an output the elemental stiffness matrix (`amatrx`) the residual vector (`rhs`) and all the state variables (`svars`).

In the following, a conceptual implementation of the single patch `UEL` subroutine corresponding to the H2ANS NURBS-based element from Section 5.5 is presented. It should be noted that only the main element subroutine is presented, while all the utility subroutines (for determining the basis functions, for polar decomposition and others) can be coded using the methodologies presented in Chapters 3 and 4 and references therein.

```
subroutine uel(rhs,amatrx,svars,energy,ndofel,nrhs,nsvars,
1  props, nprops, coords, mcrd, nnode, u, du, v, a, jtype, time,
2  dtime, kstep, kinc, jelem, params, ndload, jdltyp, adlmag,
3  predef, npredf, lflags, mlvarx, ddlmag, mdload, pnewdt, jprops,
4  njpro, period)

use ModVariables
implicit real*8(a-h,o-z)
```

```
dimension rhs(mlvarx,*),amatrx(ndofel,ndofel),props(*),
1 svars(nsvars),energy(8),coords(mcrd,nnode),u(ndofel),
2 du(mlvarx,*),v(ndofel),a(ndofel),time(2),params(3),
3 jdltyp(mdload,*),adlmag(mdload,*),ddlmag(mdload,*),
4 predef(2,npredf,nnode),lflags(*),jprops(*)
...
!Displacement increment -----
ddisp=0.0d0
do j=1,nnode*ndof
  ddisp(j,1)=dU(j,1)
end do
!Check for geometric nonlinearity -----
if(lflags(2)==0) then
  nlgeom=.false.
  tdisp=0.0d0
elseif(lflags(2)==1) then
  nlgeom=.true.
  tdisp=ddisp
end if
!Updated control points coordinates for geometric nonlinearity -----
if(nlgeom==.true.)then
  do k1=1,nnode
    updtdisp(k1*3-2,1) = u(k1*ndof-2) - du(k1*ndof-2,1)
    updtdisp(k1*3-1,1) = u(k1*ndof-1) - du(k1*ndof-1,1)
    updtdisp(k1*3  ,1) = u(k1*ndof  ) - du(k1*ndof  ,1)
  end do
endif
!Initialize residual and stiffness matrix -----
do i=1,ndofel
  do j=1,nrhs
    rhs(i,j)=zero
  end do
  do k=1,ndofel
    amatrx(k,i)=zero
  end do
end do
!Recover state variables -----
scount=1
do k1=1,npi
  do k2=1,ntens
    stress(nel,k1,k2)=svars(scount)
    scount = scount + 1
  end do
end do
...
```

```
!Gauss points parametric coordinates and weights -----
call gauleg(npi_xi, e, we)
call gauleg(npi_eta, n, wn)
call gauleg(npi_zeta, c, wc)
!Mid-point configuration for geometric nonlinear analysis
if(nlgeom==.true.)then
  tdisp = ddisp
  mdisp = ddisp/2.0d0
else
  do i=1,6
    TGL(i,i) = 1.0d0
    TCL(i,i) = 1.0d0
  end do
end if
!Gauss point cycle (xi) -----
do i=1,npi_xi
  xi = e(i)
  !Gauss point cycle (eta) -----
  do j=1,npi_eta
    eta = n(j)
    xib  = ( xi + 1.0d0)/2.0d0
    etab = (eta + 1.0d0)/2.0d0
    !ANS N array for TP cycle 1 -----
    call BSplineBasisAndDeriv(ncpx-1,p-1,xib ,ubr,Nbr,dNbrdxi)
    call BSplineBasisAndDeriv(ncpy  ,q  ,etab, vb, Mb,dMbdeta)
    count = 0
    M1 = 0.0d0
    do k2=0,q
      do k3=0,p-1
        count = count + 1
        M1(1,count) = Nbr(p-k3)*Mb(q+1-k2)
      end do
    end do
    !ANS L array for TP cycle 1 -----
    MMult1 = matmul(M1,MTP1)
    !ANS N and L arrays for TP cycle 2 -----
    ...
    !ANS N and L arrays for TP cycle 3 -----
    ...
    !Gauss point cycle (zeta) -----
    do k=1,npi_zeta
      zeta = c(k)
      !Compute NURBS-basis functions -----
      call ShapeFunc(nel,xi,eta,zeta,R,dRdx,dRdxii,detj,jac,updtdisp)
      !Weight factor -----
```

```
gwt = we(i)*wn(j)*wc(k)*detj
!Recover/Compute Local Axis -----
call local_axis(nds,jac,rconv)
!-----------------------------
!Mid-point configuration
!-----------------------------
!Compute deformation gradient(matF) -----
call DefGrad3D(inodes,nds,dRde,dRdn,dRdc,jacinv,nodes
1              mdisp,MatF_mid)
!Compute rotation matrix (matR) using polar decomposition -----
call PolarDecomp3D(nds,matF_mid,matR_mid)
!Update local coordinate axis (rconv)
call axisupdate(rconv,matR_mid,rconv_mid)
!Compute NURBS-basis functions for mid-point configuration-----
call ShapeFunc(nel,xi,eta,zeta,R,dRdx,dRdxii,detj,jac_mid,
1              updtdisp+mdisp)
!Natural-to-local transformation matrix -----
Temp33_mid= matmul(transpose(rconv_mid),jacinv_mid)
call TransformationMat3D(Temp33_mid,TCL_mid)
!Global-to-local transformation matrix -----
Trans_mid=transpose(rconv_mid)
call TransformationMat3D(Trans_mid,TGL_mid)
!ANS strain-displacement operator -----
!BANS line 3 -----
do k1=1,(p+1)*(q+1)*(w+1)
  do k2 =1,nds
    BANS_mid(3,(k1-1)*3+k2) = dRdxii_mid(k1,3)*jac_mid(3,k2)
  end do
end do
!BANS lines 1 and 5 -----
do k1=1,6
  call ShapeFunc(nel,TyPt(k1,1),TyPt(k1,2),TyPt(k1,3),RA,dRAdx,
  1              dRAdxi,dRAdxii,detjA,jacA,dxdxiA,mdisp+updtdisp)
  do k2=1,(p+1)*(q+1)*(w+1)
    do k3 =1,nds
      BANS_mid(1,(k2-1)*3+k3) = BANS_mid(1,(k2-1)*3+k3) +
      1                     dRAdxii(k2,1)*jacA(1,k3)*MMult1(1,k1)
      BANS_mid(5,(k2-1)*3+k3) = BANS_mid(5,(k2-1)*3+k3) +
      1                    (dRAdxii(k2,1)*jacA(3,k3) +
      2                    dRAdxii(k2,3)*jacA(1,k3))*MMult1(1,k1)
    end do
  end do
end do
!BANS lines 2 and 6 -----
...
```

```
      !BANS line 4 -----
      ...
      !-----------------------------
      !End configuration
      !-----------------------------
      ...
      !Strain-displacement matrix in local frame -----
      Bloc=matmul(TCL,BNG)
      if(nlgeom == .false.) Bloc_mid = Bloc
      !Compute stress and strain fields and constitutive
      !matrix using the mid-point strain-displacement operator
      !in the local frame -----
      call MatPlastic3D(Bloc_mid,ddisp,stress(nel,cpi,:),...,matD)
      if (nlgeom == .true.) then
        !Compute nonlinear stiffness -----
        do k1=1,inodes
          dRdenc(1,1)=dRdx(k1,1)
          dRdenc(2,1)=dRdx(k1,2)
          dRdenc(3,1)=dRdx(k1,3)
          dRdxyz = matmul(trans,dRdenc)
          BNL(1,k1*3-2)=dRdxyz(1,1)
          BNL(2,k1*3-2)=dRdxyz(2,1)
          ...
        end do
        do k1=1,nds
          MSTR(k1*3-2,k1*3-2) = stress(nel,cpi,1)
          MSTR(k1*3-2,k1*3-1) = stress(nel,cpi,4)
          ...
        end do
        KNLG = KNLG + matmul(matmul(transpose(BNL),MSTR),BNL)*gwt
        !Store local axis -----
        laxis(cpi,:,:) = rconv
      end if
      !Linear Stiffness matrix -----
      amatrx = amatrx + matmul(matmul(transpose(Bloc),matD),Bloc)*gwt
      !Residual vector -----
     rhs = rhs - matmul(transpose(Bloc),stress)*gwt
    end do
  end do
end do
!Elemental Stiffness matrix -----
if(nlgeom==.true)amatrx = amatrx + KNLG
!Store state variables -----
scount=1
do k1=1,npi
```

```
    do k2=1,ntens
      svars(scount) = stress(nel,k1,k2)
      scount = scount + 1
    end do
  end do
  end subroutine
```

