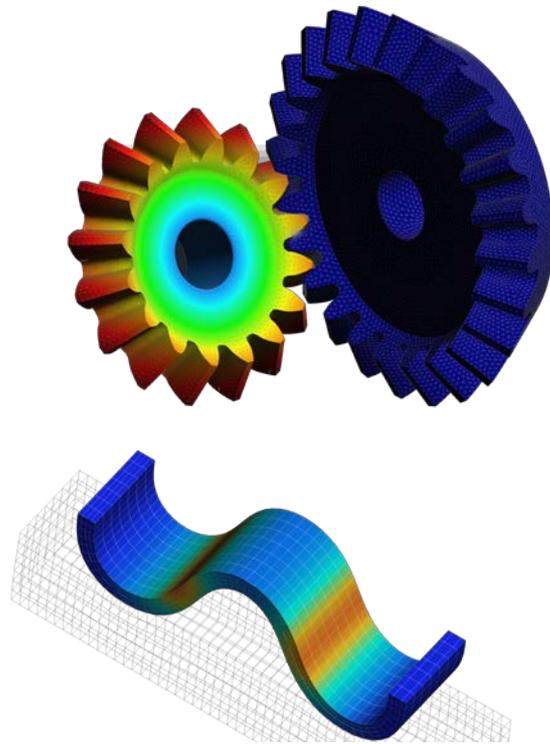


Innovative mathematical and numerical models for studying the deformation of shells during industrial forming processes with the Finite Element Method

Vicente Mataix
Eugenio Oñate
Riccardo Rossi



Innovative mathematical and numerical models for studying the deformation of shells during industrial forming processes with the Finite Element Method

Vicente Mataix
Eugenio Oñate
Riccardo Rossi

Monograph CIMNE N^o-190, October 2020

INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING
Edificio C1, Campus Norte UPC
Gran Capitán s/n
08034 Barcelona, Spain
www.cimne.com

First edition: October 2020

INNOVATIVE MATHEMATICAL AND NUMERICAL MODELS FOR STUDYING THE DEFORMATION OF SHELLS DURING INDUSTRIAL FORMING PROCESSES WITH THE FINITE ELEMENT METHOD
Monograph CIMNE M190
© The authors

“ *Ce qui est simple est toujours faux. Ce qui ne l'est pas est inutilisable* ”

Paul Valéry (1871 - 1945 AD, French poet, essayist, and philosopher)

“Essentially, all models are wrong, but some are useful

Aphorism, attributed to George E. P. Box (1919–2013 AD, British statistician)

Abstract

This document contains the result resulting from the work in the doctoral thesis *Innovative mathematical and numerical models for studying the deformation of shells during industrial forming processes with the Finite Element Method*. The objective of this thesis is to contribute to the development of finite element methods for the analysis of the stamping processes, an area of problems with a very clear industrial application. Indeed these kinds of problems involve multiple disciplines and require the understanding of different mechanical problems, being the most relevant disciplines the continuous mechanics, the plasticity, contact problems, among others, depending of the problematic of study.

To achieve the proposed goals, attention is focused in the first section of this thesis in the solid-shell elements, which are an attractive kind of element for the simulation of forming processes. This is due to the fact that any kind of generic 3D constitutive law can be employed without any kind of additional hypothesis, besides the thermomechanic problem is formulated without additional assumptions. Additionally this type of element allows the three-dimensional description of the deformable body, thus contact on both sides of the element can be treated easily.

This work will present in first place the development of a triangular prism element as a solid-shell, for the analysis of thin/thick shell, undergoing large deformations. The element is formulated in **Total Lagrangian (TL)** formulation, and employs the *neighbour (adjacent) elements* to perform a local patch to enrich the displacement field. In the original formulation a modified right *CauchyGreen* deformation tensor ($\bar{\mathbf{C}}$) is obtained; in the present work, a modified deformation gradient ($\bar{\mathbf{F}}$) is obtained, which allows to generalise the methodology and thanks to this we are able to employ the *push-forward* and *pullback* concepts. The *push-forward* and *pullback* technique provide a mathematically consistent method for define the time derivatives of the tensors; and for example it can be employed to work with elasto-plasticity.

The element is based in three modifications: (a) a classical assumed strain approach for transverse shear strains (b) an assumed strain approach for the in-plane components using information from neighbour elements and (c) an averaging of the volumetric strain over the element. The objective is to use this type of element for the simulation of shells avoiding transverse shear locking, improving the membrane behaviour of the in-plane triangle and to handle quasi-incompressible materials or materials with isochoric plastic flow. The element considers just one *Gauss* point in the plane and a chosen number of integration points along the axis, thanks to this it is possible to consider problems with a significant non-linearity related to plasticity.

This work will continue presenting the contact formulation developed, which consists in state-of-art of the numerical contact mechanics formulation for implicit simulations. Consisting in an exact mortar integration of the contact interface, this allows to obtain the most consistent integration possible between the integration domains, as well as the most exact solution possible. The implementation also considers several optimisation algorithms, like the penalty constraint optimisation, but particularly remarkable the consideration of a **Augmented Lagrangian Method (ALM)** with dual *Lagrange multipliers*, a new contribution of this work. The latter allows to static condensate the system of equations, allowing to remove the **Lagrange Multiplier (LM)** of the resolution and therefore permitting the consideration of iterative solvers. Additionally, the formulation has been properly linearised, ensuring the quadratic convergence of the problem. In order to solve the system of equations, a semi-smooth *Newton* method is considered, consisting in an active set strategy, extensible also in the case of frictional problems. The formulation works both for frictionless and frictional problems, the later essential for the simulation of forming processes. This frictional formulation is framed in the traditional friction models, like the *Coulomb* friction, but the development presented can be extended to any type of frictional model. This contact formulation is fully compatible with the solid-shell introduced on this work.

The missing ingredient necessary in order to successfully perform the industrial processes would be the constitutive models. This will be materialised in the plasticity formulation considered in this work. In these models, we will be able to consider large strain deformation plasticity models, with arbitrary combination of yield surfaces and plastic potentials, the so-called non-associative models. In order to compute the corresponding tangent tensor for these general laws, numerical implementations based on perturbation methods.

The final theoretical work of this thesis will consist in the developments of adaptive remeshing techniques. In here, different approaches will be presented. Starting with the metric based techniques, including the *level-set* and *Hessian* approaches. These techniques are of general purpose, and can be considered both in application of structural or **Computational Fluid Dynamics (CFD)** problems. In addition to this approach, the **Super convergent Patch Recovery (SPR)** error estimation method is presented. This approach is more conventional than the former ones. In both cases, in the *Hessian* and the **SPR** error estimation, have been extended in order to apply it to contact mechanics problems, main contribution of this work in this field.

With all the former developments, we will be ready for the introduction of practical cases focused in the context stamping processes. The most important point to highlight on this is that these examples will be compared with the reference solutions available in the literature as a validation of the developments presented until this point.

The present document is organised as follows. The first chapter introduces the thesis, stating the objectives and reviews the state-of-the-art on the most transcendental subjects. The second chapter does a shallow introduction of the continuous mechanics and **Finite Element Method (FEM)** concepts, required for following chapters. The third chapter derives the formulation of the solid-shell previously mentioned and the free-rotation shell, which is the theoretical inspiration of the solid-shell presented, including several academic examples that are commonly employed in the literature as a benchmark of shell elements. The fourth chapter will present the contact mechanics formulation developed, consisting in an implicit mortar **ALM** formulation, the chapter also includes several examples commonly found in the literature, which are usually considered for validation. The next chapter presents the plasticity formulation used, including some technical details in the implementation side, some examples of validation will be also introduced at the end of this chapter. The final theoretical chapter shows the adaptive remeshing algorithms developed in the context of this work, and presents several examples, including not only solid mechanics cases, but also **CFD**. The next chapter encapsules some validation and application cases for stamping processes. The final chapter comprises the conclusions, as well as the future works coming after the presented developments.

Keywords: metal-forming, stamping, shells, solid-shells, contact, plasticity, adaptive remeshing

Resumen

La tesis doctoral *Modelos matemáticos y numéricos innovadores para el estudio de la deformación de láminas durante los procesos de conformado industrial por el Método de los Elementos Finitos*. pretende contribuir al desarrollo de métodos de elementos finitos para el análisis de procesos de estampado, un área problemática con una clara aplicación industrial. De hecho, este tipo de problemas multidisciplinares requieren el conocimiento de múltiples disciplinas, como la mecánica de medios continuos, la plasticidad, la termodinámica y los problemas de contacto, entre otros.

Para alcanzar los objetivos propuestos, la primera parte de esta tesis abarca los elementos de sólido-lámina. Este tipo de elemento resulta atractivo para la simulación de procesos de conformado, dado que cualquier tipo de ley constitutiva tridimensional puede ser formulada sin necesidad de considerar ninguna conjetura adicional. Además, este tipo de elementos permite realizar una descripción tridimensional del cuerpo deformable, por tanto, el contacto de ambas caras puede ser tratado fácilmente.

Este trabajo presenta en primer lugar el desarrollo de un elemento de sólido-lámina prismático triangular, para el análisis de láminas gruesas y delgadas con capacidad para grandes deformaciones. Este elemento figura en formulación *Lagrangiana total*, y emplea los *elementos vecinos* para poder computar un campo de desplazamientos cuadráticos. En la formulación original, se obtenía un tensor de *Cauchy* derecho modificado ($\bar{\mathbf{C}}$); sin embargo, en este trabajo, la formulación se extiende obteniendo un gradiente de deformación modificado ($\bar{\mathbf{F}}$), que permite emplear los conceptos de *push-forward* y *pull-back*. Dichos conceptos proveen de un método matemáticamente consistente para la definición de derivadas temporales de tensores y, por tanto, puede ser usado, por ejemplo, para trabajar con elasto-plasticidad.

El elemento se basa en tres modificaciones: (a) una aproximación clásica de deformaciones transversales de corte mixtas impuestas; (b) una aproximación de deformaciones impuestas para las componentes en el plano tangente de la lámina; y (c) una aproximación de deformaciones impuestas mejoradas en la dirección normal a través del espesor, mediante la consideración de un grado de libertad adicional. Los objetivos son poder utilizar el elemento para la simulación de láminas sin bloquear por cortante, mejorar el comportamiento membranar del elemento en el plano tangente, eliminar el bloqueo por efecto *Poisson* y poder tratar materiales elasto-plásticos con un flujo plástico incompresible, así como materiales elásticos cuasi-incompresibles o materiales con flujo plástico isocórico. El elemento considera un único punto de *Gauss* en el plano, mientras que permite considerar un número cualquiera de puntos de integración en su eje, con el objetivo de poder considerar problemas con una significativa no linealidad en cuanto a plasticidad.

Este trabajo continúa con el desarrollo de la formulación de contacto empleada, una metodología que se encuentra en la bibliografía sobre la mecánica de contacto computacional para simulaciones implícitas. Dicha formulación consiste en una integración exacta de la interfaz de contacto mediante métodos de mortero, lo que permite obtener la integración más consistente posible entre los dominios de integración, así como la solución más exacta posible. La implementación también considera varios algoritmos de optimización, como la optimización mediante penalización. La contribución más notable de este trabajo es la consideración de *multiplicadores de Lagrange* aumentados duales como método de optimización. Estos permiten condensar estáticamente el sistema de ecuaciones, lo que permite eliminar los *multiplicadores de Lagrange* de la resolución y, por lo tanto, permite la consideración de *solvers* iterativos. Además, la formulación ha sido adecuadamente linealizada, asegurando la convergencia cuadrática del problema. Para resolver el sistema de ecuaciones, se considera un método de *Newton semi-smooth*, que consiste en una estrategia de *set* activo, extensible también en el caso de problemas friccionales. La formulación es funcional tanto para problemas sin fricción como para problemas friccionales, lo que es esencial para la simulación de procesos de estampado. Esta formulación friccional se enmarca en los modelos de fricción tradicionales, como la fricción de *Coulomb*, pero el desarrollo presentado puede extenderse a cualquier tipo de modelo de fricción. Esta formulación de contacto es totalmente compatible con el elemento sólido-lámina introducido en este trabajo.

El componente necesario restante para la simulación de procesos industriales son los modelos constitutivos. En este trabajo, esto se ve materializado en la formulación de plasticidad considerada. Estos modelos constitutivos se considerarán modelos de plasticidad para grandes deformaciones, con una combinación arbitraria de superficies de fluencia y potenciales plásticos: los llamados modelos no asociados. Para calcular el tensor tangente correspondiente a estas leyes generales, se han considerado implementaciones numéricas basadas en métodos de perturbación.

Otra contribución fundamental de este trabajo es el desarrollo de técnicas para el remallado adaptativo, de las que se presentarán distintos enfoques. Por un lado, las técnicas basadas en métricas, incluyendo los enfoques *level-set* y *Hessiano*. Estas técnicas son de propósito general y pueden considerarse tanto en la aplicación de problemas estructurales como en problemas de mecánica de fluidos. Por otro lado, se presenta el método de estimación de errores **SPR**, más convencional que los anteriores. En este ámbito, la contribución de este trabajo consiste en la estimación de error mediante las técnicas de *Hessiano* y **SPR** para la aplicación a problemas de contacto numérico. Con los desarrollos previamente introducidos, estaremos en disposición de introducir los casos de aplicación centrados en el contexto de procesos de estampado. Es relevante destacar que estos ejemplos son comparados con las soluciones de referencia disponibles en la bibliografía como forma de validar los desarrollos presentados hasta este punto.

El presente documento está organizado de la siguiente manera. El primer capítulo establece los objetivos y revisa la bibliografía acerca de los temas clave de este trabajo. El segundo capítulo hace una introducción de la mecánica de medios continuos y los conceptos relativos al **Método de los Elementos Finitos (MEF)**, necesarios en los desarrollos que se presentarán en los capítulos siguientes. El tercer capítulo aborda la formulación del elemento sólido-lámina, así como del elemento de lámina sin grados de libertad de rotación que inspira el sólido-lámina desarrollado. Esta parte muestra varios ejemplos académicos que son comúnmente empleados en la bibliografía como problemas de referencia de láminas. El cuarto capítulo presenta la formulación desarrollada para la resolución de problemas de contacto numérico, consistente en una formulación implícita de integración exacta mediante métodos mortero y *multiplicadores de Lagrange* aumentados duales. Este capítulo incluye, asimismo, varios ejemplos comúnmente encontrados en la bibliografía, que generalmente son considerados para su validación. El quinto capítulo presenta la formulación de plasticidad empleada, incluyendo algunos detalles técnicos desde el punto de vista de la implementación, así como varios ejemplos de validación. El sexto capítulo muestra los algoritmos de remallado adaptativo desarrollados en el contexto de este trabajo, y presenta varios ejemplos, que incluyen no solo casos estructurales, sino también de mecánica de fluidos. El séptimo capítulo encapsula algunos casos de validación y aplicación para procesos de estampado. El capítulo final comprende las conclusiones, así como los trabajos que podrían continuar el presente estudio.

Palabras clave: Conformado metálico, estampado, láminas, sólido-lámina, contacto, plasticidad, mallado adaptativo

Acknowledgements

I still remember the day I arrived to Barcelona 5 years ago to start my master thesis in **FSI**. My car, that I was supposed to take in order to bring all my stuff from Paris to Barcelona, just burn out the week before and I needed to take the **TGV** instead, with all my stuff. It is because of that I want to acknowledge in first place *Katie*, one of my roomies when I arrived to Barcelona, who helped me that first day that I arrived to Barcelona to settle.

The next person I meet and I want to thank is my director *Eugenio Oñate Ibañez de Navarra*, who brings me the opportunity to work in this thesis, and had enough patience to deal with all the delays and problems that I had during all this time. Especially the patience to deal with my continuous trips between Barcelona and Paris. I hope the result obtained in this work was worth it. I would like to have a special acknowledge to *Merce Alberich*, who helped me with a lot of paper work, and to communicate with *Eugenio*. In the same line, I would like to notice the administratives in the doctoral school, *Silvia* and *Rosa*, who help me with all the documents related with this thesis. I want to mention the support of the *Generalitat de Catalunya* which is funding my work with the grant **FI AGAUR**.

I want to especially thank to my co-director *Riccardo Rossi* for all the help and support during all this time. Particularly in the last year of the thesis, where the relationship became more difficult due to the distance. At the end I hope you accepted *volcanic* personality. I would like to thank to all the *Kratos* staff. Special remarks for *Carlos Roig*, aka *Charlie* for help me with all those programmation questions. To *Jordi Cotela*, who helped me in several technical issues too, and at the end I hope understood my sense of humour. To *Antonia Larese*, who gave me several academic tips. *Ilaria Iaconeta*, who was a good friend and understands the pain of having a thesis finished, but not really. And to *Pooyan Dadvand* for all the technical help and *Kratos* origin history. Thanks to all the team, to the prepost duo *Enrique Escolano-Miguel Pasenau*, to the fantastic duo *Anna-Adrià*, to the interfacier *Javier Garate* and to the doctor mesh *Abel Coll*. Without you, all this work it would have been much more difficult.

Thanks to everyone I meet these years in Barcelona at the **International Center for Numerical Methods in Engineering (CIMNE)**, especially I want to thank to *Rubén Zorrilla* who was my office partner, and we have together so much metahumour, which is the best type of humour. To the rookies that I left in the office, *Riccardo Tosi*, *Marc Nuñez* and *Agustina Giuliadori*. To *Miguel Ángel Celigueta* and *Lucia Barbu*, they are an adorable couple and I wish you a lot of happiness together. To *Eduardo Soudah*, which passion for biomechanics is fascinating. To *Guillermo Casas*, with whom I have had the pleasure to share debates about politics from time to time. *Miguel Maso*, who is a wonderful person, and I wish him the best. To *Pablo Becker* and *Marcelo Raschi*, with whom I have enjoyed technical discussions. *Ignasi Pouplana*, who was the first person I meet in **CIMNE**, and who helped me so much at the beginning, and with whom I enjoyed working at the end. To *Lorenzo Gracia*, who gave me such good times, and enjoyed so many paellas of mine. To *Tomás Varona* whose friendship I value so much and that I hope will last for many years. Finally, to *Alejandro Cornejo*, who helped me so much during the last stages of the thesis, but has also turned out to be one of my best friends. From Barcelona, I also like to address to *Miguel Zatarain*, who helped me to break routine when I was living in Barcelona, and who host me when I was not living there anymore.

To the people from *Munich*, which contribute so much to the `StructuralMechanicsApplication`, and therefore help me directly and indirectly with the developments presented on this work. Remarks for *Philipp Bucher*, who controlled a very relevant part of my developments. And to *Tobias Teschemacher* who is the coolest guy I know, and I had a great time with him. I would like to mention to *Roland Wüchner*, who did everything in order I could go to , but unfortunately I could not go at the end. Deeply thanks to *Mohamed Khalil*, who accompanied me during my first steps in the world of numerical contact. Without you probably this thesis would be right now completely different. Also coming from , thanks to *Elisa Magliozzi* who ended being a very good friend, and to *Anna Rehr*, which master thesis I had the pleasure to supervise. From *Inria* I would like to acknowledge to *Algiane Froehly*, thanks to her, and to the *Mmg* team, I could accomplish a very relevant part of my thesis.