Bibliography

[Alves de Sousa 02] R. J. Alves de Sousa, R. M. Natal Jorge, P. M. A. Areias, R. A. F. Valente & J. M. A. César de Sá. *Low order elements for 3D analyses*. In Proceedings of the Fifth World Congress on Computational Mechanics (WCCMV), 7-12 July, Vienna, Austria, 2002.

[Alves de Sousa 03] R. J. Alves de Sousa, R. M. Natal Jorge, R. A. F. Valente & J. M. A. César de Sá. *A new volumetric and shear locking-free 3D enhanced strain element*. Engineering Computations, vol. 20, no. 7, pages 896–925, 2003.

[Alves de Sousa 05] R. J. Alves de Sousa, R. P. R. Cardoso, R. A. F. Valente, J. W. Yoon, J. J. Grácio & R. M. Natal Jorge. *A new one-point quadrature enhanced assumed strain (EAS) solid-shell element with multiple integration points along thickness: Part I – Geometrically linear applications*. International Journal for Numerical Methods in Engineering, vol. 62, no. 7, pages 952–977, 2005.

[Alves de Sousa 06a] R. J Alves de Sousa. *Development of a general purpose nonlinear solid-shell element and its application to anisotropic sheet forming simulation*. PhD thesis, Universidade de Aveiro, Portugal, 2006.

[Alves de Sousa 06b] R. J. Alves de Sousa, R. P. R. Cardoso, R. A. F. Valente, J. W. Yoon, J. J. Grácio & R. M. Natal Jorge. *A new one-point quadrature enhanced assumed strain (EAS) solid-shell element with multiple integration points along thickness: Part II – Nonlinear applications*. International Journal for Numerical Methods in Engineering, vol. 67, no. 2, pages 160–188, 2006.

[Andelfinger 93] U. Andelfinger & E. Ramm. *EAS-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalence to HR-elements*. International Journal for Numerical Methods in Engineering, vol. 36, no. 8, pages 1311–1337, 1993.

[Andrade Pires 04] F. M. Andrade Pires, E. A. de Souza Neto & J. L. de la Cuesta Padilla. *An assessment of the average nodal volume formulation for the analysis of nearly incompressible solids under finite strains*. Communications in Numerical Methods in Engineering, vol. 20, no. 7, pages 569–583, 2004.

[Areias 03] P. M. A. Areias, J. M. A. César de Sá, C. A. Conceição António & A. A. Fernandes. *Analysis of 3D problems using a new enhanced strain hexahedral element*. International Journal for Numerical Methods in Engineering, vol. 58, no. 11, pages 1637–1682, 2003.

[Armero 00] F. Armero & E. N. Dvorkin. *On finite elements for nonlinear solid mechanics*. Computers and Structures, vol. 75, no. 3, page 235, 2000.

[Arnold 84] D. N. Arnold, F. Brezzi & M. Fortin. *A stable finite element for the Stokes equations*. Calcolo, vol. 21, pages 337–344, 1984.

[Auricchio 05] F. Auricchio, L. Beirão da Veiga, C. Lovadina & A. Reali. *An analysis of some mixed-enhanced finite element for plane linear elasticity*. Computer Methods in Applied Mechanics and Engineering, vol. 194, no. 27-29, pages 2947–2968, 2005.

[Auricchio 12] F. Auricchio, F. Calabro, T. R. J Hughes, G. Sangalli & A. Reali. *A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis*. ICES REPORT 12-04, Institute for Computational Engineering and Sciences, University of Texas, 2012.

[Auricchio 13] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina & A. Reali. *Locking-free isogeometric collocation methods for spatial Timoshenko rods*. Computer Methods in Applied Mechanics and Engineering, vol. 263, pages 113–126, 2013.

[Baig 06] M. M. A. I. Baig. *A consistent segment procedure for solution of 2D contact problems with large displacements*. PhD thesis, Massachusets Institute of Technology, United States of America, 2006.

[Basar 92] Y. Basar, Y. Ding & W. B. Kratzig. *Finite-rotation shell elements via mixed formulation*. Computational Mechanics, vol. 10, no. 3-4, pages 289–306, 1992.

[Bathe 86] K. J. Bathe & E. N. Dvorkin. *A formulation of general shell elements - The use of mixed interpolation of tensorial components*. International Journal for Numerical Methods in Engineering, vol. 22, no. 3, pages 697–722, 1986.

[Bathe 96] K. J. Bathe. Finite element procedures. Prentice Hall, 1996.

[Bazilevs 10] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott & T. W. Sederberg. *Isogeometric analysis using T-splines*. Computer Methods in Applied Mechanics and Engineering, vol. 199, no. 5-8, pages 229–263, 2010.

[Beirão da Veiga 12] L. Beirão da Veiga, C. Lovadina & A. Reali. *Avoiding shear locking for the Timoshenko beam problem via isogeometric collocation methods*. Computer Methods in Applied Mechanics and Engineering, vol. 241–244, pages 38–51, 2012.

[Belytschko 85] T. Belytschko, H. Stolarski, W. K. Liu, N. Carpenter & J. S. J. Ong. *Stress projection for membrane and shear locking in shell finite elements*. Computer Methods in Applied Mechanics and Engineering, vol. 51, no. 1-3, pages 221–258, 1985.