Finally, but not for that reason less important, I want to thank to my family, to my father who has proven once again to have patience with me and my "*obsessive perfectionism*", to my mother who has given me so much support, and has suffered so much for me, and to my sister, that we have both had to leave the "*terreta*" in order to dedicate ourselves to research. But I want to thank especially to my wife *Sara*, who has been the *co-pilot* of this work from the distance, and in the last stage together, she knows everything I have had to go through and do to get here, and sometimes she has had to suffer because of this.

Contents

I Theoretical framework	21
1 Introduction	23
1.1 Evolution of the stamping technology	26
1.1.1 History of the stamping and forming technology	26
1.1.2 Historical review of Finite Element Analysis (FEA) of formability and forming technology	27
1.2 Most common issues in forming processes	27
1.2.1 Springback	27
1.2.2 Wrinkling	28
1.3 Objectives	29
Bibliography of chapter	29
2 Finite Element formulation	33
2.1 Introduction	33
2.2 Finite Element Method	33
2.2.1 Introduction	33
2.2.2 Concept of Finite Element (FE)	34
2.3 Solid mechanics	36
2.3.1 Linear form	36
2.3.2 Non-linear form	36
2.3.2.1 Deformation gradient	37
2.3.2.2 Strain tensors	37
2.4 Finite Element formulation for Computational Structural Dynamics (CSD)	38
2.4.1 Weak formulation in solids	38
2.4.2 TL and Updated Lagrangian (UL) formulation	39
2.4.3 Solution of the non-linear-equilibrium equations system	40
2.4.3.1 Newton-Raphson method	40
2.4.3.2 Line-search	41
2.4.3.3 Arc-length	41
2.4.4 Time integration schemes	42
2.4.4.1 Newmark time scheme	43
2.4.4.2 Bossak algorithm	43
2.4.4.3 α -generalised	43
2.4.4.4 Backward Differentiation Formula (BDF)	44
2.4.5 Damping matrix	44
Bibliography of chapter	45
3 Rotation-free shells and solid-shell elements	49
3.1 Introduction	49
3.1.1 Historical outline	49
3.2 State of the Art in shell formulation	50
3.2.1 Element requisites and stabilisation techniques	50

3.2.2	Shell formulations	51
3.3	Rotation-free shells	52
3.3.1	Introduction	52
3.3.2	Basic Shell Triangle	52
3.3.2.1	Basics	52
3.3.2.2	Shell kinematics	53
3.3.2.3	Constitutive models	53
3.3.3	Enhanced Basic Shell Triangle	53
3.3.3.1	Total Lagrangian formulation	54
3.3.3.1.1	Computation of the membrane strains	54
3.3.3.1.2	Computation of the bending strains	54
3.3.3.1.3	Tangent stiffness matrix	55
3.4	Prismatic solid-shell	55
3.4.1	Introduction	55
3.4.2	Basic kinematics of the standard element	56
3.4.3	Modifications of the standard element	57
3.4.3.1	In plane behaviour	57
3.4.3.2	Transverse shear behaviour	59
3.4.3.3	Transverse normal behaviour	60
3.4.3.3.1	Enhanced Assumed Strain (EAS) formulation	61
3.4.3.3.2	Balance equation	61
3.4.3.3.3	Pull-Back and Push-Forward (Extension of the formulation)	62
3.5	Numerical examples	63
3.5.1	Patch test	63
3.5.1.1	Membrane patch test	63
3.5.1.2	Bending patch test	64
3.5.2	Cantilever	65
3.5.3	Cantilever subjected to end bending moment	66
3.5.4	Frequencies test	67
3.5.5	Cook membrane	68
3.5.6	Scoordelis cylindrical roof test	69
3.5.7	Sphere test	70
3.5.8	Pull-out of an open-ended cylindrical shell	71
3.5.9	Pinched semi-cylindrical	72
3.5.10	Slit test	73
3.5.11	Cylindrical panel test	74
3.5.12	Conical shell test	75
3.5.13	Wrinkling test	76
3.5.14	Fluid Structure Interaction (FSI)-Vein test	77
	Bibliography of chapter	78
4	Contact mechanics	85
4.1	Introduction	85
4.1.1	Historical outline	85
4.1.2	Contact problem	87
4.2	State of the Art in computational contact mechanics	89
4.2.1	Introduction	89
4.2.2	Discretisation	89
4.2.2.1	Node-To-Node (NTN)	90
4.2.2.2	Node-To-Segment (NTS)	90
4.2.2.3	Contact Domain Method (CDM)	91
4.2.2.4	Segment-To-Segment (STS) (<i>Mortar methods</i>)	91
4.2.2.5	Other alternative methods	92
4.2.2.5.1	Isogeometric	92
4.2.2.5.2	Smooth surface approximation	93

4.2.2.6	Conclusion	93
4.2.3	Optimisation method	93
4.2.3.1	Penalty Method (Penalty Method (PM))	93
4.2.3.2	Lagrange Multiplier Method (LMM)	94
4.2.3.3	ALM	94
4.2.3.4	Double Lagrange Multiplier Method (DLMM)	94
4.2.3.5	Other alternative methods	95
4.2.3.5.1	Perturbed Lagrangian	95
4.2.3.5.2	Nitsche	95
4.2.3.5.3	Other minor mentions	95
4.2.4	Conclusion	95
4.2.5	Frictional models	96
4.3	Formulation	96
4.3.1	Introduction	96
4.3.2	Definition of the problem	97
4.3.3	Frictionless contact	97
4.3.3.1	Strong formulation	97
4.3.3.2	Weak formulation	98
4.3.3.2.1	Scalar <i>Lagrange</i> multiplier	98
4.3.3.2.1.1	LMM	98
4.3.3.2.1.2	Penalty	99
4.3.3.2.1.3	ALM	100
4.3.3.2.2	Vector <i>Lagrange</i> multiplier	101
4.3.3.2.2.1	LMM	101
4.3.3.2.2.2	ALM	101
4.3.3.3	Augmented <i>Lagrange</i> multiplier parameters calibration	102
4.3.3.4	Discretisation and numerical integration	103
4.3.3.4.1	Dual <i>Lagrange</i> multipliers	103
4.3.3.4.1.1	Definition	103
4.3.3.4.1.2	Graphical representation	105
4.3.3.4.1.3	Derivatives	105
4.3.3.4.2	Mortar operators	105
4.3.3.4.2.1	Definition	105
4.3.3.4.2.2	Derivatives	108
4.3.3.4.3	Algebraic form of the problem	108
4.3.3.4.3.1	Scalar LMM	108
4.3.3.4.3.2	Components LMM	108
4.3.3.4.3.3	Penalty	109
4.3.3.4.3.4	Scalar ALM	109
4.3.3.4.3.5	Components ALM	109
4.3.3.4.4	Static condensation of the system in considering of the DLMM	110
4.3.3.5	Work-flow. Solution algorithm	111
4.3.3.6	Active set strategy (Semi-smooth Newton Raphson)	112
4.3.4	Frictional contact	113
4.3.4.1	Strong formulation	113
4.3.4.1.1	Tangential contact condition - <i>Coulomb's law</i>	113
4.3.4.2	Weak formulation	114
4.3.4.2.1	LMM	114
4.3.4.2.2	Contact constraints	116
4.3.4.2.3	Penalty	116
4.3.4.2.4	ALM	117
4.3.4.3	Discretisation and numerical integration	118
4.3.4.3.1	Discrete contact condition in tangential direction	118
4.3.4.3.2	Slip definition	118
4.3.4.3.3	Algebraic form of the problem	120

4.3.4.3.3.1	LMM	120
4.3.4.3.3.2	Penalty	121
4.3.4.3.3.3	ALM	121
4.3.4.4	Work-flow. Solution algorithm	122
4.3.4.5	Active set strategy (Semi-smooth Newton Raphson)	123
4.4	Contact detection. Search techniques	123
4.4.1	Introduction	123
4.4.2	Bounding volumes	124
4.4.2.1	Oriented Bounding Box (OBB) implementation	124
4.4.2.1.1	Collision detection	125
4.4.2.1.2	Collision detection with Separating Axis Theorem (SAT)	127
4.4.3	Tree structures	128
4.4.4	Penetration definition	129
4.4.5	Self-contact detection	130
4.4.5.1	Introduction	130
4.4.5.2	Algorithm considered	131
4.4.5.3	Examples	133
4.4.5.3.1	Planes detection	133
4.4.5.3.2	Tubular detection	133
4.4.5.3.3	Contact example (S-shape profile)	134
4.5	Numerical examples	135
4.5.1	Basic patch test	135
4.5.1.1	Frictionless	136
4.5.1.2	Frictional	136
4.5.2	Taylor patch test	137
4.5.2.1	2D	137
4.5.2.2	3D	137
4.5.3	Friction base test	138
4.5.4	Hertz problem	138
4.5.4.1	2D	139
4.5.4.1.1	Plane-sphere	139
4.5.4.1.2	Cylinder-cylinder	140
4.5.4.1.2.1	Frictionless case	140
4.5.4.1.2.2	Frictional case	142
4.5.4.2	3D	142
4.5.4.2.1	Plane-sphere	142
4.5.4.2.2	Sphere-sphere	144
4.5.5	Teeth model	145
4.5.6	Energy conservation	146
4.5.7	Double arc benchmark	148
4.5.7.1	Frictionless	149
4.5.7.2	Frictional	150
4.5.8	Arc pressing block	151
4.5.9	Hyperelastic tubes	152
4.5.10	Contacting cylinders	153
4.5.10.1	Horizontal movement	153
4.5.10.2	Vertical movement	154
4.5.11	Press fit	155
4.5.12	Ironing punch	157
4.5.12.1	Circular ironing	157
4.5.12.2	Shallow ironing	159
4.6	Derivatives for contact mechanics linearization	161
4.6.1	Derivatives for 2D contact	161
4.6.1.1	Jacobians	162
4.6.1.1.1	Theory	162

4.6.1.1.2	Convergence study	162
4.6.1.2	Shape functions	163
4.6.1.2.1	Theory	163
4.6.1.2.2	Integration segments	164
4.6.1.2.3	Local coordinates (<i>Gauss</i> points)	165
4.6.1.2.4	Convergence study	165
4.6.1.3	Dual shape functions	166
4.6.1.3.1	Theory	166
4.6.1.3.2	Convergence study	166
4.6.1.4	Normal and tangent vectors	167
4.6.1.4.1	Theory	167
4.6.1.4.2	Convergence study	168
4.6.2	Derivatives for 3D contact	169
4.6.2.1	Jacobians	170
4.6.2.1.1	Theory	170
4.6.2.1.1.1	Integration segments derivatives	170
4.6.2.1.1.2	Jacobian derivatives	171
4.6.2.1.2	Convergence study	172
4.6.2.2	Shape functions	173
4.6.2.2.1	Theory	173
4.6.2.2.2	Local coordinates (<i>Gauss</i> points)	174
4.6.2.2.3	Convergence study	175
4.6.2.3	Dual shape functions	176
4.6.2.3.1	Theory	176
4.6.2.3.2	Convergence study	176
4.6.2.4	Normal and tangent vectors	177
4.6.2.4.1	Theory	177
4.6.2.4.2	Convergence study	178
	Bibliography of chapter	179
5	Plasticity	189
5.1	Introduction	189
5.1.1	Historical outline	189
5.2	State of the Art in numerical plasticity	190
5.2.1	Historical outline	190
5.2.2	Constitutive models	191
5.2.2.1	Yield criteria	191
5.2.2.2	Yield surface hardening	192
5.3	Finite strain elasto-plastic models	192
5.3.1	Fundamentals	193
5.3.2	Plastic loading surface	194
5.3.2.1	Introduction	194
5.3.2.2	Isotropic hardening	194
5.3.2.3	Kinematic hardening	194
5.3.2.4	Stress-strain relation	195
5.3.2.5	Stability condition	195
5.3.2.5.1	Local stability	195
5.3.2.5.2	Global stability	196
5.3.2.6	Condition of unicity of solution	196
5.3.3	Finite strain return mapping operations	196
5.4	Large deformation elastic models	197
5.4.1	Introduction	197
5.4.2	<i>Neo-Hookean</i> material	198
5.4.3	<i>Kirchhoff</i> material	198
5.5	Implementation details	198

5.5.1	Introduction	198
5.5.2	Class structure in <i>Kratos</i>	199
5.5.3	Numerical implementation tangent constitutive tensor	200
5.5.3.1	Forward Finite Differences (FD)	201
5.5.3.2	Centered FD	201
5.5.3.3	Numerical details	201
5.6	Numerical examples	202
5.6.1	Introduction	202
5.6.2	Cube minimal example	202
5.6.3	Tensile test	203
5.6.4	Application in Computational Contact Mechanics (CCM) . Gears example	204
	Bibliography of chapter	206
6	Adaptative remeshing	211
6.1	Introduction	211
6.2	State of the Art in mesh refinement	212
6.2.1	Mesh generation	212
6.2.2	Adaptive finite element refinement techniques	213
6.3	Hessian based remeshing technique	214
6.3.1	Error estimation	214
6.3.1.1	Upper bound on the interpolation error	214
6.3.1.2	Numerical computation of the interpolation error	216
6.3.1.3	Mesh adaptation	217
6.3.1.4	Computation of the relative error	217
6.3.2	Metric based remeshing	218
6.3.2.1	Concept of metric	218
6.3.2.2	Metric intersection	219
6.3.3	Hessian based metric measure	220
6.3.3.1	<i>Hessian</i> theory	220
6.3.3.2	Numerical example	222
6.4	<i>Level set</i> based remeshing technique	223
6.4.1	Theory	223
6.4.2	Numerical example	224
6.5	SPR based remeshing technique	225
6.5.1	Introduction	225
6.5.2	Theory	225
6.5.2.1	Error measures	225
6.5.2.2	Error estimation	225
6.5.2.3	Recovery based methods	226
6.5.2.4	Superconvergence concept	226
6.5.2.5	SPR calculation	227
6.5.2.6	Local mesh size	228
6.5.2.6.1	Compute h	228
6.5.2.6.2	Compute respective metric	229
6.5.3	Example	229
6.6	Internal values interpolation	229
6.6.1	Theory	229
6.6.2	Example	230
6.7	Integration points values extrapolation	230
6.7.1	Introduction	231
6.7.2	Theory	231
6.7.3	Numerical example	232
6.8	Adaptive remeshing methods applied on CCM	232
6.8.1	Level set metric	233
6.8.1.1	Theory	233

6.8.1.2	Numerical example	233
6.8.2	Hessian metric	234
6.8.2.1	Theory	234
6.8.2.2	Numerical example	234
6.8.2.2.1	Simplest patch test	234
6.8.2.2.2	Punch test	235
6.8.3	SPR metric	237
6.8.3.1	Theory	237
6.8.3.2	Numerical examples	239
6.8.3.2.1	Simplest patch test	239
6.8.3.2.2	Punch test	239
6.9	Remeshing workflow	240
6.9.1	<i>Level set</i> and <i>Hessian</i> remeshing	240
6.9.1.1	Standard problem	241
6.9.1.2	Contact problem	241
6.9.2	SPR remeshing	241
6.9.2.1	Standard problem	241
6.9.2.2	Contact problem	242
6.10	Numerical examples	242
6.10.1	Coarse sphere. <i>Level set</i>	242
6.10.2	Beam problem. <i>Hessian</i> of displacement	243
6.10.3	<i>Hertz</i> problem. <i>Hessian</i> of contact and Von Mises (VM) stress	243
6.10.4	<i>Hertz</i> problem. SPR error computing	244
6.10.5	Contacting cylinders with adaptive remeshing	246
6.11	CFD numerical examples	249
6.11.1	Cavity. <i>Level set</i>	249
6.11.2	Embedded cylinder. <i>Level set</i>	249
6.11.3	Box isosurface example. <i>Level set</i>	250
6.11.4	<i>Lamborghini</i> . <i>Level set</i>	251
6.11.5	Channel CFD . <i>Hessian</i> of velocity	251
6.11.5.1	2D cylinder channel	252
6.11.5.2	3D sphere channel	253
	Bibliography of chapter	254

II Application 259

7	Application	261
7.1	Introduction	261
7.2	Cylinder punch	261
7.3	Spherical punch	262
	Bibliography of chapter	264

III Conclusions 267

8	Final conclusions	269
8.1	Introduction	269
8.2	Rotation free shells and solid-shell elements. Solid-shell	269
8.3	Contact mechanics	270
8.4	Plasticity	271
8.5	Adaptive remeshing	272
8.6	Application cases	272
8.7	Other remarks	272
8.7.1	Other developments	272
8.7.2	Implicit approach	273

8.7.3	Programming	273
	Bibliography of chapter	273
9	Future works	277
9.1	Solid-shell implementations	277
9.2	Contact mechanics	277
9.3	Plasticity	278
9.4	Adaptive remeshing	278
9.5	Metal stamping	278
9.6	Other points	278
	Bibliography of chapter	278
	Appendices	281
	Appendices	
A	Theoretical complements	285
A.1	Pull-Back, Push-Forward fundamental concepts	285
A.2	Mortar segmentation	285
A.2.1	Introduction	285
A.2.2	Exact integration vs Collocation	286
A.2.2.1	Theory	286
A.2.2.2	Solution study	287
A.2.3	Delaunay vs Convex polygon construct	289
A.3	Mesh tying	289
A.3.1	Introduction	289
A.3.2	Strong formulation	290
A.3.3	Weak formulation	290
A.3.4	Discretisation and numerical integration	291
A.3.5	Numerical example	292
	Bibliography of chapter	292
B	Implementation	295
B.1	Introduction	295
B.2	Kratos Multiphysics	295
B.2.1	Introduction	295
B.2.2	Framework (Application Programming Interface (API))	296
B.2.3	Introduction. Main classes	296
B.2.4	Model	297
B.2.4.1	Introduction	297
B.2.4.2	ModelPart	297
B.2.4.3	Submodelparts	297
B.2.5	Base classes	298
B.2.6	Geometry	298
B.2.7	Strategy	298
B.2.8	Input-Output (IO) classes	298
B.2.9	Testing framework	298
B.3	Mmg library	298
B.3.1	What is Mmg and how does it work?	298
B.3.2	Integration between Mmg and Kratos	300
B.3.2.1	Introduction	300
B.3.2.2	Class structure	300
B.3.2.3	Submodelpart recovery	300
	Bibliography of chapter	302