[Belytschko 91] T. Belytschko & L. P. Bindeman. *Assumed strain stabilization of the 4-node quadrilateral with 1-point quadrature for nonlinear problems*. Computer Methods in Applied Mechanics and Engineering, vol. 88, no. 3, pages 311–340, 1991.

[Belytschko 92] T. Belytschko, B. L. Wong & H.Y. Chiang. *Advances in one-point quadrature shell elements*. Computer Methods in Applied Mechanics and Engineering, vol. 96, no. 1, pages 93–107, 1992.

[Belytschko 94] T. Belytschko & I. Leviathan. *Physical stabilization of the 4-node shell element with one point quadrature*. Computer Methods in Applied Mechanics and Engineering, vol. 113, no. 3-4, pages 321–350, 1994.

[Belytschko 00] T. Belytschko, W. K. Liu & B. Moran. Nonlinear finite elements for continua and structures. John Wiley & Sons, Ltd, West Sussex, England, 2000.

[Benson 10] D. J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T. J. R. Hughes & T. Belytschko. *A generalized finite element formulation* *for arbitrary basis functions: From isogeometric analysis to XFEM*. International Journal for Numerical Methods in Engineering, vol. 83, no. 6, pages 765–785, 2010.

[Benson 11] D. J. Benson, Y. Bazilevs, M. C. Hsu & T. J. R. Hughes. *A large deformation, rotation-free, isogeometric shell*. Computer Methods in Applied Mechanics and Engineering, vol. 200, no. 13-16, pages 1367–1378, 2011.

[Benson 13] D. J. Benson, S. Hartmann, Y. Bazilevs, M. C. Hsu & T. J. R. Hughes. *Blended isogeometric shells*. Computer Methods in Applied Mechanics and Engineering, vol. 255, pages 133–146, 2013.

[Betsch 96] P. Betsch, F. Gruttmann & E. Stein. *A 4-node finite shell element for the implementation of general hyperelastic 3D-elasticity at finite strains*. Computer Methods in Applied Mechanics and Engineering, vol. 130, no. 1-2, pages 57–79, 1996.

[Bischoff 97] M. Bischoff & E. Ramm. *Shear deformable shell elements for large strains and rotations*. International Journal for Numerical Methods in Engineering, vol. 40, no. 23, pages 4427–4449, 1997.

[Bonet 98] J. Bonet & A. J. Burton. *A simple average nodal pressure tetrahedral element for incompressible and nearly incompressible dynamic explicit applications*. Communications in Numerical Methods in Engineering, vol. 14, no. 5, pages 437–449, 1998.

[Bonet 01] J. Bonet, H. Marriott & O. Hassan. *An averaged nodal deformation gradient linear tetrahedral element for large strain explicit dynamic applications*. Communications in Numerical Methods in Engineering, vol. 17, no. 8, pages 551–561, 2001.

[Borden 11] M. J. Borden, M. A. Scott, J. A. Evans & T. J. R. Hughes. *Isogeometric finite element data structures based on Bézier extraction of NURBS*. International Journal for Numerical Methods in Engineering, vol. 87, no. 1-5, pages 15–47, 2011.

[Botella 02] O. Botella. *On a collocation B-spline method for the solution of the Navier-Stokes equations*. Computers & Fluids, vol. 31, pages 397–420, 2002.

[Bouclier 12] R. Bouclier, T. Elguedj & A. Combescure. *Locking free isogeometric formulations of curved thick beams*. Computer Methods in Applied Mechanics and Engineering, vol. 245-246, pages 144–162, 2012.

[Bouclier 13a] R. Bouclier, T. Elguedj & A. Combescure. *Efficient isogeometric NURBS-based solid-shell elements: Mixed formulation and B-Bar method*. Computer Methods in Applied Mechanics and Engineering, vol. 267, pages 86–110, 2013.

[Bouclier 13b] R. Bouclier, T. Elguedj & A. Combescure. *On the development of NURBS-based isogeometric solid shell elements: 2D problems and preliminary extension to 3D*. Computational Mechanics, vol. 52, no. 5, pages 1–28, 2013.

[Brank 95] B. T. Brank, F. B. Damjanic & D. Peric. *On implementation of a nonlinear four node shell finite element for thin multilayered elastic shells*. Computational Mechanics, vol. 16, pages 341–359, 1995.

[Brezzi 91] F. Brezzi & M. Fortin. Mixed and hybrid finite element methods. Springer-Verlag, New York, 1991.

[Bucalem 93] M. L. Bucalem & K. J. Bathe. *Higher-order MITC general shell elements*. International Journal for Numerical Methods in Engineering, vol. 36, no. 21, pages 3729–3754, 1993.

[Buechter 92] N. Buechter & E. Ramm. *Shell theory versus degeneration - a comparison in large rotation finite element analysis*. International Journal for Numerical Methods in Engineering, vol. 34, no. 1, pages 39–59, 1992.

[Cardoso 08] R. P. R. Cardoso, J. W. Yoon, M. Mahardika, S. Choudhry, R. J. Alves de Sousa & R. A. F. Valente. *Enhanced assumed strain (EAS) and assumed natural strain (ANS) methods for one-point quadrature solid-shell elements*. International Journal for Numerical Methods in Engineering, vol. 75, no. 2, pages 156–187, 2008.

[Cardoso 12] R. P. R. Cardoso & J. M. A. César de Sá. *The enhanced assumed strain method for the isogeometric analysis of nearly incompressible deformation of solids*. International Journal for Numerical Methods in Engineering, vol. 92, no. 1, pages 56–78, 2012.

[Caseiro 13] J. F. Caseiro, R. J Alves de Sousa & R. A. F. Valente. *A systematic development of {EAS} three-dimensional finite elements for the alleviation of locking phenomena*. Finite Elements in Analysis and Design, vol. 73, pages 30–41, 2013.

[Castellazzi 09] G. Castellazzi & P. Krysl. *Displacement-based finite elements with nodal integration for Reissner-Mindlin plates*. International Journal for Numerical Methods in Engineering, vol. 80, no. 2, pages 135–162, 2009.

[Castellazzi 12] G. Castellazzi & P. Krysl. *Patch-averaged assumed strain finite elements for stress analysis*. International Journal for Numerical Methods in Engineering, vol. 90, no. 13, pages 1618–1635, 2012.

[Caylak 12] I. Caylak & R. Mahnken. *Stabilization of mixed tetrahedral elements at large deformations*. International Journal for Numerical Methods in Engineering, vol. 90, no. 2, pages 218–242, 2012.

[Cervera 03] M. Cervera, M. Chiumenti, Q. Valverde & C. Agelet de Saracibar. *Mixed linear/linear simplicial elements for incompressible elasticity and plasticity*. Computer Methods in Applied Mechanics and Engineering, vol. 192, pages 5249–5263, 2003.

[Cervera 04a] M. Cervera, M. Chiumenti & C. Agelet de Saracibar. *Shear band localization via local J2 continuum damage mechanics*. Computer Methods in Applied Mechanics and Engineering, vol. 193, pages 849–880, 2004.

[Cervera 04b] M. Cervera, M. Chiumenti & C. Agelet de Saracibar. *Softening, localization and stabilization: capture of discontinuous solutions in J2 plasticity*. International Journal for Numerical and Analytical Methods in Geomechanics, vol. 28, no. 5, pages 373–393, 2004.

[Cervera 09] M. Cervera & M. Chiumenti. *Size effect and localization in J2 plasticity*. International Journal of Solids and Structures, vol. 46, no. 17, pages 3301–3312, 2009.

[Chiumenti 02] M. Chiumenti, Q. Valverde, C. Agelet de Saracibar & M. Cervera. *A stabilized formulation for incompressible elasticity using linear displacement and pressure interpolations*. Computer Methods in

Applied Mechanics and Engineering, vol. 191, no. 46, pages 5253–5264, 2002.

[Chiumenti 04] M. Chiumenti, Q. Valverde, C. Agelet de Saracibar & M. Cervera. *A stabilized formulation for incompressible plasticity using linear triangles and tetrahedra*. International Journal of Plasticity, vol. 20, pages 1487–1504, 2004.

[Cho 98] C. Cho, H. C. Park & S. W. Lee. *Stability analysis using a geometrically nonlinear assumed strain solid shell element model*. Finite Elements in Analysis and Design, vol. 29, no. 2, pages 121–135, 1998.

[Chróścielewski 92] J. Chróścielewski, J. Makowski & H. Stumpf. *Genuinely resultant shell finite elements accounting for geometric and material non-linearity*. International Journal for Numerical Methods in Engineering, vol. 35, no. 1, pages 63–94, 1992.

[Cisloiu 08] R. Cisloiu, M. Lovell & J. Wang. *A stabilized mixed formulation for finite strain deformation for low-order tetrahedral solid elements*. Finite Elements in Analysis and Design, vol. 44, no. 8, pages 472–482, 2008.

[Cottrell 06] J. A. Cottrell, A. Reali, Y. Bazilevs & T. J. R. Hughes. *Isogeometric analysis of structural vibrations*. Computer Methods in Applied Mechanics and Engineering, vol. 195, pages 5257–5296, 2006.

[Cottrell 07] J. A. Cottrell, T. J. R. Hughes & A. Reali. *Studies of refinement and continuity in isogeometric structural analysis*. Computer Methods in Applied Mechanics and Engineering, vol. 196, pages 4160–4183, 2007.

[Cottrell 09] J. A. Cottrell, T. J. R. Hughes & Y. Bazilevs. Isogeometric analysis: Toward integration of CAD and FEA. John Wiley & Sons, Ltd, 2009.

[Crisfield 81] M. A. Crisfield. *A fast incremental/iterative solution procedure that handles "snap-through"*. Computers and Structures, vol. 13, pages 55–62, 1981.

[Crisfield 00] M. A. Crisfield. *Re-visiting the contact patch test*. International Journal for Numerical Methods in Engineering, vol. 48, no. 3, pages 435–449, 2000.

[César de Sá 86] J. M. A. César de Sá & D. R. J. Owen. *The imposition of the incompressibility constraint in finite elements - a review of methods with a new insight to the locking phenomena*. In Proceedings of the III Internacional Conference on Numerical Methods for Non-Linear Problems, Swansea, UK, Pineridge Press Ltd, Swansea (1986), 1986.

[César de Sá 99] J. M. A. César de Sá & R. M. Natal Jorge. *New enhanced strain elements for incompressible problems*. International Journal for Numerical Methods in Engineering, vol. 44, no. 2, pages 229–248, 1999.

[César de Sá 02] J. M. A. César de Sá, R. M. Natal Jorge, R. A. F. Valente & P. M. A. Areias. *Development of shear locking-free shell elements using an enhanced assumed strain formulation*. International Journal for Numerical Methods in Engineering, vol. 53, no. 7, pages 1721–1750, 2002.

[De Boor 78] C. De Boor. A practical guide to splines. Springer, 1978.

[de Lorenzis 11] L. de Lorenzis, I. Temizer, P. Wriggers & G. Zavarise. *A large deformation frictional contact formulation using NURBS-based isogeometric analysis*. International Journal for Numerical Methods in Engineering, vol. 87, no. 13, pages 1278–1300, 2011.

[de Lorenzis 12] L. de Lorenzis, P. Wriggers & G. Zavarise. *A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method*. Computational Mechanics, vol. 49, no. 1, pages 1–20, 2012.

[de Souza Neto 96] E. A. de Souza Neto, D. Peric, M. Dutko & D. R. J. Owen. *Design of simple low order finite elements for large strain analysis of nearly incompressible solids*. International Journal of Solids and Structures, vol. 33, no. 20-22, pages 3277–3296, 1996.

[de Souza Neto 05] E. A. de Souza Neto, F. M. Andrade Pires & D. R. J. Owen. *F-bar-based linear triangles and tetrahedra for finite strain analysis of nearly incompressible solids: Part I – formulation and benchmarking*. International Journal for Numerical Methods in Engineering, vol. 62, no. 3, pages 353–383, 2005.

[Demko 85] S. Demko. *On the existence of interpolating projections onto spline spaces*. Journal of Approximation Theory, vol. 43, no. 2, pages 151–156, 1985.

[Dimitri 14] R. Dimitri, L. de Lorenzis, M. A. Scott, P. Wriggers, R. L. Taylor & G. Zavarise. *Isogeometric large deformation frictionless contact using T-splines*. Computer Methods in Applied Mechanics and Engineering, vol. 269, pages 394–414, 2014.

[Dittmann 14] M. Dittmann, M. Franke, I. Temizer & C. Hesch. *Isogeometric analysis and thermomechanical Mortar contact problems*. Computer Methods in Applied Mechanics and Engineering, vol. 274, pages 192–212, 2014.

[Doghri 00] I. Doghri. Mechanics of deformable solids - linear, nonlinear, analytical and computational aspects. Springer, 2000.

[Dohrmann 00] C. R. Dohrmann, M. W. Heinstein, J. Jung, S. W. Key & W. R. Witkowski. *Node-based uniform strain elements for three-node triangular and four-node tetrahedral meshes*. International Journal for Numerical Methods in Engineering, vol. 47, no. 9, pages 1549–1568, 2000.

[Dörfel 10] M. R. Dörfel, B. Jüttler & B. Simeon. *Adaptive isogeometric analysis by local h-refinement with T-splines*. Computer Methods in Applied Mechanics and Engineering, vol. 199, no. 5-8, pages 264–275, 2010.

[Dunne 06] F. Dunne & N. Petrinic. Introduction to computational plasticity. Oxford University Press, 2006.

[Dvorkin 84] E. N. Dvorkin & K. J. Bathe. *A continuum mechanics based four-node shell element for general non-linear analysis*. Engineering Computations, vol. 1, pages 77–88, 1984.

[Dvorkin 95] E. N. Dvorkin, D. Pantuso & E. A Repetto. *A formulation of the MITC4 shell element for finite strain elasto-plastic analysis*. Computer Methods in Applied Mechanics and Engineering, vol. 125, no. 1-4, pages 17–40, 1995.