C Automatic differentiation	305
C.1 Introduction	305
C.2 Mathematical concepts	306
C.3 Implementation	307
C.3.1 Principles of Automatic Differentiation	307
C.3.1.1 Derivation modes	307
C.3.1.2 Automatic Differentiation exceptions	308
C.3.2 Kratos integration	309
Bibliography of chapter	310
D Constrained optimisation problems	313
D.1 Introduction	313
D.2 Penalty method	314
D.2.1 Introduction	314
D.2.2 Formulation	314
D.2.3 Applicability on contact problems	315
D.2.3.1 Adapted Penalty Method	315
D.3 Lagrange Multiplier method	316
D.3.1 Introduction	316
D.3.2 Formulation	316
D.3.3 Applicability on contact problems	317
D.4 Augmented Lagrange Multiplier method	318
D.4.1 Introduction	318
D.4.2 Formulation	318
D.4.2.1 Standard formulation	318
D.4.2.2 Uzawa iteration	319
D.4.3 Applicability on contact problems	319
D.4.3.1 Adapted Augmented Lagrangian Method	320
D.5 MultiPoint Constraint (Master-Slave elimination method)	320
D.5.1 Introduction	321
D.5.2 Formulation	321
D.5.3 Applicability on contact problems	321
D.6 Summary of the different methods	322
D.7 Numerical examples	322
D.7.1 Initial spring wall problem	322
D.7.1.1 Penalty method	323
D.7.1.2 Lagrange multiplier method	324
D.7.2 Non-linear spring contact problem with wall	324
D.7.2.1 Initial wall	325
D.7.2.1.1 Lagrange multiplier method	325
D.7.2.1.1.1 Active set 1	325
D.7.2.1.1.2 Active set 2	326
D.7.2.1.1.3 Active set 3	326
D.7.2.2 Moved wall	327
D.7.2.2.1 Penalty method	327
D.7.2.2.2 Lagrange multiplier method	328
D.7.2.2.2.1 Active set 1	328
D.7.2.2.2.2 Active set 2	329
D.7.2.2.2.3 Active set 3	330
D.7.2.2.3 Augmented Lagrange multiplier method	331
D.7.2.2.3.1 $\epsilon = 0.5$	332
D.7.2.2.3.2 $\epsilon = 1.0$	332
D.7.2.2.3.3 $\epsilon = 5.0$	333
D.7.2.2.3.4 $\epsilon = 10.0$	334
D.7.3 Over-constrained optimisation problem	334

Bibliography of chapter	336
E Mortar mapper	339
E.1 Introduction	339
E.2 Theory	340
E.2.1 General mapping theory	340
E.2.2 Dual <i>Lagrange</i> multiplier mapping	341
E.2.2.1 Introduction	341
E.2.2.2 Theory	341
E.2.3 Discontinuous meshes mapping	342
E.2.3.1 Introduction	342
E.2.3.2 Theory	342
E.3 Numerical examples	343
E.3.1 Non-matching meshes of triangles	343
E.3.2 Non-matching meshes of triangles and quadrilaterals	344
E.3.3 Discontinuous meshes	344
Bibliography of chapter	346
List of Tables	349
List of Figures	351
List of Algorithms	357
Glossaries	359
Acronyms	359
Mathematical symbols	363
Rotation-free shell and solid-shell elements	364
Constraint enforcement and optimization	365
Finite element framework	366
Other terms	368
Mortar formulation	368
Contact mechanics	369
Adaptive remeshing	371
Plasticity and constitutive relations	373

Part I

Theoretical framework

Chapter 1

Introduction

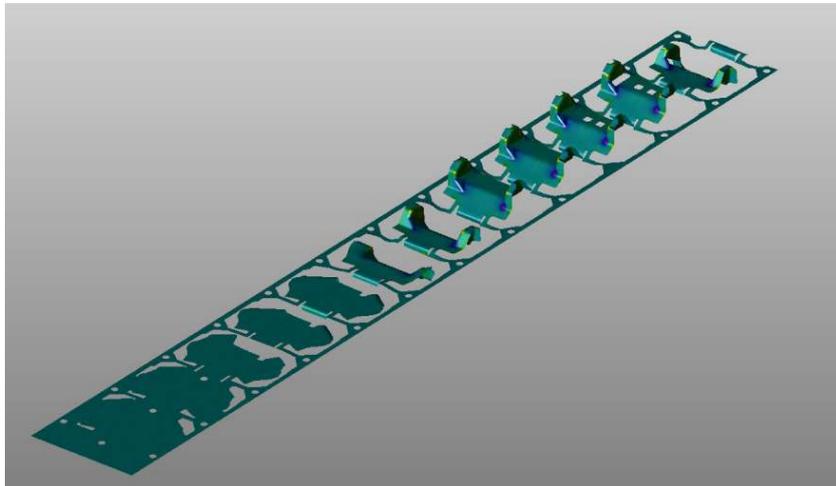


Figure 1.1: Examples of numerical simulation of forming. Source[OnIShe]

Stamping is a kind of manufacturing technology[BookAT12b; BookHu+13], where a sheet blank that has a simple shape is plastically formed between tools or dies, to obtain product components with certain shape, size, and performance (see Figure 1.1). Sheet metal, mould and stamping equipment are three major factors for stamping. Indeed, sheet metal forming processes usually produce little scraps and generate the final part geometry in a very short time, usually in one stroke or a few strokes of a press. As a result, sheet forming offers potential savings in energy and material, especially in medium and large production quantities, where tool costs can be easily amortised. We can differentiate mainly two different technologies according to the working temperature, the hot stamping and the cold stamping; the first one is suitable to process a kind of sheet which has high resistance to deformation and low plasticity, while the second one is employed for metal sheet stamping processes at room temperature.

The concept of formability[BookBan10] can be introduced, which is the capability of sheet metal to undergo plastic deformation to a given shape without defects, and thus is one of the most relevant fields of study for the stamping processes. The defects have to be considered separately for the fundamental sheet metal forming procedures of deep drawing and stretching. The difference between these types of stamping procedures is based on the mechanics of the forming process. Figure 1.2 illustrates that formability is a complex characteristic.

The increase along the time of the costs of material, energy and manpower require that sheet metal forming processes and tooling be projected and developed avoiding as much as possible the trial and error methodology, with the shortest possible lead times. Owing to this reason, the concept of virtual manufacturing has been developed in order to increase the industrial performances[BookBan10], being the most efficient way to reduce manufacturing times and improving the final quality of the products. The finite element method is currently the most widely used numerical procedure for simulating sheet metal forming processes. The structure of an expert system for the analysis of sheet metal formability is illustrated by Figure 1.3. For the expert system[OnIShe] two broad divisions of methodologies can

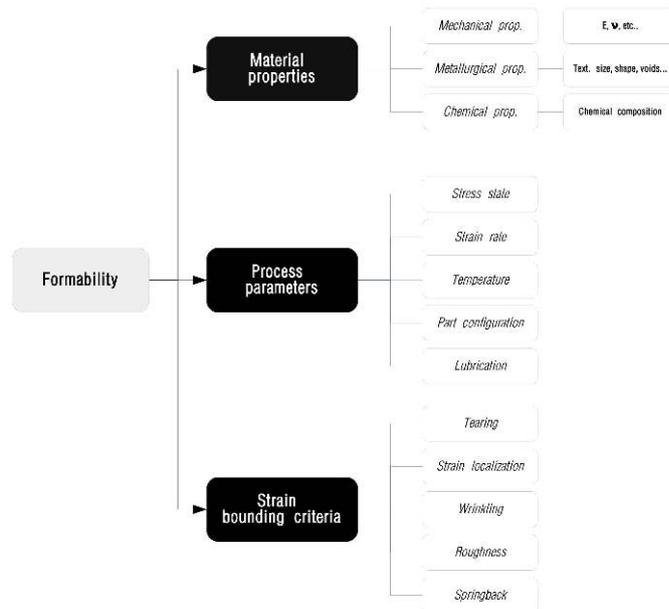


Figure 1.2: Parameters influencing sheet metal formability. Inspired in [BookBan10]

be applied:

- **Inverse One-step:** In this approach all the deformation is assumed to happen in one increment or step and is the inverse of the process which the simulation is meant to represent. The mesh initially is considered with the shape and material characteristics of the finished geometry, and is deformed to the flat pattern blank. Then the strain computed in this inverse forming operation is then inverted to predict the deformation potential of the flat blank being deformed into the final part shape.
- **Incremental analysis:** This method starts with the mesh of the flat blank and simulate the deformation of the blank inside of tools modelled to represent a proposed manufacturing process. This incremental forming is computed "forward" from initial shape to finals, and is calculated over a number of time increments for start to finish.

As the **Incremental analysis** includes the model of the tooling and allows for the definition of boundary conditions which more fully replicate the manufacturing proposal, incremental methods are more commonly used for process validation; in the other hand, **Inverse One-step** with its lack of tooling and therefore poor representation of process is limited to geometry based feasibility checks.

The increasing acceptance of these numerical approaches [BookBHS07] within both research and industrial environments is due to improved awareness, enhanced maturity of computational models and associated algorithms and, more importantly, dramatic increases in computational power/cost ratios, due to this the cost of the simulations has dropped drastically in the last years.

We can highlight the fact that during the last years a lot of significant advances have been archived in the development of finite element tools for the simulation of forming processes. Now the use of commercial codes is widely used for the design and optimisation of these processes, particularly in highly competitive sectors of manufacturing industry. The widespread acceptance of such methods stems in their capabilities to predict deformed shapes, manufacturing defects, forces involved in the process, residual stresses and inelastic strains. Such predictions, which can be usually obtained through relatively inexpensive numerical simulations, are of considerable assistance to the designer of forming operations, allowing to optimise the process parameters and manufactured product properties as well as significantly reducing conception-to-production times. Despite such advances, there are still many complex problems to solve, some of these problems are related directly with the complexity of the manufacturing process; but from a mechanical point of view forming problems are complex due to fact they join all the origins of non-linearity in the mechanical problems, which are [BookWri08]:

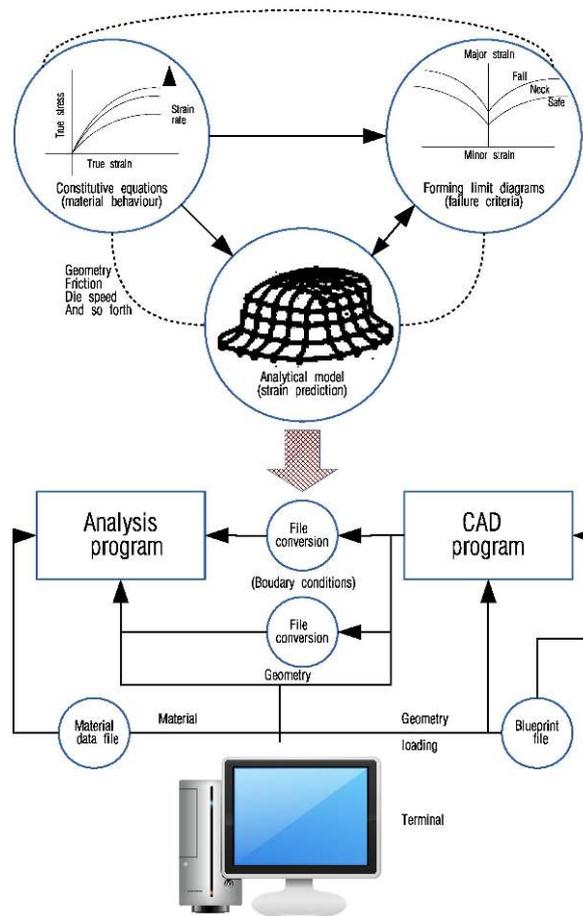


Figure 1.3: Structure of an expert system for the analysis of sheet metal formability. Inspired in [BookBan10]

- **Geometrical nonlinearity:** Large displacements and rotations have to be considered.
- **Finite deformations:** Not only the displacements are large, but also the strains.
- **Physical nonlinearity:** The non linear behaviour of the material, like in the case plasticity, essential to work in forming simulations.
- **Stability problems:** Geometrical and material instability. Geometrical instability includes bifurcations like buckling, or any snap-through behaviour. The material instability could be because of a instability in the equilibrium equations.
- **Nonlinear boundary conditions:** Contact between two bodies or deformation dependent of the loading, phenomena very frequent in the stamping processes.
- **Coupled problems:** Like thermomechanical problems, which are very frequent in forming processes or **FSI**, which appears for example in the hydroforming manufacturing processes.

For the practical [BookAT12b] and efficient use of these technologies, the knowledge of the principal variables of the sheet metal forming processes and their interactions is required. These variables include:

1. The flow behaviour and formability of the formed sheet material under processing conditions
2. Die geometry, materials and coatings
3. Friction and lubrication
4. The mechanics of deformation, *i.e.* strains, stresses and forces

5. Characteristics of the sheet metal forming presses and tooling
6. Geometry, tolerances, surface finish and mechanical properties of the formed parts
7. The effects of the process on the environment

1.1 Evolution of the stamping technology

1.1.1 History of the stamping and forming technology

Stamping[BookHu+13] is a kind of metal processing method that has already existed in ancient times. The ancient struck preformed metal blocks with hammers. Lately, 4000 years ago, in ancient China[ArtKC06] metal plastic work can be traced back, mainly forging as a processing method. Today, sheet metal forming can be considered a fully automatic mass production industry. Its applications extend from household commodities like pans or cans, to automotive and aeronautical industry, among other applications. We can consider the rise of the modern stamping era 200 years ago, during the Industrial Revolution in **United Kingdom (UK)**. During these years, the development of modern iron technology contributed to the development of stamping procedures and machinery.

The car industry was probably the main responsible of the development of the stamping technology when the price became a problem in the process of car popularisation. Thus using stamping to produce automotive parts became popular because of its contribution to lower the production cost; nowadays about 60-65% of the car parts are made by stamping.

We can enumerate the five stages that the stamping technology has experienced in developed countries[BookHu+13] during the 20th century as follows:

1. Before the **Second World War (WWII)**, the stamping line was composed of manually loaded double-action drawing press and several single-action drawing presses. Being manually feed carried as result:
 - Low production efficiency.
 - Poor security conditions.
 - Poor product quality.
2. During the **1960s**, still the same technology was considered, but each press was equipped with a robot hand in order to reduce manual labour.
3. In the **1970s**, the automatic stamping line was established and could be controlled only by one or two operators. This increased productivity, as well as lowered costs and improved quality.
4. Later, in the **1980s**, owing to the emergence of multistation presses, a double-action drawing press and a multi-station press were used for composing a stamping line.
5. Since the **1990s**, the double-action drawing press was replaced as the leading equipment of stamping lines. To replace it, **Numerical Controlled (NC)** hydraulic cushion was installed into the first station of the large multi-station press, which made this press form a flexible production unit independently.

Moving forward to the present, the automotive industry is the pillar industry of the national economy in many industrially developed countries or newly industrialising countries. For this reason the development of numerical methods and design technologies has been crucial; where **Computer-Aided Design (CAD)/Computer Aided Engineering (CAE)/Computer Aided Manufacturing (CAM)** integration technology is playing an increasingly important role in product design, mould design and manufacturing process.

1.1.2 Historical review of FEA of formability and forming technology

FEM[BookBHS07; BookBan10] has been developed and used for sheet forming simulations since the **1960s** and early **1970s**, when the continuum mechanics foundations for problems involving large displacements and large strains became well established. The establishment of 3D **Non-Linear (NL)** shell theory made it possible to analyse the forming process of some complex parts such as car bodies cover, and the algorithm for contact and friction problems took geometric nonlinearity into consideration.

It was at the beginning of the **1970s** when the first theoretically correct **FE**-formulation of the large deformation problem was presented by *Hibbit et al.*[ArtHMR70] in a **TL** frame. Five years later, in 1975, *McMeeking*[ArtMR75], *Rice* and *Bathe*[ArtBRW75] presented correct **UL** formulations of the problem. At that time the simulations were based in plane strain or axisymmetric formulations, and at that time sometimes the models lack full consistency from a theoretical point of view. Later the 3D formulation of the sheet forming problem was presented by *Wang* and *Budiansky* in 1978[ArtWB78]. The following decade saw a high activity in the field.

The development took place along several different paths. Methods based on 2D as well as 3D formulations were developed. Depending on the choice of description of motion, type of constitutive relations, and solution procedures, the methods were described as the solid approach, the *static-implicit* approach, the *static-explicit* approach, the *rigid-plastic* approach and the flow approach (which have the disadvantage that any phenomenon related with elasticity cannot be simulated, as the *springback*).

During this period of time (the **1980s**) the flow and *rigid-plastic* approaches were more popular than the *static-implicit one*, mainly because they were more stable. The practical application of sheet forming simulations was too unstable procedures and excessive computing times, even for very small problems.

In 1989 *Honecker*[ArtHen89] and *Mattiasson* presented results from a study, in which the *dynamic-explicit* approach was evaluated in application to sheet metal stamping, the results from this study were very promising. Since sheet forming processes usually have much longer durations than what the dynamic, explicit method normally is intended for, the prerequisites for the problem have to be modified in some way, in order to fully utilise the benefits of the method, but with additional backwards like virtual inertial forces.

At early **1990s** there was significant increase of the practical utilisation of sheet forming simulations within the industry, and from the middle of this decade most companies within the automotive industry were performing sheet stamping simulations on a regular basis. *Dynamic-explicit* codes were dominating the software market, codes like **LS-DYNA** and **ABAQUS/Explicit** appeared in the market. From the side of the *static-implicit* the code **AutoForm** emerged from a research project at **Eidgenössische Technische Hochschule (ETH)** in *Zurich* at that time, which used some innovative algorithms to enhance stability and computational efficiency. Besides this code is very extended today, the software market is still dominated by various *dynamic-explicit* codes like **LS-DYNA**, **ABAQUS/Explicit**, **PAM-STAMP 2G** and **STAMPAK**.

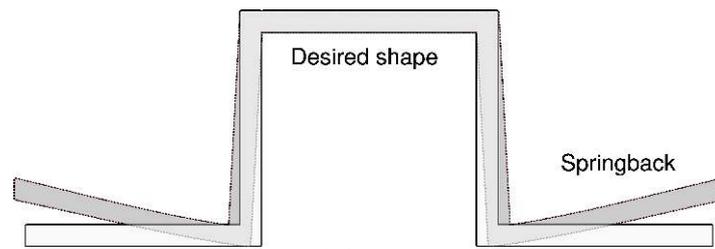
To promote the research of sheet metal forming simulation and investigate the reliability of numerical analysis algorithms, international research organisations have designed a series of standard questions, such as **Ohio State University (OSU)**, **Verein Deutscher Ingenieure (VDI)** and **NUMISHEET**. These questions aim to assess the performance of the existing software to predict *splitting*, *wrinkling*, *buckling*, and *springback*.

1.2 Most common issues in forming processes

The following are the most common issues that every engineer who want to work with forming simulations should deal with to perform a realistic and correct simulation of the problem. In addition to the following one the splitting problem can be mentioned, which as the name indicates this defect appears when the material breaks, producing splits in it.