[Eberlein 99] R. Eberlein & P. Wriggers. *Finite element concepts for finite elastoplastic strains and isotropic stress response in shells: theoretical and computational analysis*. Computer Methods in Applied Mechanics and Engineering, vol. 171, no. 3-4, pages 243–279, 1999.

[Echter 10] R. Echter & M. Bischoff. *Numerical efficiency, locking and unlocking of NURBS finite elements*. Computer Methods in Applied Mechanics and Engineering, vol. 199, no. 5–8, pages 374–382, 2010.

[Echter 13] R. Echter, B. Oesterle & M. Bischoff. *A hierarchic family of isogeometric shell finite elements*. Computer Methods in Applied Mechanics and Engineering, vol. 254, pages 170–180, 2013.

[El-Abbasi 00] N. El-Abbasi & S. A. Meguid. *A new shell element accounting for through-thickness deformation*. Computer Methods in Applied Mechanics and Engineering, vol. 189, no. 3, pages 841–862, 2000.

[El-Abbasi 01] N. El-Abbasi & K. J. Bathe. *Stability and patch test performance of contact discretizations and a new solution algorithm*. Computers & Structures, vol. 79, no. 16, pages 1473–1486, 2001.

[Elguedj 08] T. Elguedj, Y. Bazilevs, V. M. Calo & T. R. J. Hughes. *B-bar and F-bar projection methods for nearly incompressible linear and nonlinear elasticity and plasticity using higher-order NURBS Elements*. Computer Methods in Applied Mechanics and Engineering, vol. 197, pages 2732–2762, 2008.

[Eriksson 02] A. Eriksson & C. Pacoste. *Element formulation and numerical techniques for stability problems in shells*. Computer Methods in Applied Mechanics and Engineering, vol. 191, no. 35, pages 3775–3810, 2002.

[Frisch-Fay 62] R Frisch-Fay. Flexible bars. Butterworths, London, 1962.

[Gee 09] M. W. Gee, C. R. Dohrmann, S. W. Key & W. A. Wall. *A uniform nodal strain tetrahedron with isochoric stabilization*. International Journal for Numerical Methods in Engineering, vol. 78, no. 4, pages 429–443, 2009.

[Harnau 06] M. Harnau & K. Schweizerhof. *Artificial kinematics and simple stabilization of solid-shell elements occurring in highly constrained situations and applications in composite sheet forming simulation*. Finite Elements in Analysis and Design, vol. 42, no. 12, pages 1097–1111, 2006.

[Hauptmann 98] R. Hauptmann & K. Schweizerhof. *A systematic development of ’solid-shell’ element formulations for linear and non-linear analyses* *employing only displacement degrees of freedom*. International Journal for Numerical Methods in Engineering, vol. 42, no. 1, pages 49–69, 1998.

[Hinton 00] E. Hinton. NAFEMS introduction to nonlinear finite element analysis. NAFEMS, 2000.

[Horrigmoe 78] G. Horrigmoe & P. G. Bergan. *Nonlinear analysis of free-form shells by flat finite elements*. Computer Methods in Applied Mechanics and Engineering, vol. 16, no. 1, pages 11–35, 1978.

[Hosseini 13] S. Hosseini, J. J. C. Remmers, C. V. Verhoosel & R. de Borst. *An isogeometric solid-like shell element for nonlinear analysis*. International Journal for Numerical Methods in Engineering, vol. 95, no. 3, pages 238–256, 2013.

[Hughes 76] T. J. R. Hughes, R. L. Taylor, J. L. Sackman, A. Curnier & W. Kanoknukulchai. *A finite element method for a class of contact-impact problems*. Computer Methods in Applied Mechanics and Engineering, vol. 8, no. 3, pages 249–276, 1976.

[Hughes 77] T. J. R. Hughes. *Equivalence of finite elements for nearly incompressible elasticity*. Journal of Applied Mechanics, vol. 44, pages 1671–1685, 1977.

[Hughes 78] T. J. R Hughes, M. Cohen & M. Haroun. *Reduced and selective integration techniques in finite element analyses of plates*. Nuclear Engineering Design, vol. 46, pages 203–222, 1978.

[Hughes 80] T. J. R. Hughes. *Generalization of selective integration procedures to anisotropic and nonlinear media*. International Journal for Numerical Methods in Engineering, vol. 15, no. 9, pages 1413–1418, 1980.

[Hughes 81] T. J. R. Hughes & T.E. Tezduyar. *Finite elements based upon Mindlin plate theory with particular reference to the four-node bilinear isoparametric element*. Journal of Applied Mechanics, vol. 48, no. 3, pages 587–596, 1981.

[Hughes 84] T. J. R. Hughes. *Numerical Implementation of Constitutive Models: Rate-Independent Deviatoric Plasticity*. In Siavouche Nemat-Nasser, RobertJ. Asaro & GilbertA. Hegemier, editeurs, Theoretical foundation for large-scale computations for nonlinear material behavior, volume 6 of *Mechanics of elastic and inelastic solids 6*, pages 29–63. Springer Netherlands, 1984.

[Hughes 87] T. J. R. Hughes. The finite element method: Linear static and dynamic finite element analysis. Prentice Hall, Inc., 1987.

[Hughes 95] T. J. R Hughes. *Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods*. Computer Methods in Applied Mechanics and Engineering, vol. 127, pages 387–401, 1995.

[Hughes 05] T. J. R. Hughes, J. A. Cottrell & Y. Bazilevs. *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*. Computer Methods in Applied Mechanics and Engineering, vol. 194, pages 4135–4195, 2005.

[Hughes 10] T. J. R Hughes, A. Reali & G. Sangalli. *Efficient quadrature for NURBS-based isogeometric analysis*. Computer Methods in Applied Mechanics and Engineering, vol. 199, pages 301–313, 2010.

[Ibrahimbegović 90] A. Ibrahimbegović, R. L. Taylor & E. L. Wilson. *A robust quadrilateral membrane finite element with drilling degrees of freedom*. International Journal for Numerical Methods in Engineering, vol. 30, no. 3, pages 445–457, 1990.

[Ibrahimbegović 94] A. Ibrahimbegović & F. Frey. *Stress resultant geometrically nonlinear shell theory with drilling rotations: Part II – Computational aspects*. Computer Methods in Applied Mechanics and Engineering, vol. 118, no. 3-4, pages 285–308, 1994.

[Johnson 85] K. L. Johnson. Contact mechanics. Cambridge University Press, 1985.

[Kasper 00] E. P. Kasper & R. L. Taylor. *A mixed-enhanced strain method: Part I – Geometrically linear problems*. Computers and Structures, vol. 75, no. 3, pages 237–250, 2000.

[Key 82] S. W. Key & R. D. Krieg. *On the numerical implementation of inelastic time dependent and time independent, finite strain constitutive equations in structural mechanics*. Computer Methods in Applied Mechanics and Engineering, vol. 33, no. 1-3, pages 439–452, 1982.

[Kiendl 09] J. Kiendl, K. U. Bletzinger, J. Linhard & R. Wuchner. *Isogeometric shell analysis with Kirchhoff-Love elements*. Computer Methods in Applied Mechanics and Engineering, vol. 198, no. 49-52, pages 3902–3914, 2009.