1.2.1 Springback

Springback[BookBan07] is a particularly critical aspect of sheet metal forming (See Figure 1.4). Even relatively small amounts of *springback* in structures that are formed to a significant depth may cause the blank to distort to the point that tolerances cannot be held. Such geometrical differences caused by *springback* mainly occur for three reasons:

Figure 1.4: *Springback*

- Due to interferences in the residual stress state of the formed parts
- After removal of the external forces
- Due to the *Bauschinger* [ArtBau86] effect. In a typical sheet metal forming process, a considerable amount of material undergoes non-proportional loading. It is well known that the mechanical response of metals depends not only on the current state but also on the previous deformation history. One of the phenomena related to the change of mechanical properties during a non-proportional loading is the *Bauschinger* effect [BookBan10]. New materials such as high-strength steel, aluminium and magnesium are particularly prone to *springback*.

To compensate for the *springback* effects the tool must be reworked, which causes further costs and requires more time. To avoid these additional efforts during die design, the simulation of *springback* becomes more and more important in the forming simulation.

1.2.2 Wrinkling

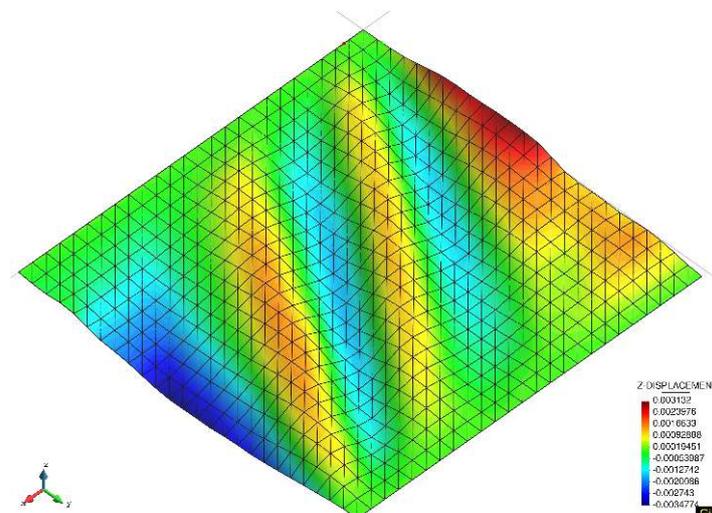


Figure 1.5: Wrinkling phenomena

Compressive stresses are formed in the plane of the sheet results in *wrinkling* [ArtFO11; ArtCFM03; ArtCW00; ArtXSZ15], this kind of instability could occur during the forming process, producing an undesired shape and defects in the final product. The tendency of *wrinkling* increases with:

- Unsupported length of sheet metal
- Decreasing thickness
- Not uniformity in thickness
- Lubricants trapped can also contribute to *wrinkling*

At the end this problematic is an instability that occurs due to the same reasons that buckling occurs for one-dimensional elements like beams, extended for two-dimensional elements. The Figure 3.32 shows us some example of *wrinkling* for simple problem of drape simulation.

1.3 Objectives

The main objective of this work is to advance in the development of **FEM** for the stamping processes, and do it in such a way that the resulting *product* of this work will be a robust tool with the capability of solve problems demanded by the industry, or in other words, a *product* with an able to compete in the market with the existing commercial solutions. To reach this aim the following goals are set:

- The implementation of a fully functional prismatic solid-shell element based in the formulation of Flores[ArtFlo13c; ArtFlo13a; ArtFlo13b], with the convenient modification of the formulation to be able of computing the **push-forward** and **pull-back**. This goal has been reached satisfactory during the current course of doctorate.
- With the aim of a robust methodology to compute forming processes, the computations shown in this work will focus in **implicit** methods, in contrast to the extended **explicit** approaches. The drawbacks of the implicit method[BookBan07] were the computation speed, the high memory requirements and the frequent convergence problems with complex processes; but fortunately the power of implicit methods is increasing more rapidly than that of explicit methods. Implicit[BookBel+14] methods for the treatment of nonlinear constraints, such as contact and friction, have been improved tremendously. Sparse iterative solvers have also become much more effective.
- The implementation of a fully functional contact method. In the literature[BookWri06; BookLau10; PhDYas11; PhDPop12; BookBel+14; ArtOAM08] it exists an extent discussion of which method could be the more adequate for simulate these kinds of problems. In order to perform the most consistent simulation, the method chosen will be the **mortar method**, which will be based in Popp[PhDPop12] work.
- The implementation of the required constitutive laws in order to obtain a good approach of the behaviours that will experiment the forming problems.
- Strategies of self-refinement[PhDHat03; ArtXSZ15; BookZZT13] will be considered. The objective is to be able of computing any kind problem, starting from a coarse mesh; and refine whenever is necessary to obtain a feasible solution of the problem.
- At the end of the development cycle, different routines coming from the know-how[BookBan10; BookVal10; BookKoc08; BookHu+13; BookAT12a; BookTP07] of the industry will be implemented with the aim of obtain a fully functional commercial software.

For more details related with the objectives the chapters 8.*Final conclusions* and 9.*Future works* contain these points developed in greater depth.

Bibliography

Books

- [BookAT12a] *Sheet Metal Forming - Fundamentals*. Taylan Altan and A. Erman Tekkaya. ASM International. 2012.
- [BookAT12b] *Sheet Metal Forming: Processes and Applications*. Taylan Altan and Erman Tekkaya. ASM International. 2012.
- [BookBan07] *Advanced Methods in Material Forming*. Dorel Banabic. 2007.
- [BookBan10] *Sheet Metal Forming Processes: Constitutive Modelling and Numerical Simulation*. Dorel Banabic. Springer-Verlag Berlin Heidelberg. 1st ed. 2010.
- [BookBel+14] *Nonlinear Finite Elements for Continua and Structures*. Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. Wiley. 2nd ed. 2014.
- [BookBHS07] *Encyclopedia of Computational Mechanics (3 Volume Set)*. René de Borst, T.J.R. Hughes, and Erwin Stein. Wiley. 1st ed. 2007.
- [BookHu+13] *Theories, Methods and Numerical Technology of Sheet Metal Cold and Hot Forming: Analysis, Simulation and Engineering Applications*. Ping Hu, Ning Ma, Li-zhong Liu, and Yi-guo Zhu. Springer-Verlag London. 1st ed. 2013.
- [BookKoc08] *Hydroforming for Advanced Manufacturing*. M. Koc. CRC Press. 2008.
- [BookLau10] *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Tod A. Laursen. Springer. 2010.
- [BookTP07] *Mechanics Modeling of Sheet Metal Forming*. Sing C. Tang and Jwo Pan. Society of Automotive Engineers, Inc. 2007.
- [BookVal10] *Applied Metal Forming: Including FEM Analysis*. Henry S. Valberg. Cambridge University Press. 1st ed. 2010.
- [BookWri06] *Computational Contact Mechanics*. Peter Wriggers. Springer. 2nd. 2006.
- [BookWri08] *Nonlinear finite element methods*. Peter Wriggers. Springer. 1st ed. 2008.
- [BookZZT13] *The Finite Element Method: its Basis and Fundamentals*. O. C. Zienkiewicz, J.Z. Zhu, and Robert L. Taylor. Butterworth-Heinemann. 7th ed. 2013.

Articles

- [ArtBau86] “Über die Veränderung der Elastizitätsgrenze und der Festigkeit des Eisens und Stahls durch Strecken und Quetschen, durch Erwärmen und Abkühlen und durch oftmals wiederholte Beanspruchung”. Johann Bauschinger. In: *Mitteilungen des mechanisch-technischen Laboratoriums der Königlich Technischen Hochschule München*. No. 1, Vol. 13, 1886,
- [ArtBRW75] “Finite element formulations for large deformation dynamic analysis”. Klaus-Jürgen Bathe, Ekkehard Ramm, and Edward L. Wilson. In: *International Journal for Numerical Methods in Engineering*. No. 2, Vol. 9, 1975, pp. 353–386. John Wiley & Sons, Ltd. DOI: [10.1002/nme.1620090207](https://doi.org/10.1002/nme.1620090207).

- [ArtCFM03] “Analytical and numerical investigation of wrinkling for deep-drawn anisotropic metal sheets”. J. P. de Magalhaes Correia, G. Ferron, and L. P. Moreira. In: *International journal of mechanical sciences*. No. 6, Vol. 45, 2003, pp. 1167–1180. Elsevier.
- [ArtCW00] “An analytical model for plate wrinkling under tri-axial loading and its application”. Jian Cao and Xi Wang. In: *International journal of mechanical sciences*. No. 3, Vol. 42, 2000, pp. 617–633. Elsevier. DOI: [10.1016/s0020-7403\(98\)00138-6](https://doi.org/10.1016/s0020-7403(98)00138-6).
- [ArtFlo13a] “A “Prism” solid element for large strain shell analysis”. Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 253, 2013, pp. 274–286. Elsevier. DOI: [10.1016/j.cma.2012.10.001](https://doi.org/10.1016/j.cma.2012.10.001).
- [ArtFlo13b] “Development of a non-linear triangular prism solid-shell element using ANS and EAS techniques”. Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 266, 2013, pp. 81–97. Elsevier. DOI: [10.1016/j.cma.2013.07.014](https://doi.org/10.1016/j.cma.2013.07.014).
- [ArtFlo13c] “Un Elemento Prisma Triangular De Sólido-lámina Para El Análisis Con Grandes Deformaciones”. Fernando G. Flores. In: *Mecánica Computacional*. Vol. Xxxii, 2013, pp. 63–87.
- [ArtFO11] “Wrinkling and folding analysis of elastic membranes using an enhanced rotation-free thin shell triangular element”. Fernando G. Flores and Eugenio Oñate. In: *Finite Elements in Analysis and Design*. No. 9, Vol. 47, 2011, pp. 982–990. Elsevier. DOI: [10.1016/j.finel.2011.03.014](https://doi.org/10.1016/j.finel.2011.03.014).
- [ArtHen89] “Finite-Element Procedures for 3D Sheet Forming Simulation”. A Henecker. In: . Vol. 457, 1989,
- [ArtHMR70] “A finite element formulation for problems of large strain and large displacement”. Hugh D Hibbitt, Pedro V Marcal, and James R Rice. In: *International Journal of Solids and Structures*. No. 8, Vol. 6, 1970, pp. 1069–1086. Elsevier. DOI: [10.1016/0020-7683\(70\)90048-x](https://doi.org/10.1016/0020-7683(70)90048-x).
- [ArtKC06] “Achievements of metal plastic working in ancient China”. WANG Kezhi and S Chen. In: *J Plast Eng*. Vol. 6, 2006, pp. 114–125.
- [ArtMR75] “Finite-element formulations for problems of large elastic-plastic deformation”. Robert M McMeeking and JR Rice. In: *International Journal of Solids and Structures*. No. 5, Vol. 11, 1975, pp. 601–616. Elsevier. DOI: [10.1016/0020-7683\(75\)90033-5](https://doi.org/10.1016/0020-7683(75)90033-5).
- [ArtOAM08] “Algorithms and strategies for treatment of large deformation frictional contact in the numerical simulation of deep drawing process”. M. C. Oliveira, J. L. Alves, and L. F. Menezes. In: *Archives of Computational Methods in Engineering*. No. 2, Vol. 15, 2008, pp. 113–162. Springer. DOI: [10.1007/s11831-008-9018-x](https://doi.org/10.1007/s11831-008-9018-x).
- [ArtWB78] “Analysis of sheet metal stamping by a finite-element method”. N-M Wang and Bernard Budiansky. In: *Journal of Applied Mechanics*. No. 1, Vol. 45, 1978, pp. 73–82. American Society of Mechanical Engineers. DOI: [10.1115/1.3424276](https://doi.org/10.1115/1.3424276).
- [ArtXSZ15] “Drape simulation using solid-shell elements and adaptive mesh subdivision”. Q. Xie, K.Y. Sze, and Y.X. Zhou. In: *Finite Elements in Analysis and Design*. Vol. 106, 2015, pp. 85–102. Elsevier. DOI: [10.1016/j.finel.2015.08.001](https://doi.org/10.1016/j.finel.2015.08.001).

Ph.D.'s thesis

- [PhDHat03] “Automated modeling and remeshing in metal forming simulation”. Nitin Vasant Hattangady. 2003.
- [PhDPop12] “Mortar Methods for Computational Contact Mechanics and General Interface Problems”. Alexander Popp. 2012.
- [PhDYas11] “Computational contact mechanics: geometry, detection and numerical techniques”. Vladislav A. Yastrebov. 2011.

Online resources

- [OnlShe] *Wikipedia. Sheet metal forming simulation*. URL: https://en.wikipedia.org/wiki/Sheet_metal_forming_simulation.

Chapter 2

Finite Element formulation

“The limitations of the human mind are such that it cannot grasp the behaviour of its complex surroundings and creations in one operation. Thus the process of subdividing all systems into their individual components or elements, whose behaviour is readily understood, and then rebuilding the original system from such components to study its behaviour is a natural way in which the engineer, the scientist, or even the economist proceeds.”

Olgierd C. (Olek)
Zienkiewicz[BookZZT13]
(1921 - 2009 AD, British academic of
Polish descent. **FEM** pioneer)

2.1 Introduction

This chapter introduces the concepts underlying the nonlinear continuum mechanics for **FEA**. This is necessary in order to understand the formulation exposed in following chapters, both for solid elements and for the understanding of the contact mechanics theory.

In the first section, the **FEM** theory is introduced (2.2.Finite Element Method), in the second section the solid mechanics theory is presented (2.3.Solid mechanics), the third and last section presents the application of the **FEM** in the resolution of solid mechanics problems (2.4.Finite Element formulation for **CSD**).

2.2 Finite Element Method

2.2.1 Introduction

Physicists, engineers and mathematicians have been considering the **Ordinary Differential Equation (ODE)**, and its generalisation in multiple variables the **Partial Differential Equation (PDE)**, as a key tool in their fields since the calculus was created. Particularly physicists, and by extension engineers too, have been using the **PDE** as a tool to describe a wide variety of physical phenomena such sound, heat, diffusion, electrostatics, electrodynamics, fluid dynamics, elasticity, among others. The description provided by this mathematical tool introduces a continuous problem with a set of **Boundary Conditions (BC)**, solvable only in a very limited range of cases and with a series of hypotheses taken in order to simplify the problem. So that we extend the number of problems we are able to solve we

can *discretise* the problem, and solve approximately considering a numerical approximation instead of an analytical solution. With the advent of modern computers *discrete* problems can generally be solved readily even if the number of elements is very large.

This discretisation process has been an object of discussion, as well as has brought fruitful developments and successive works. Each one of the methodologies arose has its advantages and its disadvantages, so when choosing one it may become a compromise solution or even a solution that takes into consideration several ones at the same time. Mathematicians have developed general techniques applicable to different types of problems[BookZZT13], the most significative ones are the **FD**[ArtRic11], weighted residual procedures[BookFin13], the use of variational methods[ArtPom69], among others.

On the other hand, engineers with their more practical point of view took this decomposition principle and create an analogy between the discrete entities and finite portions of the continuum domain[BookZZT13], leading to the development of the first **FE** methods by *Turner*[ArtTur56] in 1956 when working for *Boeing* group. In any case, the first one considers the term **FEM** to refer to the method was *Clough*[ArtClo60] in 1960. Since then the developments of the **FEM** have been staggering¹, and has been seen as a general method able to deal with any kind of continuum mechanics problem, this means a general method to solve problems defined via **PDE**. This includes the development of different codes for the **FEM**, at the beginning these code developed at *Berkeley* was nameless[BookBel+14], engineers developed new applications by modifying and extending these early codes. The next generations became more sophisticated, name included, and grow and even became a commercial product. After that, many programmes appear, many commercial, many with educational purpose and even **Free and Open Source Software (FOSS)** projects arise.

Other kinds of methods that it is relevant to mention, which follows a similar strategy on dividing the domain in discrete entities, is the **Finite Volume Method (FVM)**[BookLL92]. This method allows to solve conservative problem by applying the divergence theorem in the discrete volumes, then the volume integrals in a **PDE** that contain a divergence term are converted to surface integrals. These terms are then evaluated as fluxes at the surfaces of each finite volume. Due to the latter limitation, only conservative problems can be solved², the **FEM** are a method more suitable to solve multi-physics and more general problems. This is the reason why this methodology is the chosen one, to deal with the physics of the different types of problems presented in this work.

2.2.2 Concept of FE

Here the concept of **FE** early introduced is detailed. Once the continuous problem is formulated in a **PDE** form, we can proceed[BookZZT13] recasting the problem in an alternate form, denominated *weak form* from which we can approximate the solution. A weak form to a set of **PDE** is obtained using the following steps[BookZZT13]:

1. Multiply each equation by an appropriate arbitrary function, denominated *test function*, defined in the domain of interest (Ω^i). The expression obtained is a function of functions called a *functional*.
2. Integrate this product over the space domain of the problem. This can be achieved following different methodologies, the most common use, and the one considered in this work, is the *weighted residuals Galerkin*[ArtGal15] method, usually referenced as *Galerkin* method only. This approach consists in considering the same shape functions for the *test functions* than for the *unkowns*. The method of weighted residuals[BookFin13] is previous to the **FEM**, and its consideration as the integration method leads to the **Generalized Finite Element Method (GFEM)**.
3. Use integration by parts to reduce the order of derivatives to a minimum. Or use more advance techniques as the *Green's theorem*.
4. Introduce **BC** if possible. This depends on the nature of the **BC**, which may lead to a modification of the equations which define the problem, or just simple **BC** directly imposed over the system.

¹For years journals as *Journal of Applied Mechanics* rejected papers on the **FEM** because it was considered no scientifically relevant[Book-Bel+14].

²It is possible to deal with not conservative problems with its respective additional treatment, see for example the work of *Castro*[ArtCGP06].

The former procedure can be summarised conceptually as follows. We can take the Figure 2.1 as reference problems, a stationary heat equation with a non-uniform set of **BC**. Then we can reduce the **FE** problem to solve in the following three steps:

1. We want to solve the *Laplacian* equation for the heat problem on a certain domain with a certain **BC** from Figure 2.1a. Both the geometry and the **BC** considered on this setup are not trivial, meaning is not easy to obtain an analytical solution. Then the weak problem is formulated. The **Left-Hand Side (LHS)** and **Right-Hand Side (RHS)** is obtained from this weak form. We will solve the resulting system as any standard linear solver.
2. The domain is divided in discrete elements, like triangles as seen on Figure 2.1b. The previous weak form is integrated in the discrete mesh, obtaining the respective system of equations.
3. After integrating on this discrete domain we are able to solve the **PDE** with our **BC**, obtaining as result the temperatures seen on Figure 2.1c.