[Kikuchi 88] N. Kikuchi & J. T. Oden. Contact problems in elasticity – a study of variational inequalities and finite element methods. SIAM, Philadelphia, 1988.

[Kim 05] K. D. Kim, G. Z. Liu & S. C. Han. *A resultant 8-node solid-shell element for geometrically nonlinear analysis*. Computational Mechanics, vol. 35, pages 315–331, 2005.

[Kim 12] J. Y. Kim & S. K. Youn. *Isogeometric contact analysis using mortar method*. International Journal for Numerical Methods in Engineering, vol. 89, no. 12, pages 1559–1581, 2012.

[Klinkel 97] S. Klinkel & W. Wagner. *A geometrical non-linear brick element based on the EAS method*. International Journal for Numerical Methods in Engineering, vol. 40, pages 4529–4545, 1997.

[Klinkel 06] S. Klinkel, F. Gruttmann & W. Wagner. *A robust non-linear solid shell element based on a mixed variational formulation*. Computer Methods in Applied Mechanics and Engineering, vol. 195, no. 1-3, pages 179–201, 2006.

[Korelc 96] J. Korelc & P. Wriggers. *An efficient 3D enhanced strain element with Taylor expansion of the shape functions*. Computational Mechanics, vol. 19, pages 30–40, 1996.

[Korelc 10] J. Korelc, A. Ursa & P. Wriggers. *An improved EAS brick element for finite deformation*. Computational Mechanics, vol. 46, pages 641–659, 2010.

[Krstulovic-Opara 02] L. Krstulovic-Opara, P. Wriggers & J. Korelc. *A C1-continuous formulation for 3D finite deformation frictional contact*. Computational Mechanics, vol. 29, no. 1, pages 27–42, 2002.

[Krysl 08] P. Krysl & B. Zhu. *Locking-free continuum displacement finite elements with nodal integration*. International Journal for Numerical Methods in Engineering, vol. 76, no. 7, pages 1020–1043, 2008.

[Krysl 12] P. Krysl & H. Kagey. *Reformulation of nodally integrated continuum elements to attain insensitivity to distortion*. International Journal for Numerical Methods in Engineering, vol. 90, no. 7, pages 805–818, 2012.

[Lee 09] M. C. Lee, S. H. Chung, S. M. Jang & M. S. Joun. *Three-dimensional simulation of forging using tetrahedral and hexahedral elements*. Finite Elements in Analysis and Design, vol. 45, no. 11, pages 745–754, 2009.

[Legay 03] A. Legay & A. Combescure. *Elastoplastic stability analysis of shells using the physically stabilized finite element SHB8PS*. International Journal for Numerical Methods in Engineering, vol. 57, no. 9, pages 1299–1322, 2003.

[Li 00] M. Li & F. Zhan. *The finite deformation theory for beam, plate and shell: Part V – The shell element with drilling degree of freedom based on Biot strain*. Computer Methods in Applied Mechanics and Engineering, vol. 189, no. 3, pages 743–759, 2000.

[Liu 94] W. K. Liu, Y. K. Hu & T. Belytschko. *Multiple quadrature underintegrated finite elements*. International Journal for Numerical Methods in Engineering, vol. 37, no. 19, pages 3263–3289, 1994.

[Liu 98] W. K. Liu, Y. Guo, S. Tang & T. Belytschko. *A multiple-quadrature eight-node hexahedral finite element for large deformation elastoplastic analysis*. Computer Methods in Applied Mechanics and Engineering, vol. 154, no. 1-2, pages 69–132, 1998.

[Lovadina 03] C. Lovadina & F. Auricchio. *On the enhanced strain technique for elasticity problems*. Computers and Structures, vol. 81, pages 777–787, 2003.

[Lu 11] J. Lu. *Isogeometric contact analysis: Geometric basis and formulation for frictionless contact*. Computer Methods in Applied Mechanics and Engineering, vol. 200, pages 726–741, 2011.

[MacNeal 82] R. H. MacNeal. *Derivation of element stiffness matrices by assumed strain distributions*. Nuclear Engineering and Design, vol. 70, no. 1, pages 3–12, 1982.

[MacNeal 94] R. H. MacNeal. Finite elements: Their design and performance. Marcel Dekker, Inc., 1994.

[Mahnken 08a] R. Mahnken & I. Caylak. *Stabilization of bi-linear mixed finite elements for tetrahedra with enhanced interpolation using volume and area bubble functions*. International Journal for Numerical Methods in Engineering, vol. 75, no. 4, pages 377–413, 2008.

[Mahnken 08b] R. Mahnken, I. Caylak & G. Laschet. *Two mixed finite element formulations with area bubble functions for tetrahedral elements*. Computer Methods in Applied Mechanics and Engineering, vol. 197, no. 9-12, pages 1147–1165, 2008.

[Masud 00a] A. Masud & C. L. Tham. *Three-Dimensional Corotational Framework for Elasto-Plastic Analysis of Multilayered Composite Shells*. AIAA Journal, vol. 38, pages 2320–2327, 2000.

[Masud 00b] A. Masud, C. L. Tham & W. K. Liu. *A stabilized 3D co-rotational formulation for geometrically nonlinear analysis of multi-layered composite shells*. Computational Mechanics, vol. 26, pages 1–12, 2000.

[Matzen 13] M. E. Matzen, T. Cichosz & M. Bischoff. *A point to segment contact formulation for isogeometric, NURBS based finite elements*. Computer Methods in Applied Mechanics and Engineering, vol. 255, pages 27–39, 2013.

[Miehe 98] C. Miehe. *A theoretical and computational model for isotropic elastoplastic stress analysis in shells at large strains*. Computer Methods in Applied Mechanics and Engineering, vol. 155, pages 193–233, 1998.

[Miehe 04] C. Miehe & N. Apel. *Anisotropic elastic-plastic analysis of shells at large strains: A comparison of multiplicative and additive approaches to enhanced finite element design and constitutive modeling*. International Journal for Numerical Methods in Engineering, vol. 61, pages 1977–2181, 2004.

[Militello 90] C. Militello & C. A. Felippa. *A variational justification of the assumed natural strain formulation of finite elements I - Variational principles*. Computers & Structures, vol. 34, no. 3, pages 431–438, 1990.

[Oñate 04] E. Oñate, J. Rojek, R. L. Taylor & O. C. Zienkiewicz. *Finite calculus formulation for incompressible solids using linear triangles* *and tetrahedra*. International Journal for Numerical Methods in Engineering, vol. 59, no. 11, pages 1473–1500, 2004.

[Pantuso 95] D. Pantuso & K. J. Bathe. *A four-node quadrilateral mixed-interpolated element for solids and fluids*. Mathematical Models and Methds in Applied Sciences, vol. 5, no. 8, pages 1113–1128, 1995.

[Pantuso 97] D. Pantuso & K. J. Bathe. *On the stability of mixed finite elements in large strain analysis of incompressible solids*. Finite Elements in Analysis and Design, vol. 28, no. 2, pages 83–104, 1997.