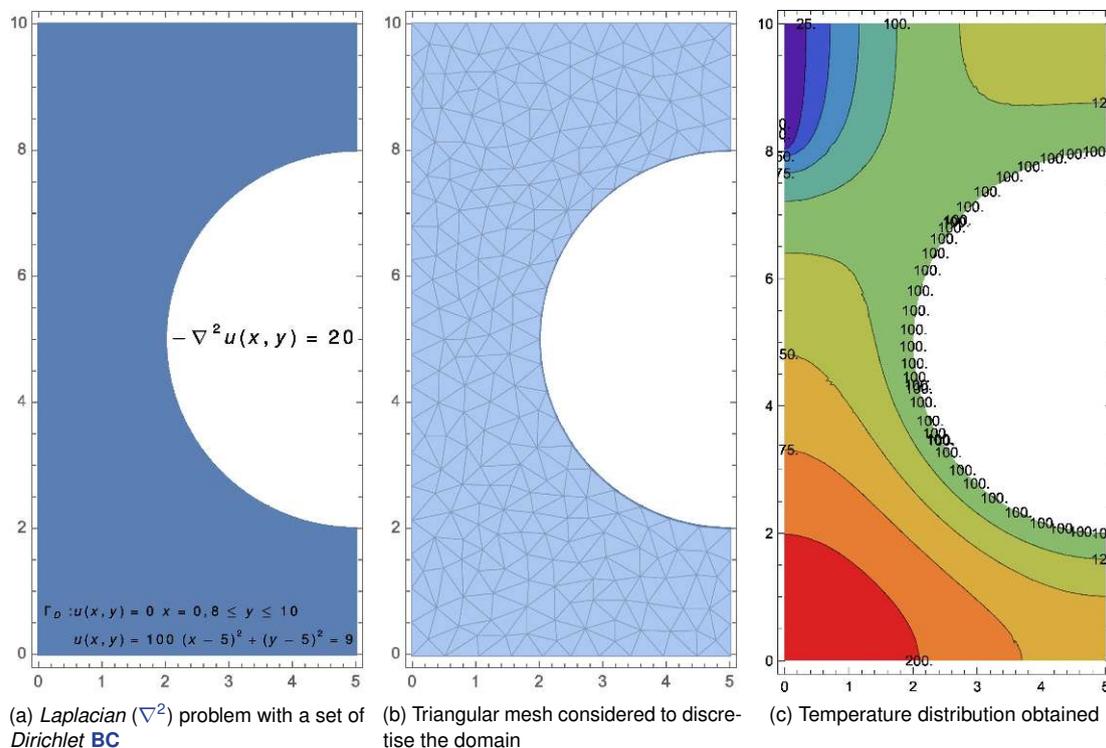


Figure 2.1: Minimal example of **FEA** for a *Laplacian* problem. Resolution using *Mathematica*[OniWol]

In order to obtain the system of equations for the **FEA** problem, we need to derive the respective **LHS** and **RHS**. In order to do so, we need to use the derivative, which is a linear operation, and then can be applied systematically, and therefore it is a potentially automated task. This can be done in fact in an automatic manner using the **Automatic Differentiation (AD)** procedure, which is detailed on the corresponding appendix [C. Automatic differentiation](#).

The **FEM** can be used in order to solve a large range of problems, including complex **NL** problems. Without an understanding of the fundamentals, a finite element program is a black box that provides simulations[BookBel+14]. However nonlinear **FEA** requires the understanding of the underlying theory, as well as an adequate level of experience. On this kind of simulation, many choices and pitfalls are taken into account, without it the analyst could provide an incorrect diagnosis. On **NL** problems bifurcation or localisation may occur, and many times convergence of the numerical analysis is not always obtained in nonlinear problems[BookWri08].

Because of this we directly address the literature where this nonlinear **FEM** is studied on detail. The work of Zienkiewicz[BookZTF14], Belytschko[BookBel+14], de Borst[BookBor+12] and Wriggers[BookWri08] are good references. In summary, any **NL FEA** has the following steps[BookBel+14]:

1. **Development of a model:** Which represents the phenomenon from a physical point of view.
2. **Formulation of the governing equations:** Translate the previous model into a set of **PDE** and its respective **BC** which allows to treat the problem mathematically.
3. **Discretisation of the equations:** This includes the selection of the most adequate method to solve it and the corresponding framework.
4. **Solution of the equations:** This means not only the algebraic resolution of the system of equations, but also the definition of the discrete mesh and the proper solution strategy.
5. **Interpretation of the results:** Including its proper representation and post-process and reach the pertinent conclusions.

We need to differentiate between the origins of the **NL** in order to properly address its problematic. We can list the causes of the nonlinearities in a **FEA** as the following six ones[BookWri08]:

- **Geometrical nonlinearity:** When large displacements and rotations have to be considered, because the effects on the change on the geometry cannot be obviated.
- **Finite deformations:** Here not only the displacements are large but also the strains. Both of this **NL** are discussed on the chapter dedicated to the element developed on this work, see 3.Rotation-free shells and solid-shell elements.
- **Physical nonlinearity:** Many materials depict nonlinear behaviour, like plasticity or damage. This is studied on this work on the respective chapter 5.Plasticity.
- **Stability problems:** We can differentiate between the geometrical and material instabilities. This instability is complex, and usually require specific strategies in order to be taken into account like the *arc-length*.
- **Nonlinear boundary conditions:** Like for example the contact between two bodies. This subject is treated on deep in this work, for more information see 4.Contact mechanics.
- **Coupled problems:** When more than one type of problem is combined. Problems like **FSI**, thermomechanical, chemical reactions, etc.

2.3 Solid mechanics

2.3.1 Linear form

If we consider an **infinitesimal strain-displacement** the equations of *Navier-Cauchy*[BookZZT13] are expressed in tensor notation as Equation (2.1).

Equation of motion (linear momentum): Which is an expression of *Newton's* second law.

$$(2.1a) \quad \nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \rho \ddot{\mathbf{u}}$$

Strain-displacement equations: For infinitesimal strain ($\|\nabla \mathbf{u}\| \ll 1$ and $\|\mathbf{u}\| \ll 1$) $\mathbf{E} \approx \mathbf{e} \approx \boldsymbol{\varepsilon}$.

$$(2.1b) \quad \boldsymbol{\varepsilon} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$$

Constitutive equations: For elastic materials, *Hooke's* law represents the material behaviour and relates the unknown stresses and strains. In this equation \mathbb{C} represents the fourth-order constitutive tensor.

$$(2.1c) \quad \boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}) \text{ for elastic materials } \boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}$$

2.3.2 Non-linear form

Now we consider a **NL** geometry and kinematics[OnlFin; BookBHS07; BookNPO09] the development for problems where the solid deals with deformations in which **both rotations and strains are arbitrarily large**.

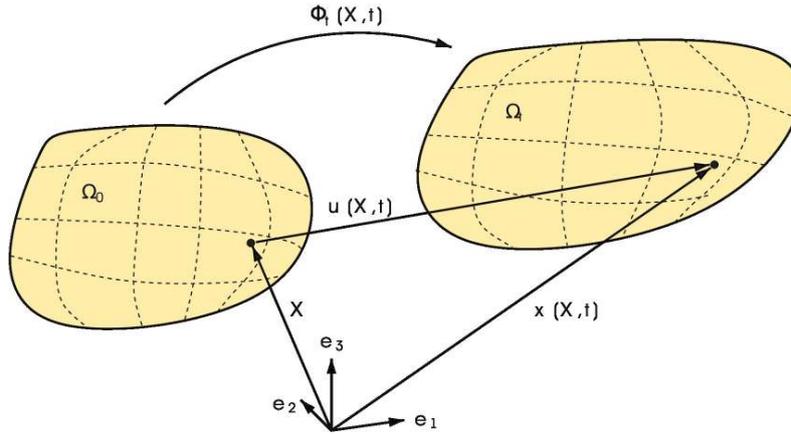


Figure 2.2: Deformation of a continuum body

2.3.2.1 Deformation gradient

Considering a generic continuum body Ω (Figure 2.2) which occupies a region of the three-dimensional *Euclidean* space \mathbb{R}^3 in its reference configuration, with boundaries $\partial\Omega$ or Γ . Let Ω be subjected to a motion ϕ so that for each time t , the deformation (2.2a) maps each material particle \mathbf{X} of Ω into the position \mathbf{x} it occupies at time t . The set (2.2b) is called the *current or deformed* configuration. The two-point tensor \mathbf{F} defined in (2.2a) by (3.26) is termed the *deformation gradient*, where $\{\mathbf{E}_i\}_{i=1,2,3}$ and $\{\mathbf{e}_i\}_{i=1,2,3}$ are fixed orthonormal bases in the reference and deformed configuration, respectively, typically chosen to be coincident with the standard basis in \mathbb{R}^3 . The *Jacobian* (J) of mapping (2.2a) can be represented as (3.15a).

$$(2.2a) \quad \phi(\cdot, t) : \Omega \rightarrow \mathbb{R}^3$$

$$(2.2b) \quad \mathbf{x} = \phi(\mathbf{X}, t)$$

$$(2.2c) \quad \mathbf{F}(\mathbf{X}, t) = D\phi(\mathbf{X}, t) = \frac{\partial \phi_i}{\partial X_I} \mathbf{e}_i \otimes \mathbf{E}_I$$

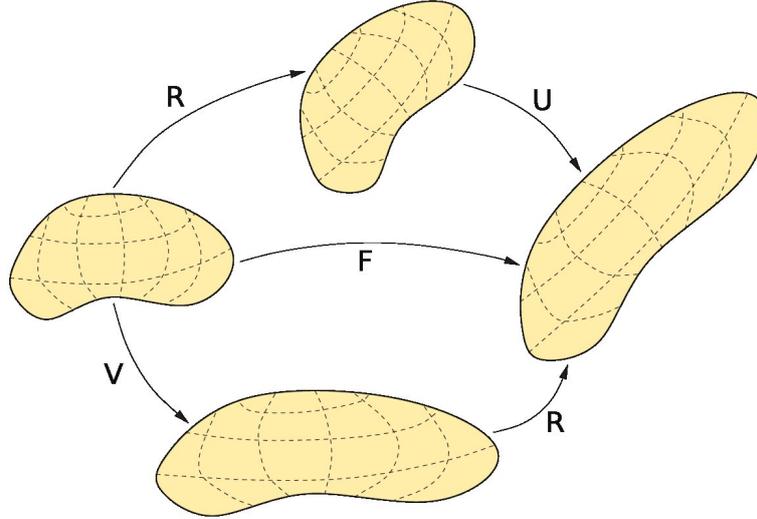
$$(2.2d) \quad J = \det(\mathbf{F})$$

2.3.2.2 Strain tensors

Imposing the standard condition on the deformation gradient $\det(\mathbf{F}) > 0$ ensuring that (2.2a) is a one-to-one mapping, the polar decomposition theorem admits the unique representation of the deformation gradient (3.26) in the form (2.3a), see Figure 2.3, where:

- **U** and **V**: Are positive definite symmetric tensors. These tensors are known as the right and left stretch tensors respectively, and measure local stretching near \mathbf{X} . Since **U** and **V** are symmetric, it follows from the spectral theorem that they admit the spectral decomposition as can be seen in (2.3b).
- **R**: Is an orthogonal tensor, called rotation tensor and measures the local rigid rotation of points close to \mathbf{X} .

In (2.3c) the right and left *Cauchy-Green* (**C** and **b**) tensors are introduced. For the definition of the strain tensors in (2.3d) we can find the definition of a set of strain measures in function of the parameter m . The *GreenLagrange* strain tensor, $\mathbf{E}^{(m)}$, is a family particular member of this (with $m = 2$). Other commonly employed members of this family are the *Biot* ($m = 1$), *Hencky* ($m = 0$) and *Almansi* ($m = -2$) strain tensors. It should be emphasised that, in order to preserve the invariance of the stress work per unit mass, a particular choice of strain measure necessitates usage of the associated dual stress measure within the constitutive relations. When linearised about the reference

Figure 2.3: Polar decomposition of \mathbf{F}

state, all strain measures give the standard small strain tensor.

$$(2.3a) \quad \mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{R}$$

$$(2.3b) \quad \mathbf{U} = \sum_{i=1}^3 \lambda_i \mathbf{l}_i \otimes \mathbf{l}_i, \quad \mathbf{V} = \sum_{i=1}^3 \lambda_i \mathbf{e}_i \otimes \mathbf{e}_i, \quad \mathbf{l}_i = \mathbf{R} \mathbf{e}_i$$

$$(2.3c) \quad \mathbf{C} = \mathbf{F}^T \cdot \mathbf{F} = \mathbf{U}^2, \quad \mathbf{b} = \mathbf{F} \cdot \mathbf{F}^T = \mathbf{V}^2$$

$$(2.3d) \quad \mathbf{E}^{(m)} = \begin{cases} \frac{1}{m} (\mathbf{U}^m - \mathbf{I}) & m \neq 0 \\ \ln(\mathbf{U}) & m = 0 \end{cases}$$

Equivalently, in terms of spectral decomposition can be reformulated as (2.3e).

$$(2.3e) \quad \begin{cases} \mathbf{E}^{(m)} = \sum_{i=1}^3 f(\lambda_i) \mathbf{l}_i \otimes \mathbf{l}_i \\ f(\lambda_i) = \begin{cases} \frac{1}{m} (\lambda_i^m - 1) & m \neq 0 \\ \ln(\lambda_i) & m = 0 \end{cases} \end{cases}$$

We add to the former that all these strain tensors ($\mathbf{E}^{(m)}$) for any strain measure can be transformed between them in considering the **push-forward** (ϕ_*) and **pull-back** (ϕ^*) operations. This operation just requires, in addition to the given strain tensor in a given strain measure, the deformation gradient (\mathbf{F}). For example, using the **push-forward** we can transform the *Green-Lagrange* strain tensor into the *Almansi* strain tensor. With the **pull-back** operation, we can proceed in the opposite direction, from *Almansi* to *Green-Lagrange*. For more information about the procedure, we address to a dedicated Appendix section [A.1.Pull-Back, Push-Forward fundamental concepts](#) where the operation is detailed.

2.4 Finite Element formulation for CSD

2.4.1 Weak formulation in solids

Here we introduce a summarise of the *Galerkin* method[[OnlGal](#)] for the deduction of the weak formulation, using as main reference the work of *Zienkiewicz*[[BookZZT13](#)]. The deduction can be followed in (2.4).

First we divide the domain Ω in subdomains Ω_e (elements), as well as the boundary Γ , (2.4a). In the following we will represent the discrete form of the former continuous expressions as $\hat{\cdot}$.

$$(2.4a) \quad \begin{cases} \Omega \approx \hat{\Omega} = \sum_e \Omega_e \\ \Gamma \approx \hat{\Gamma} = \sum_e \Gamma_e = \sum_{et} \Gamma_{te} (\text{Neumann}) + \sum_{eu} \Gamma_{ue} (\text{Dirichlet}) \end{cases}$$

For the displacement (\mathbf{u}) we consider an approximation employing the shape functions (\mathbf{N}_b) corresponding to the elements considered in our formulation, (2.4b).

$$(2.4b) \quad \mathbf{u}(\xi, t) = \hat{\mathbf{u}} = \sum_b N_b(\xi) \mathbf{u}_b(t) = \mathbf{N}(\xi) \mathbf{u}(t)$$

With the shape forms we can define the derivatives, needed to define the strain matrix (\mathbf{B}), (2.4c). This matrix is necessary in order to formulate the *strain-displacement* equation $\varepsilon = \mathbf{B}\mathbf{u}$.

$$(2.4c) \quad \mathbf{B} = \frac{\partial N_a}{\partial \xi^i} = \frac{\partial x_j}{\partial \xi^i} \frac{\partial N_a}{\partial x_j}, \text{ and in matrix form } \frac{\partial N_a}{\partial \xi} = \mathbf{J} \frac{\partial N_a}{\partial \mathbf{x}}$$

Finally, the weak form of the equilibrium's equation can be expressed as (2.4d). In this case we include the dynamic effects in order to be a proper **CSD**. For that we will consider the respective velocity $\dot{\mathbf{u}}$ and acceleration $\ddot{\mathbf{u}}$. The damping coefficient will be simplified with the parameter c , a better alternative will be shown at 2.4.5. **Damping matrix**. In here the body forces (\mathbf{b}) as well as the external forces ($\bar{\mathbf{i}}$) are taken into consideration.

$$(2.4d) \quad \delta W_e(\mathbf{u}) = \delta \mathbf{u}^T \left[\int_{\Omega_e} \mathbf{N}^T \rho \mathbf{N} d\Omega \ddot{\mathbf{u}} + \int_{\Omega_e} \mathbf{N}^T c \mathbf{N} d\Omega \dot{\mathbf{u}} + \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega_e} \mathbf{N}^T \mathbf{b} d\Omega - \int_{\Gamma} \mathbf{N}^T \bar{\mathbf{i}} d\Gamma \right]$$

That can be expressed in a semi-discrete form from (2.4e). In here we can express the stress divergence or stress force term (\mathbf{P}), the mass matrix (\mathbf{M}), the damping matrix (\mathbf{C}) and the resulting external forces (\mathbf{f}). \mathbf{P} includes the resulting internal forces of the system.

$$(2.4e) \quad \mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{P}(\boldsymbol{\sigma}) = \mathbf{f}$$

Where the former components can be expressed as (2.4f). In case of considering a linear problem, the stress divergence (\mathbf{P}) can be obtained directly with the stiffness matrix (\mathbf{K}) and its displacements (\mathbf{u}).

$$(2.4f) \quad \begin{cases} \mathbf{M} = \sum_e \int_{\Omega_e} \mathbf{N}^T \rho \mathbf{N} d\Omega \\ \mathbf{C} = \sum_e \int_{\Omega_e} \mathbf{N}^T c \mathbf{N} d\Omega \\ \mathbf{P} = \sum_e \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma} d\Omega \\ \text{For linear elasticity } \mathbf{P}^{(e)}(\boldsymbol{\sigma}) = \mathbf{K}^{(e)} \mathbf{u} \\ \mathbf{f} = \sum_e \left(\int_{\Omega_e} \mathbf{N}^T \mathbf{B} d\Omega + \int_{\Gamma_{te}} \mathbf{N}^T \bar{\mathbf{i}} d\Gamma \right) \end{cases}$$

2.4.2 TL and UL formulation

Extending the previous section, here we introduce the two possible formulation for the mechanic problem, the **UL** and **TL** formulation [BookBel+14]. In the **TL** form the only variable which depends on the deformation is the nominal stress (\mathbf{P} , see Table 2.1), that is, it is the only variable which varies with time. In the **UL** form the domain of the element (or

body), the spatial derivatives $\partial N_i/\partial x_i$ and the *Cauchy stress* (σ) depends on the deformation, and hence on time, which complicates the derivation of the tangent stiffness.

$$(2.5a) \quad \begin{cases} \mathbf{f}^{int} = \int_{\Omega_0} \mathbf{B}_0 \mathbf{P} d\Omega_0 \\ \mathbf{K}_{IJ}^{mat} = \int_{\Omega_0} \mathbf{B}_{0I}^T [\mathbb{C}^{SE}] \mathbf{B}_{0J} d\Omega_0 \\ \mathbf{K}_{IJ}^{geo} = \mathbf{I} \int_{\Omega_0} \mathbf{B}_{0I}^T [\mathbf{S}] \mathbf{B}_{0J} d\Omega_0 \end{cases}$$

$$(2.5b) \quad \begin{cases} \mathbf{f}^{int} = \int_{\Omega} \mathbf{B} \sigma d\Omega \\ \mathbf{K}_{IJ}^{mat} = \int_{\Omega} \mathbf{B}_I^T [\mathbb{C}^{\sigma T}] \mathbf{B}_J d\Omega \\ \mathbf{K}_{IJ}^{geo} = \mathbf{I} \int_{\Omega} \mathbf{B}_I^T [\sigma] \mathbf{B}_J d\Omega \end{cases}$$

The **UL** forms (2.5b) are generally easier to use than the **TL** forms (2.5a), since **B** is more easily constructed than **B₀** and many material laws are developed in terms of rates of *Cauchy stress* (σ). The material stiffness in **TL** form can be combined with the geometric stiffness in **UL** form or vice versa. In order to compute the material stiffness matrix (\mathbf{K}^{mat}), in addition to compute **B** for the **TL** and **B₀** for the **UL**, we need to compute the consistent constitutive tensor (\mathbb{C}), \mathbb{C}^{SE} for **TL** and $\mathbb{C}^{\sigma T}$ for **UL**. The geometric stiffness matrix (\mathbf{K}^{geo}) depends on **Second Piola-Kirchhoff (PK2) (S)** on the **TL** frame, but in the **UL** it depends on the *Cauchy stress* (σ). The element **SPRISM** (3.4. Prismatic solid-shell) has been developed in **TL** formulation, but there is an intention to implement in **UL** as well.

The following, Table 2.1, will help to understand the different types stress measures and the possible transformation applying the corresponding considering the **push-forward** (ϕ_*) and **pull-back** (ϕ^*) operations.

Cauchy stress σ	Nominal stress P	Second PK stress S	Corotational Cauchy stress $\hat{\sigma}$
$\sigma =$	$J^{-1} \mathbf{F} \cdot \mathbf{P}$	$J^{-1} \mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T$	$J\mathbf{R} \cdot \hat{\sigma} \cdot \mathbf{R}^T$
$\mathbf{P} = J\mathbf{F}^{-1} \cdot \sigma$		$\mathbf{S} \cdot \mathbf{F}^T$	$J\mathbf{U}^{-1} \cdot \hat{\sigma} \cdot \mathbf{R}^T$
$\mathbf{S} = J\mathbf{F}^{-1} \cdot \sigma \mathbf{F}^{-T}$	$\mathbf{P} \cdot \mathbf{F}^{-T}$		$J\mathbf{U}^{-1} \cdot \hat{\sigma} \cdot \mathbf{U}^{-1}$
$\hat{\sigma} = \mathbf{R}^T \cdot \sigma \cdot \mathbf{R}$	$J^{-1} \mathbf{U} \cdot \mathbf{P} \cdot \mathbf{R}$	$J^{-1} \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{U}$	
$\tau = J\sigma$	$\mathbf{F} \cdot \mathbf{P}$	$\mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T$	$J\mathbf{R} \cdot \hat{\sigma} \cdot \mathbf{R}^T$

Table 2.1: Transformations of stresses. Source[BookBel+14]

2.4.3 Solution of the non-linear-equilibrium equations system

When we solve non-linear problems[BookOll14] we need to consider efficient methods designed for this purpose. This means the resolution of $\Delta \mathbf{f}_i^{t+\Delta t} = -\mathcal{J}_i^{t+\Delta t} \cdot \Delta \mathbf{u}_{i+1}^{t+\Delta t}$, which can be carried basically by two kinds of methods, a linearisation based in the standard/modified **Newton-Raphson (NR)** technique or a *Quasi-Newton* approximation technique of the *Jacobian* matrix (\mathcal{J}). Both methods can be improved with the use of the *convergence accelerators* algorithms. In the following we are going to present the **NR** method, the concept of line search, and the *arc-length* method.

2.4.3.1 Newton-Raphson method

This is the fastest convergence method to solve non-linear equations systems using the linearisation technique. This technique assumes that the solution is convergent in the surroundings of it, being the convergence ratio in this case quadratic. Its solution comes from the inversion of the *Jacobian* operator (\mathcal{J}) from Equation (2.6). In our system of equations \mathcal{J} acts as the **LHS**.

$$(2.6) \quad [\Delta \mathbf{u}]_{i+1}^{t+\Delta t} = -[\mathcal{J}_i^{t+\Delta t}]^{-1} \cdot [\Delta \mathbf{f}]_i^{t+\Delta t}$$

The advantages of the method can be listed as the followings:

- It always needs the tangent operator, which is not always easy to obtain, store and invert (for example in the case of antisymmetric operators)

- The convergence velocity depends significantly on the initial guess solution
- The method can get stuck in a local minima, which could be difficult to leave afterwards

This will be the default strategy followed in our developments and problems. In the contact chapter, it will be shown the strategies followed in order to adapt the NR strategy to the proper resolution of the contact problem.

2.4.3.2 Line-search

Zienkiewicz[BookZTF14] classifies this method in the category of *convergence accelerators*. The approach[BookZTF14] is based in the concept of the descent direction. This descent direction can be obtained applying different procedures, such the gradient descent, NR or *quasi-Newton method*. Independently of the approach, the method will proceed as the following Algorithm 1.

Algorithm 1 Line-search

- 1: **procedure** Line-search
 - 2: Set iteration counter $i = 0$, and make an initial guess \mathbf{x}_0 for the minimum
 - 3: **while** ($rhs_{total} > tolerance_{abs}$ and $ratio > tolerance_{rel}$) and $i < iteration_{max}$ **do**
 - 4: Compute descent (or search) direction $d\mathbf{x}$
 - 5: Choose η_k (step size) to minimize the projection of the residual on the search direction: $G(\eta_k) = f(\mathbf{x}_k + \eta_k d\mathbf{x})$
 - 6: Update $\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k d\mathbf{x}$
 - 7: Update $i = i + 1$ and the RHS
-

This method is discarded, as cannot be used, at least without additional considerations, for CCM. We will then focus on the other two methods. In general, it is complex to consider *convergence accelerators* on contact problems, as the BC of the problem is continuously changing and this is troublesome for this kind of strategy.

2.4.3.3 Arc-length

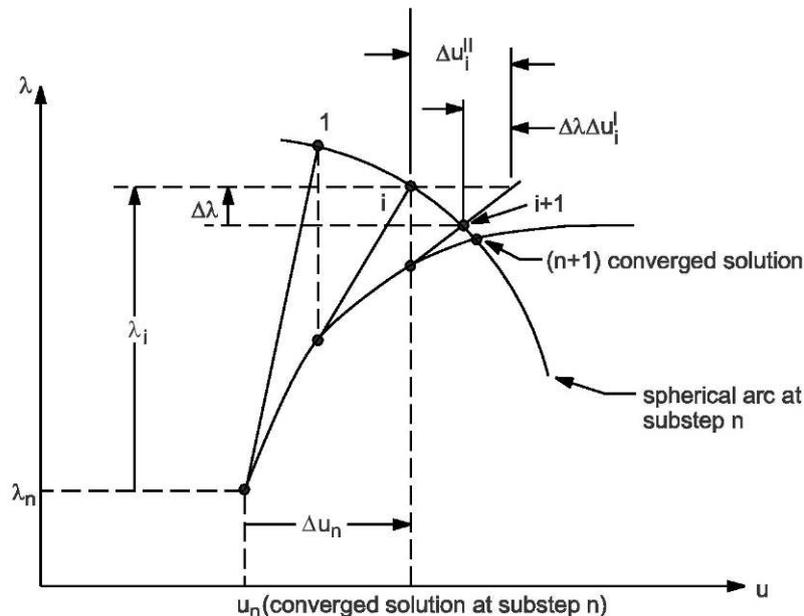


Figure 2.4: Graphic representation of the arc-length method

Unstable behaviours[BookOII14] occur for various problems in structural mechanics, for which it is difficult to find a solution. To avoid this, a system of equilibrium equations with restrictions of type $\Delta \mathbf{f}(\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}, \lambda)$ are usually used, where the magnitude of external force $\lambda \mathbf{f}^{ext}$ is an unknown conditioned by an additional equation $c(\mathbf{u}, \lambda)$. The basic concepts of the method are presented in (2.7) (see Figure 2.4).

Considering the following relations stating from the equilibrium equations (2.7a).

$$(2.7a) \quad \left(\frac{\partial \Delta \mathbf{f}}{\partial \mathbf{u}} \right)_i = \mathcal{J}_i, \quad \left(\frac{\partial \Delta \mathbf{f}}{\partial \lambda} \right)_i = \mathbf{f}^{ext}$$

It allows to rewrite the equilibrium equations as (2.7b).

$$(2.7b) \quad \begin{cases} \mathbf{0} = \Delta \mathbf{f}(\mathbf{u}_i, \lambda_i) + \mathcal{J}_i \cdot \delta \mathbf{u}_{i+1} - \mathbf{f}^{ext} \cdot \delta \lambda_{i+1} \\ \mathbf{0} = \mathbf{c}(\mathbf{u}, \lambda)_{i+1} \end{cases}$$

The displacement increment can be written as (2.7c), where \mathbf{u}^{max} is the total displacement with the last or maximal value of the external force and $\delta \hat{\mathbf{u}}$ is the solution of the system of equations without correction and $\delta \lambda_{i+1}$ is the change of the application factor of the load.

$$(2.7c) \quad \delta \mathbf{u}_{i+1} = \delta \hat{\mathbf{u}}_{i+1} + \mathbf{u}_{i+1}^{max} \delta \lambda_{i+1}$$

$$(2.7d) \quad \begin{cases} \mathcal{J}_i \cdot \delta \hat{\mathbf{u}}_{i+1} = - [\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{f}_i^{int}(\dot{\mathbf{u}}, \mathbf{u}) - \lambda_i \mathbf{f}^{ext}] \\ \mathcal{J}_i \cdot \mathbf{u}^{max} = \mathbf{f}^{ext} \\ \lambda_{i+1} = \lambda_i + \delta \lambda_{i+1} \end{cases}$$

With the basis developed previously (2.7) it is possible to implement different versions of the method. The version implemented in **Kratos** is based in different implementations from the literature[ArtFM93; ArtTL98; ArtCri83].

2.4.4 Time integration schemes

Taking the *semi-discrete* form of the dynamic principle, Equation (2.8), we obtain the equation governing the solid dynamics (**CSD**). We will express in this manner, instead of the proper residual form, $\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{LHS}^{static} \Delta \mathbf{u} = \mathbf{RHS}^{static}$, for being more illustrative showing the resulting structure when applying a certain time integration scheme.

$$(2.8) \quad \mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} + \mathbf{f} = \mathbf{0}$$

We can now choose between the implicit and explicit approach, following[BookOll14] we can define them as the following. If the response at current time depends completely from the solution in the current step then we are working in a explicit solution, but in the other hand if the solution depends on velocity and the acceleration at current time, then we have an implicit solution. Considering the set of equations (2.9), which depend on the α parameter, then we can say:

- **Explicit:** It states equilibrium at time t , with $\alpha = 0$ (Forward *Euler*). The displacement in next step is obtained depending on the velocity and displacement of the previous step. This methodology if simple in terms of structure requires less memory storage, does not require expensive tangent operators, the algorithms obtained are reliable and it requires a very small time increment what it can be numerically expensive and unapproachable.
- **Implicit:** It formulates the equilibrium at time $t + \Delta t$ with $\alpha = 1$ (Backward *Euler*). The displacement in the next step is obtained depending on the current time velocity and on the displacement of the previous step. For this approach the time increments can be much larger preserving stability, allows more precise solutions, but requires the computation of the tangent operators (**LHS**) and large storage demand.

$$(2.9) \quad \begin{cases} \mathbf{u}_{t+\Delta t} = \mathbf{u}_t + \Delta t \dot{\mathbf{u}}_{t+\alpha \Delta t} \\ \dot{\mathbf{u}}_{t+\Delta t} = (1 - \alpha) \dot{\mathbf{u}}_t + \alpha \dot{\mathbf{u}}_{t+\Delta t} \end{cases}$$

As we have said previously, the aim of this work will focus in an implicit solution; thus we need to discretise (2.8) with some temporal scheme, the ones that we present corresponds with the most extended in structural dynamics, the **Newmark- β** (2.4.4.1. **Newmark time scheme**) scheme, the *Bossak* scheme (2.4.4.2. **Bossak algorithm**) and the **α -generalised** (2.4.4.3. **α -generalised**). For additional literature related with structural dynamics, and in particular **CSD**, we recommend to consult the bibliography [BookCP12; BookOll14; BookRR12], where alternative schemes are explained, and the stability and convergence of each one is deduced.

2.4.4.1 Newmark time scheme

Below with one step method employing *Newmark* schema we obtain (2.10).

$$(2.10) \quad \begin{cases} (c_3 \mathbf{M} + c_2 \mathbf{C} + c_1 \mathbf{K}) \mathbf{u}^{n+1} + \mathbf{f}^{n+1} = \mathbf{0} \\ \mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \dot{\mathbf{u}}^n + \left(\frac{1}{2} - \beta\right) \Delta t^2 \ddot{\mathbf{u}}^n + \beta \Delta t^2 \ddot{\mathbf{u}}^{n+1} \\ \dot{\mathbf{u}}^{n+1} = \dot{\mathbf{u}}^n + (1 - \gamma) \Delta t \ddot{\mathbf{u}}^n + \gamma \Delta t \ddot{\mathbf{u}}^{n+1} \end{cases}$$

Where the coefficients of the scheme are $c_1 = 1$, $c_2 = \frac{\gamma}{\beta \Delta t}$ and $c_3 = \frac{1}{\beta \Delta t^2}$.

2.4.4.2 Bossak algorithm

The *Kratos* framework uses the *Bossak* [ArtWBZ80] algorithm by default for the time integration. This method is based in the *Newmark* method, introducing additional parameters to dissipate spurious oscillations in the high frequencies. The parameters introduced modify the form of the equilibrium equation including an averaging process between the current and the old-time station so to preserve the consistency requirements. The only variation respect *Newmark* is in the equilibrium equation (2.11).

$$(2.11) \quad \mathbf{M} \ddot{\mathbf{u}}^{n+1-\alpha_B} + \mathbf{C} \dot{\mathbf{u}}^{n+1} + \mathbf{K} \mathbf{u}^{n+1} + \mathbf{f} = \mathbf{0}$$

Where $\frac{-1}{3} < \alpha_B < 0$, considering the *Newmark* parameters defined as $\gamma = \frac{1}{2} - \alpha_B$ and $\beta = \frac{(1-\alpha_B)^2}{4}$. If $\alpha_B = 0$ we obtain the *Newmark* scheme unconditionally stable. The *Bossak* method present some implementation advantages for non-linear problems, owing to that the mass matrix is constant on *Lagrangian* meshes.

2.4.4.3 α -generalised

The **α -generalised** (Chung [ArtCH93]) is the generalisation of the time integration schemes, being the *Newmark* and *Bossak* particular versions of him. In this case the method involves a modification of the equilibrium equation, with the α_f the damping and elastic forces are taken into a linear combination and with α_m the inertial forces are modified. With $\alpha_m = \alpha_f = 0$ the method obtained corresponds with *Newmark*. The expression of the method is presented in (2.12).

The equations from (2.10) corresponding to \mathbf{u}^{n+1} and $\dot{\mathbf{u}}^{n+1}$ remain exactly the same, except that the value the *Newmark* constants which are $\gamma = \frac{1}{2} + \alpha_m - \alpha_f$ and $\beta \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m)$ for an unconditionally stable solution.

$$(2.12a) \quad \mathbf{M} \ddot{\mathbf{u}}^{n+1-\alpha_m} + \mathbf{C} \dot{\mathbf{u}}^{n+1-\alpha_f} + \mathbf{K} \mathbf{u}^{n+1-\alpha_f} + \mathbf{f} = \mathbf{0}$$

Where each component of the displacement and its derivatives is calculated as:

$$(2.12b) \quad \begin{cases} \ddot{\mathbf{u}}^{n+1-\alpha_m} = (1 - \alpha_f) \ddot{\mathbf{u}}^{n+1} + \alpha_f \ddot{\mathbf{u}}^n \\ \dot{\mathbf{u}}^{n+1-\alpha_f} = (1 - \alpha_m) \dot{\mathbf{u}}^{n+1} + \alpha_m \dot{\mathbf{u}}^n \\ \mathbf{u}^{n+1-\alpha_f} = (1 - \alpha_m) \mathbf{u}^{n+1} + \alpha_m \mathbf{u}^n \end{cases}$$

2.4.4.4 BDF

The **BDF** is a family of implicit methods for numerical integration of **ODE**, originally proposed by *Curtiss*[ArtCH52]. This family of methods is commonly used in *stiff*³ **ODE**. Despite the method can be applied to any type of **ODE**, if applied directly to the *Newton's* second laws we can obtain the following (2.13) for the **BDF1** (backward *Euler* method) and **BDF2** schemes.