[Papadopoulos 92] P. Papadopoulos & R. L. Taylor. *A mixed formulation for the finite element solution of contact problems*. Computer Methods in Applied Mechanics and Engineering, vol. 94, no. 3, pages 373–389, 1992.

[Parisch 91] H. Parisch. *An investigation of a finite rotation four node assumed strain shell element*. International Journal for Numerical Methods in Engineering, vol. 31, no. 1, pages 127–150, 1991.

[Parisch 95] H. Parisch. *A continuum-based shell theory for non-linear applications*. International Journal for Numerical Methods in Engineering, vol. 38, no. 11, pages 1855–1883, 1995.

[Piegl 97] L. Piegl & W. Tiller. The NURBS Book. Springer, 1997.

[Pietrzak 99] G. Pietrzak & A. Curnier. *Large deformation frictional contact mechanics: continuum formulation and augmented Lagrangian treatment*. Computer Methods in Applied Mechanics and Engineering, vol. 177, pages 351–381, 1999.

[Piltner 00] R. Piltner. *An implementation of mixed enhanced finite elements with strains assumed in Cartesian and natural element coordinates using sparse B-matrices*. Engineering Computations, vol. 17, pages 933–949, 2000.

[Puso 00] M. A. Puso. *A highly efficient enhanced assumed strain physically stabilized hexahedral element*. International Journal for Numerical Methods in Engineering, vol. 49, no. 8, pages 1029–1064, 2000.

[Reddy 95] B. D. Reddy & J. C. Simo. *Stability and convergence of a class of enhanced strain methods*. SIAM Journal on Numerical Analysis, vol. 32, no. 6, pages 1705–1728, 1995.

[Reese 00] S. Reese, P. Wriggers & B. D. Reddy. *A new locking-free brick element technique for large deformation problems in elasticity*. Computers and Structures, vol. 75, no. 3, pages 291–304, 2000.

[Reese 02] S. Reese. *On the equivalence of mixed element formulations and the concept of reduced integration in large deformation problems*. International Journal of Nonlinear Sciences and Numerical Simulation, vol. 3, no. 1, pages 1–33, 2002.

[Reese 07] S. Reese. *A large deformation solid-shell concept based on reduced integration with hourglass stabilization*. International Journal for Numerical Methods in Engineering, vol. 69, no. 8, pages 1671–1716, 2007.

[Roehl 96] D. Roehl & E. Ramm. *Large elasto-plastic finite element analysis of solids and shells with the enhanced assumed strain concept*. International Journal of Solids and Structures, vol. 33, no. 20-22, pages 3215–3237, 1996.

[Rogers 01] D. F. Rogers. An introduction to NURBS with historical prespective. Academic Press, 2001.

[Sansour 92] C. Sansour & H. Bufler. *An exact finite rotation shell theory, its mixed variational formulation and its finite element implementation*. International Journal for Numerical Methods in Engineering, vol. 34, no. 1, pages 73–115, 1992.

[Sansour 98] C. Sansour & J. Bocko. *On hybrid stress, hybrid strain and enhanced strain finite element formulations for a geometrically exact shell theory with drilling degrees of freedom*. International Journal for Numerical Methods in Engineering, vol. 43, no. 1, pages 175–192, 1998.

[Sansour 00] C. Sansour & F. G. Kollmann. *Families of 4-node and 9-node finite elements for a finite deformation shell theory: An assesment of hybrid stress, hybrid strain and enhanced strain elements*. Computational Mechanics, vol. 24, pages 435–447, 2000.

[Schillinger 14] D. Schillinger, S. J. Hossain & T. J. R Hughes. *Reduced Bezier element quadrature rules for quadratic and cubic Splines in Isogeometric Analysis*. ICES REPORT 14-02, Institute for Computational Engineering and Sciences, University of Texas, 2014.

[Schoenberg 46] I. J. Schoenberg. *Contributions to the problem of approximation of equidistant data by analytic functions*. Quarterly of Applied Mathematics, vol. 4, pages 45–99, 112–141, 1946.

[Schwarze 09] M. Schwarze & S. Reese. *A reduced integration solid-shell finite element based on the EAS and the ANS concepts: Geometrically linear problems*. International Journal for Numerical Methods in Engineering, vol. 80, no. 10, pages 1322–1355, 2009.

[Schwarze 11] M. Schwarze & S. Reese. *A reduced integration solid-shell finite element based on the EAS and the ANS concepts: Large deformation problems*. International Journal for Numerical Methods in Engineering, vol. 85, no. 3, pages 289–329, 2011.

[Scordelis 69] A. C. Scordelis & K. S. Lo. *Computer analysis of cylindrical shells*. Journal of American Concrete Institute, vol. 61, pages 539–561, 1969.

[Scott 11] M. A. Scott, M. J. Borden, C. V. Verhoosel, T .W. Sederberg & T. J. R. Hughes. *Isogeometric finite element data structures based on Bézier extraction of T-splines*. International Journal for Numerical Methods in Engineering, vol. 88, no. 2, pages 126–156, 2011.

[Sederberg 03] T. W. Sederberg, J. Zheng, A. Bakenov & A. Nasri. *T-splines and T-NURCCs*. ACM Transactions on Graphics, vol. 22, no. 3, pages 477–484, 2003.

[Simo 85] J. C. Simo, R. L. Taylor & K. S. Pister. *Variational and projection methods for the volume constraint in finite deformation elasto-plasticity*. Computer Methods in Applied Mechanics and Engineering, vol. 51, no. 1-3, pages 177–208, 1985.

[Simo 88] J. C. Simo, D. D. Fox & M. S. Rifai. *Formulation and computational aspects of a stress resultant geometrically exact shell model*. In Computational Mechanics, pages 751–759. Springer Berlin Heidelberg, 1988.

[Simo 90a] J. C. Simo, D. D. Fox & M. S. Rifai. *On a stress resultant geometrically exact shell model: Part III – Computational aspects of the nonlinear theory*. Computer Methods in Applied Mechanics and Engineering, vol. 79, no. 1, pages 21–70, 1990.

[Simo 90b] J. C. Simo & M. S. Rifai. *A class of mixed assumed strain methods and the method of incompatible modes*. International Journal for Numerical Methods in Engineering, vol. 29, no. 8, pages 1595–1638, 1990.

[Simo 90c] J. C. Simo, M. S. Rifai & D. D. Fox. *On a stress resultant geometrically exact shell model: Part IV – Variable thickness shells with through-the-thickness stretching*. Computer Methods in Applied Mechanics and Engineering, vol. 81, no. 1, pages 91–126, 1990.

[Simo 92] J. C. Simo & F. Armero. *Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes*. International Journal for Numerical Methods in Engineering, vol. 33, no. 7, pages 1413–1449, 1992.

[Simo 98] J. C. Simo & T. J. R. Hughes. Computational inelasticity. Springer, 1998.

[Stadler 03] M. Stadler, G. A. Holzapfel & J. Korelc. *Cn continuous modelling of smooth contact surfaces using NURBS and application to 2D problems*. International Journal for Numerical Methods in Engineering, vol. 57, no. 15, pages 2177–2203, 2003.