The general formula of the **BDF**, for a variable depending on time, can be written as:

$$(2.13a) \quad \begin{aligned} \frac{dy}{dt} &= f(t, y), \quad y(t_0) = y_0 \\ \sum_{k=0}^s a_k y_{n+k} &= \Delta t \beta f(t_{n+s}, y_{n+s}) \end{aligned}$$

In here s corresponds with the desired order of the scheme, and Δt with the time increment. The a_k and β parameters are obtained via *Lagrange interpolation polynomials* in order to achieve $\mathcal{O}(h)^s$. The **BDF1** and **BDF2** coefficients correspond with:

$$(2.13b) \quad \begin{aligned} \text{BDF1: } y_{n+1} - y_n &= \Delta t f(t_{n+1}, y_{n+1}) \\ \text{BDF2: } y_{n+2} - \frac{4}{3}y_{n+1} + \frac{1}{3}y_n &= \frac{2}{3}\Delta t f(t_{n+2}, y_{n+2}) \end{aligned}$$

Applying this to *Newton's* second laws, we obtain and to our system of equations (**LHS** and **RHS**). The definition of \mathbf{u}^{n+1} depends on the **BC** of the problem, and can be deduced from (2.13b) or by means of a second-order approximation like $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \dot{\mathbf{u}}^n + \frac{\Delta t^2}{2} \ddot{\mathbf{u}}^n$.

$$(2.13c) \quad \frac{dy^{n+1}}{dt} = \sum_{i=0}^s bdf_i y^{n+1-i}$$

$$(2.13d) \quad \begin{aligned} \text{BDF1: } \begin{cases} \ddot{\mathbf{u}}^{n+1} = \frac{1}{\Delta t} \dot{\mathbf{u}}^{n+1} - \frac{1}{\Delta t} \dot{\mathbf{u}}^n \\ \dot{\mathbf{u}}^{n+1} = \frac{1}{\Delta t} \mathbf{u}^{n+1} - \frac{1}{\Delta t} \mathbf{u}^n \end{cases} & \quad \text{BDF2: } \begin{cases} \ddot{\mathbf{u}}^{n+1} = \frac{3}{2\Delta t} \dot{\mathbf{u}}^{n+1} - \frac{2}{\Delta t} \dot{\mathbf{u}}^n + \frac{1}{2\Delta t} \dot{\mathbf{u}}^{n-1} \\ \dot{\mathbf{u}}^{n+1} = \frac{3}{2\Delta t} \mathbf{u}^{n+1} - \frac{2}{\Delta t} \mathbf{u}^n + \frac{1}{2\Delta t} \mathbf{u}^{n-1} \end{cases} \end{aligned}$$

$$(2.13e) \quad \begin{aligned} \text{BDF1: } \begin{cases} \text{LHS} = \mathbf{K} + \frac{1}{\Delta t^2} \mathbf{M} + \frac{1}{\Delta t} \mathbf{C} \\ \text{RHS} = \mathbf{f} - \mathbf{M}\ddot{\mathbf{u}} - \mathbf{C}\dot{\mathbf{u}} \end{cases} & \quad \text{BDF2: } \begin{cases} \text{LHS} = \mathbf{K} + \frac{9}{4\Delta t^2} \mathbf{M} + \frac{3}{2\Delta t} \mathbf{C} \\ \text{RHS} = \mathbf{f} - \mathbf{M}\ddot{\mathbf{u}} - \mathbf{C}\dot{\mathbf{u}} \end{cases} \end{aligned}$$

In addition, in case an adaptive time step (Δt) is considered we need to update the **BDF** coefficients. If considering $\rho = \frac{\Delta t_n}{\Delta t_{n+1}}$, then:

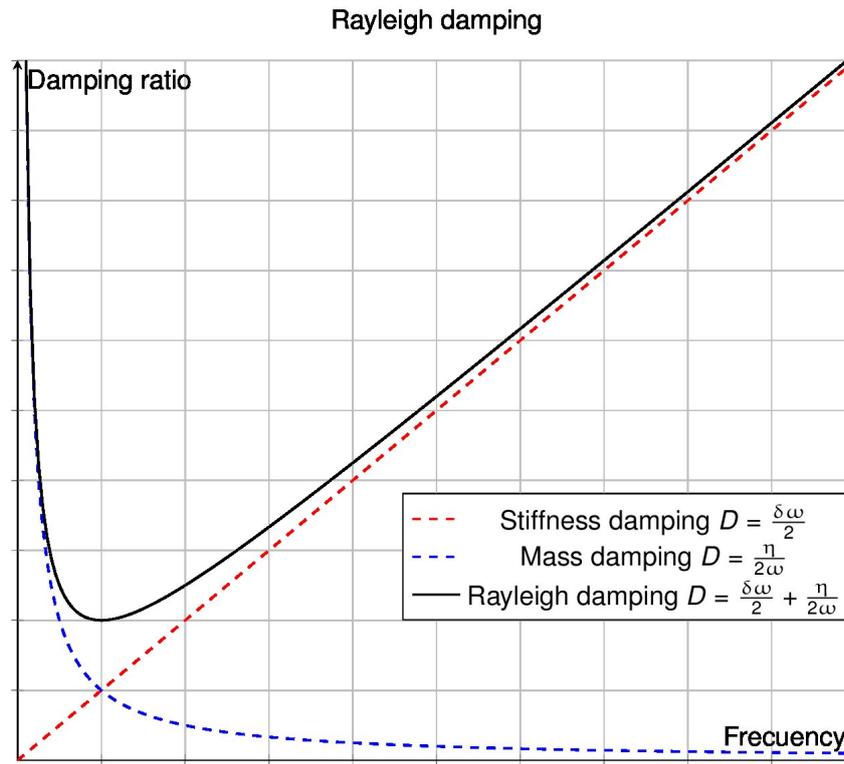
$$(2.13f) \quad \begin{aligned} \text{BDF1: } bdf_0 &= \frac{\rho}{\Delta t}, \quad bdf_1 = -\frac{\rho}{\Delta t} \\ \text{BDF2: } t_{coeff} &= \frac{1}{\Delta t + \Delta t \rho}, \quad bdf_0 = t_{coeff}(\rho^2 + 2\rho), \quad bdf_1 = -t_{coeff}(\rho^2 + 2\rho + 1), \quad bdf_2 = t_{coeff} \end{aligned}$$

2.4.5 Damping matrix

The damping matrix presented in (2.4e) depends on the damping coefficient (c) of the given element e , which is not always available. Usually a critical-damping ratio respect the critical is considered (ξ). With this critical-damping ratio we can estimate the damping matrix. In order to do so we will consider the classical *Lord Rayleigh*[BookRay77] damping. In this formulation, the damping matrix (**C**) is assumed to be proportional to the mass (**M**) and stiffness (**K**) matrices as in Equation (2.14a).

$$(2.14a) \quad \mathbf{C} = \eta \mathbf{M} + \delta \mathbf{K}$$

³**ODE** unstable unless a small time increment is taken into account.

Figure 2.5: Graphic representation of *Rayleigh* damping matrix

In here η is the mass-proportional damping coefficient and δ are the stiffness-proportional damping coefficient. Taking into consideration that the properties of the modal equations and the orthogonality conditions, we can express (2.14a) as (2.14b).

$$(2.14b) \quad \xi_n = \frac{1}{2\omega_n}\eta + \frac{\omega_n}{2}\delta$$

In this expression we have the relationship between the n critical-damping ratio and the n natural frequency. Considering two different natural frequencies, we can define a system of equations such as:

$$(2.14c) \quad \begin{bmatrix} \xi_i \\ \xi_j \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{1}{\omega_i} & \omega_i \\ \frac{1}{\omega_j} & \omega_j \end{bmatrix} \begin{bmatrix} \eta \\ \delta \end{bmatrix}$$

If damping for both frequencies is set to be equal, then the conditions associated with the proportionality factors can be simplified as:

$$(2.14d) \quad \xi_i = \xi_j = \xi \quad \text{therefore} \quad \delta = \frac{2\xi}{\omega_i + \omega_j} \quad \text{and} \quad \eta = \omega_i\omega_j\delta$$

Bibliography

Books

- [BookBel+14] *Nonlinear Finite Elements for Continua and Structures*. Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. Wiley. 2nd ed. 2014.
- [BookBHS07] *Encyclopedia of Computational Mechanics (3 Volume Set)*. René de Borst, T.J.R. Hughes, and Erwin Stein. Wiley. 1st ed. 2007.
- [BookBor+12] *Nonlinear Finite Element Analysis of Solids and Structures, 2nd edition*. René de Borst, Mike A. Crisfield, Joris J. C. Remmers, and Clemens V. Verhoosel. Wiley. 2nd Edition. 2012.
- [BookCP12] *Dynamics of Structures*. Ray W. Clough and Joseph Penzien. McGraw-Hill. 3rd ed. 2012. P. 752.
- [BookFin13] *The method of weighted residuals and variational principles*. Bruce A Finlayson. Siam. 2013.
- [BookLL92] *Numerical methods for conservation laws*. Randall J LeVeque and Randall J LeVeque. Springer. 1992.
- [BookNPO09] *Computational Methods for Plasticity Theory and Applications*. E. A. de Souza Neto, D. Periaé, and D.R.J. Owen. Wiley. 2009.
- [BookOll14] *Nonlinear Dynamics of Structures*. Sergio Oller. Springer International Publishing. 2014.
- [BookRay77] *Theory of Sound*. Lord Rayleigh. 1877. P. 93.
- [BookRR12] *Elements of Structural Dynamics: A New Perspective*. G. Visweswara Rao and Debasish Roy. 2012.
- [BookWri08] *Nonlinear finite element methods*. Peter Wriggers. Springer. 1st ed. 2008.
- [BookZTF14] *The Finite Element Method for Solid and Structural Mechanics*. O. C. Zienkiewicz, Robert L. Taylor, and David Fox. Butterworth-Heinemann. 7th ed. 2014.
- [BookZZT13] *The Finite Element Method: its Basis and Fundamentals*. O. C. Zienkiewicz, J.Z. Zhu, and Robert L. Taylor. Butterworth-Heinemann. 7th ed. 2013.

Articles

- [ArtCGP06] “High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow-water systems”. Manuel Castro, José Gallardo, and Carlos Parés. In: *Mathematics of computation*. No. 255, Vol. 75, 2006, pp. 1103–1134. DOI: [10.1090/s0025-5718-06-01851-5](https://doi.org/10.1090/s0025-5718-06-01851-5).
- [ArtCH52] “Integration of stiff equations”. CF Curtiss and Joseph O Hirschfelder. In: *Proceedings of the National Academy of Sciences of the United States of America*. No. 3, Vol. 38, 1952, p. 235. National Academy of Sciences. DOI: [10.1073/pnas.38.3.235](https://doi.org/10.1073/pnas.38.3.235).
- [ArtCH93] “A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method”. Jintai Chung and GM Hulbert. In: *Journal of applied mechanics*. No. 2, Vol. 60, 1993, pp. 371–375. American Society of Mechanical Engineers. DOI: [10.1115/1.2900803](https://doi.org/10.1115/1.2900803).
- [ArtClo60] “The finite element method in plane stress analysis”. Ray W Clough. In: *Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh Pa., Sept. 8 and 9, 1960*. 1960,

- [ArtCri83] “An arc-length method including line searches and accelerations”. M. A. Crisfield. In: *International Journal for Numerical Methods in Engineering*. No. 9, Vol. 19, 1983, pp. 1269–1289. John Wiley & Sons, Ltd. DOI: [10.1002/nme.1620190902](https://doi.org/10.1002/nme.1620190902).
- [ArtFM93] “Geometrical interpretation of the arc-length method”. M. Fafard and B. Massicotte. In: *Computers & Structures*. No. 4, Vol. 46, 1993, pp. 603–615. DOI: [10.1016/0045-7949\(93\)90389-u](https://doi.org/10.1016/0045-7949(93)90389-u).
- [ArtGal15] “Series solution of some problems of elastic equilibrium of rods and plates”. Boris Grigoryevich Galerkin. In: *Vestn. Inzh. Tekh.* Vol. 19, 1915, pp. 897–908.
- [ArtPom69] “K. Washizu, Variational Methods in Elasticity and Plasticity.(International Series of Monographs in Aeronautics and Astronautics). X+ 348 S. m. Fig. Oxford/London/Edinburgh/New York/Toronto/Sydney/Paris/Braunschweig 1968. Preis geb. 120 s. net”. W Pompe. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*. No. 5, Vol. 49, 1969, pp. 319–319. Wiley Online Library. DOI: [10.1002/zamm.19690490535](https://doi.org/10.1002/zamm.19690490535).
- [ArtRic11] “IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam”. Lewis Fry Richardson. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*. No. 459-470, Vol. 210, 1911, pp. 307–357. The Royal Society London.
- [ArtTL98] “A user-controlled arc-length method for convergence to predefined deformation states”. J. G. Teng and Y. F. Luo. In: *Communications in Numerical Methods in Engineering*. No. 1, Vol. 14, 1998, pp. 51–58. John Wiley & Sons, Ltd. DOI: [10.1002/\(sici\)1099-0887\(199801\)14:1<51::aid-cnml130>3.0.co;2-l](https://doi.org/10.1002/(sici)1099-0887(199801)14:1<51::aid-cnml130>3.0.co;2-l).
- [ArtTur56] “Stiffness and deflection analysis of complex structures”. MJ Turner. In: *Journal of the Aeronautical Sciences*. No. 9, Vol. 23, 1956, pp. 805–823. DOI: [10.2514/8.3664](https://doi.org/10.2514/8.3664).
- [ArtWBZ80] “An alpha modification of Newmark’s method”. WL Wood, M Bossak, and OC Zienkiewicz. In: *International Journal for Numerical Methods in Engineering*. No. 10, Vol. 15, 1980, pp. 1562–1566. Wiley Online Library. DOI: [10.1002/nme.1620151011](https://doi.org/10.1002/nme.1620151011).

Online resources

- [OnlFin] *Wikipedia. Finite strain theory*. URL: http://en.wikipedia.org/wiki/Finite_strain_theory.
- [OnlGal] *Wikipedia. Galerkin method*. URL: http://en.wikipedia.org/wiki/Galerkin_method.
- [OnlWol] *Mathematica, Version 11.3. Wolfram Research Inc.*

Chapter 3

Rotation-free shells and solid-shell elements

“ Things should be described as simply as possible, but no simpler. ”

Albert Einstein
(1879 - 1955 AD, German-born theoretical physicist)

3.1 Introduction

In first place we will introduce a simple historical outline in the shell theory. This will be extended in the state-of-the-art section (3.2.State of the Art in shell formulation), the current state of the standard shell theory. The theories relative to rotation-free formulation will be introduced in its respective sections.

In the following sections, we will introduce the element solid-shell developed to tackle the main aim of this work, the prismatic solid-shell element. This element has some resemblance with a previous work, the **Enhanced Basic Shell Triangle (EBST)** element, a rotation-free shell, which considers the neighbour elements with the objective of enrich quadratically the in-plane behaviour.

For that reason we will introduce in first place the rotation-free elements (3.3.Rotation-free shells), and later on we will present the prismatic solid-shell element (3.4.Prismatic solid-shell). The examples of performance of the prismatic solid-shell element can be found in the section concluding this chapter to this one, 3.5.Numerical examples.

3.1.1 Historical outline

The base of the shell theories can be found in the plate theory. This theory takes a similar approach to the beam theory applied in *thin* geometries. The classic thin plate theory establishes that the normal remains straight and orthogonal to the middle plane after deformation[BookOña13]. This theory is based in the postulates presented in the original work of *Kirchoff* in 1850[ArtKir50], despite some people associate this theory to some previous works by *Marie Sophie Germain*[BookBD80; BookBD12] in 1811. We have some analytical solutions for this theory for rectangular plates, as the one presented by *Navier* in 1820. In here *Naviernavier1820memoire* introduced a simple method for finding the displacement and stress when a plate is simply supported. The idea was to express the applied load in terms of *Fourier* components, find the solution for a sinusoidal load (a single *Fourier* component), and then superimpose the *Fourier* components to get the solution for an arbitrary load. Another analytical solution was proposed by *Lévy*[ArtLev99] in 1899, where the main difference with the *Navier* solution consists in start with an

assumed form of the displacement and try to fit the parameters so that the governing equation and the boundary conditions are satisfied.

An evolution to this theory, which can be applied to thick plates, is the *thick* plate theory proposed by *Reissner*[ArtRei45] and *Mindlin*[ArtMin51]. This theory assumes that normals remain straight, though not necessarily orthogonal to the middle plane after deformation. These two theories are complementary, and it is relevant to know when to choose between them depending on the relative thickness of the element (usually we consider as thin elements when $\frac{\text{thickness}}{\text{average side}} \leq \frac{1}{10}$).

The shell theory will be therefore an extension of the previous postulates, but applied to non-planar surfaces[BookOña13]. The non-coplanarity introduces membrane behaviour (axial force) in addition to the plate flexural resistance (bending and shear forces). The governing equations of a curved shell (equilibrium and kinematic equations, etc.) are more complex[BookKra67; ArtNio85; ArtNoo90a; BookTW59; ArtNoo90b] than in the case of plates, mainly due to the curvature of the middle surface. This theory is introduced in first place by *Love*[ArtLov88] in 1888, applied only for *thin* shells. This short introduction to the history of shell theory will be expanded in the shell state-of-the-art section at 3.2.State of the Art in shell formulation, where some of these standard shell models will be presented.

3.2 State of the Art in shell formulation

3.2.1 Element requisites and stabilisation techniques

As it is presented in the book of *Wriggers*[BookWri08], but also in the work of *Belytschko*[BookBel+14], the target properties of any new element, extensible to shell elements are the following¹:

- a) Locking free behaviour for incompressible materials
- b) Good bending performance
- c) No locking in thin elements
- d) No sensitivity against mesh distortions
- e) Good coarse mesh accuracy,
- f) Simple implementation of nonlinear constitutive equations
- g) Efficiency (particularly relevant in explicit simulations)

In order to achieve this objective there are different techniques[BookWri08], here we present a short summarises of the most relevant:

1. **Reduced integration and stabilisation:** Consist in the consideration of reduced number of integration points, with the corresponding reduction in the computational cost and memory, especially in explicit algorithms. This methodology was developed in order to avoid *locking* in case of incompressibility. The principle from which derives this technique is the fact that reduced integration is associated with rank deficiency of the tangent matrices, which is cured with the consideration of stabilisation techniques. Because of this these elements need the choice of artificial stabilisation parameters. Its main advantages are:
 - Locking free for incompressibility
 - Good coarse mesh accuracy
 - Not sensitive against mesh distortions
 - Can be used for arbitrary constitutive equations
2. **Hybrid or mixed variational principles:** These elements include additional **Degree Of Freedom (DOF)** in order to stabilise the formulation. The main disadvantage is the requirement of adapting the constitutive relationships.

¹As we will see the element developed in this work, 3.4.Prismatic solid-shell, enjoy these advantages.

3. **Enhanced strain elements based on the *Hu-Washizu* principle:** Within the enhanced strain formulations, nonconforming strain measures are introduced within the *Hu-Washizu* principle[ArtWas82]. The main elements developed concerning this principle are the ones present in the work of *Simo*[ArtSF89; ArtSFR89; ArtSFR90; ArtSRF90; ArtSK92; ArtSRF92; ArtSim93]. These elements fulfil all points of the previously mentioned requirement, therefore they are well suited for all applications. Additionally these methods have been proven to be efficient in order to overcome the *hour glassing* effect. This method includes techniques such as **EAS** and **Assumed Normal Strain (ANS)**.
4. **Mixed variational principles for problems with rotational DOF:** In here the momentum and moment of momentum are weakly enforced. This leads to the symmetry of the stress tensor and therefore the rotational **DOF** can be introduced as independent field variables.
5. **Composite or macro elements:** Here elements are constructed from subelements, which use simplified or special shape functions.
6. **Higher order displacement elements:** This technique consists in the consideration of higher order geometries as quadratic or cubic, or even hierarchical shape functions using polynomials or **Non-Uniform Rational B-spline (NURB)**. In fact, this last option allows a more efficient formulation, being competitive with the other approaches.

In *Wriggers*[BookWri08], some other techniques are mentioned, such as *nodal based elements* or *Cosserat point elements*.