[Sussman 87] T. Sussman & K. J. Bathe. *A finite element formulation for nonlinear incompressible elastic and inelastic analysis*. Computers and Structures, vol. 26, pages 357–409, 1987.

[Sze 00] K. Y. Sze & L. Q. Yao. *A hybrid stress ANS solid-shell element and its generalization for smart structure modelling: Part I – Solid-shell element formulation*. International Journal for Numerical Methods in Engineering, vol. 48, no. 4, pages 545–564, 2000.

[Sze 02] K. Y. Sze, W. K. Chan & T. H. H. Pian. *An eight-node hybrid-stress solid-shell element for geometric non-linear analysis of elastic shells*. International Journal for Numerical Methods in Engineering, vol. 55, no. 7, pages 853–878, 2002.

[Sze 04] K. Y. Sze, X. H. Liu & S. H. Lo. *Popular benchmark problems for geometric nonlinear analysis of shells*. Finite Elem. Anal. Des., vol. 40, pages 1551–1569, July 2004.

[Taylor 91] R. L. Taylor & P. Papodopoulos. On a patch test for contact problems in two dimensions. Computational Methods in Nonlinear Mechanics, Wriggers P, Wanger W (eds). Springer: Berlin, 1991.

[Taylor 00] R. L. Taylor. *A mixed-enhanced formulation tetrahedral finite elements*. International Journal for Numerical Methods in Engineering, vol. 47, no. 1-3, pages 205–227, 2000.

[Taylor 11] R. L. Taylor. *Isogeometric analysis of nearly incompressible solids*. International Journal for Numerical Methods in Engineering, vol. 87, no. 1-5, pages 273–288, 2011.

[Taylor 13] R. L. Taylor. Feap - finite element analysis program. University of California, Berkeley, 2013.

[Temizer 11] I. Temizer, P. Wriggers & T. J. R. Hughes. *Contact treatment in isogeometric analysis with NURBS*. Computer Methods in Applied Mechanics and Engineering, vol. 200, pages 1100–1112, 2011.

[Temizer 12] I. Temizer, P. Wriggers & T. J. R. Hughes. *Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS*. Computer Methods in Applied Mechanics and Engineering, vol. 209-212, pages 115–128, 2012.

[Thai 12] C. H. Thai, H. Nguyen-Xuan, N. Nguyen-Thanh, T. H. Le, T. Nguyen-Thoi & T. Rabczuk. *Static, free vibration, and buckling analysis of laminated composite Reissner-Mindlin plates using NURBS-based isogeometric approach*. International Journal for Numerical Methods in Engineering, vol. 91, no. 6, pages 571–603, 2012.

[Timoshenko 51] S. Timoshenko & J. N. Goodier. Theory of elasticity. MacGraw Hill Inc., 1951.

[Valente 03] R. A. F. Valente, R. M. Natal Jorge, R. P. R. Cardoso, J. M. A. César de Sá & J. J. A. Grácio. *On the use of an enhanced transverse shear strain shell element for problems involving large rotations*. Computational Mechanics, vol. 30, pages 286–296, 2003.

[Valente 04a] R. A. F. Valente. *Developments on shell and solid-shell finite elements technology in nonlinear continuum mechanics*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, Portugal, 2004.

[Valente 04b] R. A. F. Valente, R. J. Alves De Sousa & R. M. Natal Jorge. *An enhanced strain 3D element for large deformation elastoplastic thin-shell applications*. Computational Mechanics, vol. 34, pages 38–52, 2004.

[Vu-Quoc 03a] L. Vu-Quoc & X. G. Tan. *Optimal solid shells for non-linear analyses of multilayer composites I – Statics*. Computer Methods in Applied Mechanics and Engineering, vol. 192, no. 9-10, pages 975–1016, 2003.

[Vu-Quoc 03b] L. Vu-Quoc & X. G. Tan. *Optimal solid shells for non-linear analyses of multilayer composites II – Dynamics*. Computer Methods in Applied Mechanics and Engineering, vol. 192, no. 9-10, pages 1017–1059, 2003.

[Wriggers 85] P. Wriggers & J. C. Simo. *A note on tangent stiffness for fully nonlinear contact problems*. Communications in Applied Numerical Methods, vol. 1, no. 5, pages 199–203, 1985.

[Wriggers 93] P. Wriggers & F. Gruttmann. *Thin shells with finite rotations formulated in biot stresses: Theory and finite element formulation*. International Journal for Numerical Methods in Engineering, vol. 36, no. 12, pages 2049–2071, 1993.

[Wriggers 96] P. Wriggers, R. Eberlein & S. Reese. *A comparison of three-dimensional continuum and shell elements for finite plasticity*. International Journal of Solids and Structures, vol. 33, no. 20-22, pages 3309–3326, 1996.

[Wriggers 01] P. Wriggers, L. Krstulovic-Opara & J. Korelc. *Smooth C1-interpolations for two-dimensional frictional contact problems*. International Journal for Numerical Methods in Engineering, vol. 51, no. 12, pages 1469–1495, 2001.

[Wriggers 02] P. Wriggers. Computational contact mechanics. John Wiley & Sons, Ltd, 2002.

[Yastrebov 13] V. A. Yastrebov. Numerical methods in contact mechanics. John Wiley & Sons, Inc., 2013.

[Yoon 99a] J. W. Yoon, D. Y. Yang & K. Chung. *Elasto-plastic finite element method based on incremental deformation theory and continuum based* *shell elements for planar anisotropic sheet materials*. Computer Methods in Applied Mechanics and Engineering, vol. 174, pages 23–56, 1999.

[Yoon 99b] J. W. Yoon, D. Y. Yang, K. Chung & F. Barlat. *A general elasto-plastic finite element formulation based on incremental deformation theory for planar anisotropy and its application to sheet metal forming*. International Journal of Plasticity, vol. 15, no. 1, pages 35–67, 1999.

[Zavarise 09a] G. Zavarise & L. de Lorenzis. *A modified node-to-segment algorithm passing the contact patch test*. International Journal for Numerical Methods in Engineering, vol. 79, no. 4, pages 379–416, 2009.

[Zavarise 09b] G. Zavarise & L. de Lorenzis. *The node-to-segment algorithm for 2D frictionless contact: Classical formulation and special cases*. Computer Methods in Applied Mechanics and Engineering, vol. 198, pages 3428–3451, 2009.

[Zienkiewicz 71] O. C. Zienkiewicz, R. L. Taylor & J. M. Too. *Reduced integration technique in general analysis of plates and shells*. International Journal for Numerical Methods in Engineering, vol. 3, pages 275–290, 1971.

[Zienkiewicz 00] O. C. Zienkiewicz & R. L. Taylor. The finite element method: Volume 1, the basis. McGraw-Hill, 2000.

[Zienkiewicz 05] O. C. Zienkiewicz, R. L. Taylor & J. Z. Zhu. The finite element method: Its basis and fundamentals. Butterworth-Heinemann, 2005.