3.2.2 Shell formulations

We can differentiate three different approaches[BookOña13] for the definition of a shell theory:

1. **Classical shell theories:** These formulations can be found summarised in several sources[BookNov59; ArtNio85; BookWem82; BookGou12; BookCal89]. We can mention particularly the *Discrete-Kirchhoff* assumptions, as an extensively used methodology[ArtWOK68; ArtWem69; ArtDha69; ArtDha70; ArtDMM86; ArtMG86].
2. **3D elements of small thickness:** This one it is not practical as it requires the consideration of very small elements in order to avoid the ill-conditioning coming from a slender aspect ratio of the shell geometry.
3. **Shell elements by degeneration of solid elements:** These formulations, *aka Degenerated Shell (DS)*, comes from the concept of enforcing the shell theory assumptions into the general 3D elasticity[ArtAIZ70]. This denomination can be misleading and it is often used in order to differentiate the formulation from curved shell elements derived from standard shell theory. This kind of elements have enjoyed a lot of popularity, and several authors have significant contributions in this formulation[BookBD90; ArtPar79; ArtKan79; ArtHL81; ArtFV82]. These kinds of elements, as a generalisation of the *ReissnerMindlin* shell theory suffer of membrane locking and therefore require of some kind of stabilisation as the ones presented in 3.2.1. **Element requisites and stabilisation techniques**, particularly reduced integration or assumed strain fields. The main difference between the **DS** and the standard formulation is that in the case of the **DS** the **Principle of Virtual Work (PVW)** is written in terms of stresses and strains, as in solid elements, and the integrals are computed in the volume.
 - **Continuum-Based Resultant (CBR) shell theory:** This theory introduces simplifications in **DS** elements, so that the **PVW** can be expressed in terms of resultant stresses and generalised strains, therefore the integrals can be computed over the shell surface only. It exists a variation of this formulation called **CBR-S**, whereas an assumption we eliminate the *z-dependence* in the surface *Jacobian* (*J*).

Buechter and *Ramm*[ArtBR92] have compared between degenerated shell elements and those based on standard *Reissner-Mindlin/Naghdi* type shell theory. In this work, they highlight the standard shell for practical purposes as their formulation is generally simpler.

In addition, it is relevant to highlight the so-called . As mentioned in the stabilisation techniques in 3.2.1. **Element requisites and stabilisation techniques**, this methodology consists in the consideration of the same basis functions for representing the geometry in **CAD**, like the **NURB**[BookHug+14; BookFHK02; BookPT97; BookRog00]. Thanks to this, the **FE** analysis works on a geometrically exact model and no meshing is necessary[BookHBC09; ArtBen+10; ArtHCB05].

3.3 Rotation-free shells

3.3.1 Introduction

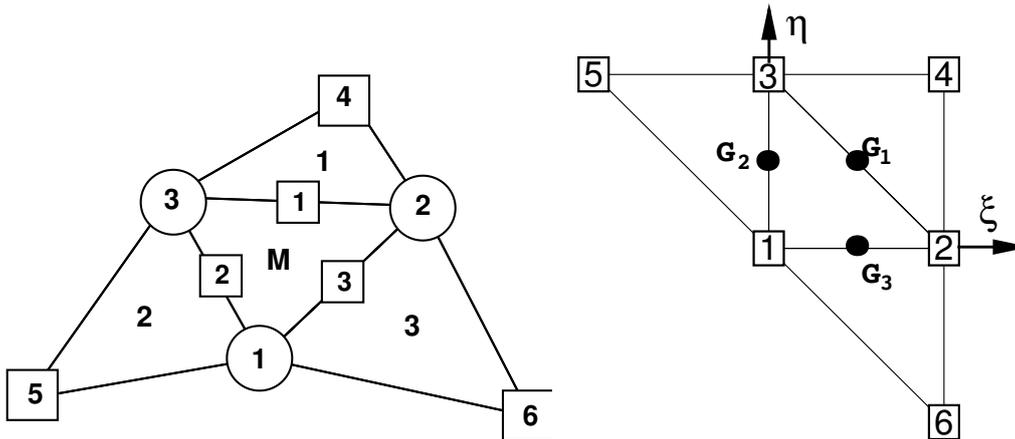
The next section introduces the rotation-free plates and shells developed in the last years by *Eugenio Oñate et al.*[ArtOF05; ArtFO05; ArtFO01; ArtOZ00; ArtFO11]; that despite the modifications and improvements realised in all this time, the principle is the same, a plate/shell that **just consider displacement** as degree of freedom in the nodes.

The traditional *Kirchoff* theory is enough to describe the most common cases of plates and shells, but in the cases where a *thick shells* approach is interesting to be considered, some difficulties appear associated to the C^{12} continuity, and then just cases which need C^0 continuity have been used; in this sense the rotation free theories have been presented as a very interesting alternative.

3.3.2 Basic Shell Triangle

3.3.2.1 Basics

Let us consider a patch of four three node triangles from Figure 3.1a, where the main triangle corresponds with the one with the nodes marked with circles, and the external ones the marked with squares. To develop the formulation we assume a constant curvature field, then the deflection is interpolated linearly in a standard finite element manner as (3.1a), and then calling the curvatures as (3.1b).



(a) Patch of three nodes triangular elements including the central triangle (M) and three adjacent triangles (1, 2 and 3)

(b) Patch of elements in the isoparametric space

Figure 3.1: Patch of elements for **Basic Shell Triangle (BST)** and **EBST**

$$(3.1a) \quad w = \sum_{i=1}^3 L_i^e w_i^e$$

Where L_i^e are the linear shape functions and w_i^e the nodal deflection of each node.

$$(3.1b) \quad \kappa = \begin{Bmatrix} \kappa_{xx} \\ \kappa_{yy} \\ \kappa_{xy} \end{Bmatrix} = \hat{\kappa}$$

Where κ is the curvature vector and $\hat{\kappa}$ is the constant curvature field defined in (3.1c).

$$(3.1c) \quad \hat{\kappa} = \frac{1}{A_M} \int \int_{A_M} \left[-\frac{\partial^2 w}{\partial x^2}, -\frac{\partial^2 w}{\partial y^2}, -2\frac{\partial^2 w}{\partial x \partial y} \right]^T dA$$

Where A_M is the area of the central triangle.

²The function f is said to be of (differentiability) class C^k if the derivatives f' , f'' , ..., $f^{(k)}$ exist and are continuous (the continuity is implied by differentiability for all the derivatives except for $f^{(k)}$).

3.3.2.2 Shell kinematics

Considering a shell of the initial thickness h_0 and with an undeformable middle surface in Ω_0 (\mathbb{R}^3) with a boundary Γ_0 . The positions \mathbf{X} and \mathbf{x} of a point in the undeformed and the deformed configurations can be written as a function of the coordinates of the middle surface φ and the normal \mathbf{t}_3 at the point as (3.2a), where ξ and η are the curvilinear principal coordinates and ζ is the distance from the point to the middle surface in the undeformed configuration.

$$(3.2a) \quad \begin{cases} \mathbf{X}(\xi, \eta, \zeta) = \varphi^0(\xi, \eta) + \lambda \mathbf{t}_3^0 \\ \mathbf{x}(\xi, \eta, \zeta) = \varphi(\xi, \eta) + \zeta \lambda \mathbf{t}_3 \end{cases}$$

Defining λ as the parameter that relates the thickness at the present and initial configuration (3.2b).

$$(3.2b) \quad \lambda = \frac{h}{h^0}$$

A convective system is computed at each point as (3.3).

$$(3.3) \quad \begin{cases} \mathbf{g}_i(\zeta) = \frac{\partial \mathbf{x}}{\partial \xi_i} \quad i = 1, 2, 3 \\ \mathbf{g}_\alpha(\zeta) = \frac{\partial(\varphi(\xi, \eta) + \zeta \lambda \mathbf{t}_3)}{\partial \xi_\alpha} = \varphi'_{,\alpha} + \zeta (\lambda \mathbf{t}_3)'_{,\alpha} \quad \alpha = 1, 2 \\ \mathbf{g}_3(\zeta) = \frac{\partial(\varphi(\xi, \eta) + \zeta \lambda \mathbf{t}_3)}{\partial \zeta} = \lambda \mathbf{t}_3 \end{cases}$$

The curvatures of the middle surface are obtained by (3.4a).

$$(3.4a) \quad \kappa_{\alpha\beta} = \frac{1}{2} (\varphi'_{,\alpha} \cdot \mathbf{t}_{3',\beta} + \varphi'_{,\beta} \cdot \mathbf{t}_{3',\alpha}) = -\mathbf{t}_3 \cdot \varphi'_{,\alpha\beta} \quad \alpha, \beta = 1, 2$$

The deformation gradient tensor (\mathbf{F}) is (3.4b).

$$(3.4b) \quad \mathbf{F} = [\mathbf{x}'_{,1}, \mathbf{x}'_{,2}, \mathbf{x}'_{,3}] = [\varphi'_{,1} + \zeta (\lambda \mathbf{t}_3)'_{,1} \quad \varphi'_{,2} + \zeta (\lambda \mathbf{t}_3)'_{,2} \quad \lambda \mathbf{t}_3]$$

The product $\mathbf{F}^T \mathbf{F} = \mathbf{U}^2 = \mathbf{C}$ (where \mathbf{U} is the right stretch tensor, and \mathbf{C} the right *Cauchy-Green* deformation tensor) can be written as (3.4c).

$$(3.4c) \quad \mathbf{U}^2 = \begin{bmatrix} a_{11} + 2\kappa_{11}\zeta\lambda & a_{12} + 2\kappa_{12}\zeta\lambda & 0 \\ a_{12} + 2\kappa_{12}\zeta\lambda & a_{22} + 2\kappa_{22}\zeta\lambda & 0 \\ 0 & 0 & \lambda^2 \end{bmatrix} \rightarrow \text{if } \kappa_{ij}^0 = 0 \rightarrow \begin{bmatrix} a_{11} + 2\chi_{11}\zeta\lambda & a_{12} + 2\chi_{12}\zeta\lambda & 0 \\ a_{12} + 2\chi_{12}\zeta\lambda & a_{22} + 2\chi_{22}\zeta\lambda & 0 \\ 0 & 0 & \lambda^2 \end{bmatrix}$$

It is also useful to compute the eigendecomposition of \mathbf{U} as (3.4d), where λ_α and \mathbf{r}_α are the eigenvalues and eigenvectors of \mathbf{U} .

$$(3.4d) \quad \mathbf{U} = \sum_{\alpha=1}^3 \lambda_\alpha \mathbf{r}_\alpha \otimes \mathbf{r}_\alpha$$

3.3.2.3 Constitutive models

For the constitutive model, we recommend to consult directly the literature [ArtOF05; ArtFO05; ArtFO01; ArtOZ00; ArtFO11]. It is possible to consider more than one constitutive model, and the ones presented in the references are focused in cases where the objective is to calculate large strains of metal forming processes.

3.3.3 Enhanced Basic Shell Triangle

The following model corresponds with the *enhanced* version of the **BST**, called **EBST**, being the main features of this element:

- The geometry of the patch formed by the central element and the three adjacent elements is quadratically interpolated from the position of the six nodes in the patch.

- Interpolated from the position of the six nodes in the patch. The membrane strains are assumed to vary linearly within the central triangle and are expressed in terms of the (continuous) values of the deformation gradient at the mid-side points of the triangle.
- The assumed constant curvature field within the central triangle is obtained using now twice the values of the (continuous) deformation gradient at the mid-side points.

3.3.3.1 Total Lagrangian formulation

As mentioned above a quadratic approximation of the geometry of the four elements patch is chosen using the position of the six nodes in the patch. It is useful to define the patch in the isoparametric space using the nodal positions given in the Table 3.1 and Figure 3.1b. The quadratic interpolation is defined as (3.5a).

$$(3.5a) \quad \varphi = \sum_{i=1}^6 N_i \varphi_i$$

$$(3.5b) \quad \begin{aligned} N_1 &= \zeta + \xi\eta & N_4 &= \frac{\zeta}{2}(\zeta - 1) \\ N_2 &= \xi + \eta\zeta & N_5 &= \frac{\xi}{2}(\xi - 1) \\ N_3 &= \eta + \zeta\xi & N_6 &= \frac{\eta}{2}(\eta - 1) \end{aligned} \quad \text{with } \zeta = 1 - \xi - \eta$$

The Cartesian derivatives are defined as (3.5c).

$$(3.5c) \quad \begin{bmatrix} N_{i,1} \\ N_{i,2} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{bmatrix} = \begin{bmatrix} \varphi_{i,\xi}^0 \cdot \mathbf{t}_1 & \varphi_{i,\eta}^0 \cdot \mathbf{t}_1 \\ \varphi_{i,\xi}^0 \cdot \mathbf{t}_2 & \varphi_{i,\eta}^0 \cdot \mathbf{t}_2 \end{bmatrix} \begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{bmatrix}$$

	1	2	3	4	5	6
ξ	0	1	0	1	-1	1
η	0	0	1	1	1	-1

Table 3.1: Isoparametric coordinates of the six nodes in the patch of Figure 3.1b

3.3.3.1.1 Computation of the membrane strains :

We define the *Green-Lagrange* strains as (3.6a), and the virtual membrane strains as (3.6b). Differently from the original **BST** element the membrane strains within the **EBST** element are now a function of the displacements of the six patch nodes.

$$(3.6a) \quad \epsilon_m = \sum_{i=1}^3 \frac{1}{2} \tilde{N}_i \begin{bmatrix} \varphi_{i,1}^i \cdot \varphi_{i,1}^i - 1 \\ \varphi_{i,2}^i \cdot \varphi_{i,2}^i - 1 \\ 2\varphi_{i,1}^i \cdot \varphi_{i,2}^i \end{bmatrix}$$

$$(3.6b) \quad \delta \epsilon_m = \sum_{i=1}^3 \tilde{N}_i \begin{bmatrix} \varphi_{i,1}^i \cdot \delta' \mathbf{1}^i \\ \varphi_{i,2}^i \cdot \delta \varphi_{i,2}^i \\ \delta \varphi_{i,1}^i \cdot \varphi_{i,2}^i + \varphi_{i,1}^i \cdot \delta \varphi_{i,2}^i \end{bmatrix} = \mathbf{B}_m \delta \mathbf{a}^p$$

3.3.3.1.2 Computation of the bending strains :

We employ the same curvature field assumed in the **BST**.

$$(3.7a) \quad \kappa = 2 \sum_{i=1}^3 \begin{bmatrix} L_{i,1}^M & 0 \\ 0 & L_{i,2}^M \\ L_{i,2}^M & L_{i,1}^M \end{bmatrix} \begin{bmatrix} \mathbf{t}_3 \cdot \varphi_{i,1}^i \\ \mathbf{t}_3 \cdot \varphi_{i,2}^i \end{bmatrix}$$

The gradient is computed as (3.7b).

$$(3.7b) \quad \begin{bmatrix} \varphi_{i,1}^i \\ \varphi_{i,2}^i \end{bmatrix} = \begin{bmatrix} N_{1,1}^i & N_{2,1}^i & N_{3,1}^i & N_{i+3,1}^i \\ N_{1,2}^i & N_{2,2}^i & N_{3,2}^i & N_{i+3,2}^i \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_{i+3} \end{bmatrix}$$

Finally, the variation for curvature is (3.7c).

$$(3.7c) \quad \delta \boldsymbol{\kappa} = 2 \sum_{i=1}^3 \begin{bmatrix} L_{i,1}^M & 0 \\ 0 & L_{i,2}^M \\ L_{i,2}^M & L_{i,1}^M \end{bmatrix} \left\{ \sum_{i=1}^3 \begin{bmatrix} N_{j,1}^i (\mathbf{t}_3 \cdot \delta \mathbf{u}_j) \\ N_{j,2}^i (\mathbf{t}_3 \cdot \delta \mathbf{u}_j) \end{bmatrix} + \begin{bmatrix} N_{i+3,1}^i (\mathbf{t}_3 \cdot \delta \mathbf{u}^{i+3}) \\ N_{i+3,2}^i (\mathbf{t}_3 \cdot \delta \mathbf{u}^{i+3}) \end{bmatrix} \right\} \\ - \sum_{i=1}^3 \begin{bmatrix} (L_{i,1}^M \rho_{11}^1 + L_{i,2}^M \rho_{11}) \\ (L_{i,1}^M \rho_{22}^1 + L_{i,2}^M \rho_{22}) \\ (L_{i,1}^M \rho_{12}^1 + L_{i,2}^M \rho_{12}) \end{bmatrix} (\mathbf{t}_3 \cdot \delta \mathbf{u}_i) = \mathbf{B}_b \delta \mathbf{a}^p$$

3.3.3.1.3 Tangent stiffness matrix :

The tangent stiffness matrix is divided into material and geometric components. Being the material tangent matrix the one showed in (3.8a), and the geometric tangent matrix the one from (3.8b).

Material stiffness matrix:

$$(3.8a) \quad \mathbf{K}^M = \int \int_{A_M^0} \mathbf{B}^T \mathbf{D}_{ep} \mathbf{B} dA \text{ where } \mathbf{B} = \mathbf{B}_m + \mathbf{B}_b$$

Geometric stiffness matrix:

$$(3.8b) \quad \mathbf{K}^G = \mathbf{K}_m^G + \mathbf{K}_b^G$$

Where:

$$(3.8c) \quad \delta \mathbf{u}^T \mathbf{K}_m^G \Delta \mathbf{u} = \frac{A^M}{3} \sum_{k=1}^3 \sum_{i=1}^6 \sum_{j=1}^6 \left\{ \delta \mathbf{u}_i \begin{bmatrix} N_{i,1}^k & N_{i,2}^k \end{bmatrix} \begin{bmatrix} N_{1,1}^k & N_{1,2}^k \\ N_{2,1}^k & N_{2,2}^k \end{bmatrix} \begin{bmatrix} N_{j,1}^k \\ N_{j,2}^k \end{bmatrix} \Delta \mathbf{u}_j \right\} \text{ with } N_{ij} = \sigma_{mij}$$

Numerical experiments [ArtFO05; ArtFO11] have shown that the bending part of the geometric stiffness is not so important and can be disregarded in the iterative process.

3.4 Prismatic solid-shell

3.4.1 Introduction

Solid-shells have been during the last fifteen years [ArtDB84; ArtHS98; ArtHSD00; ArtHau+01; ArtSou+05; ArtKGW06a; ArtPar+06] an important improvement in the shells simulations, providing reliable simulations and avoiding the problematic that is associated to the kinematics hypothesis and plane stress constitutive laws related to the use of the shell element (or any kind of bidimensional elements as plates and membranes). The main advantages when using **solid-shell** elements are:

- General 3D constitutive relations
- Large transverse shear can be considered, and considering additional elements along the thickness improve this behaviour
- There does not need to consider transitions between solid and shell elements (all the elements are solids)
- Contact forces can be introduced directly in the geometry and in a realistic way without any additional technique, which is especially important for the consideration of friction

- e) The element is **rotation-free**, avoiding the storage and computation of these variables
- f) In the case where we have non-parallel boundaries this can be modelled correctly

For the consideration of strongly non-linear problems, problems where the contact-friction, large deformations and complex constitutive laws are considered, the use of low interpolation order elements is preferred. Most of the existing **solid-shells** are linear hexahedron [ArtDB84; ArtHS98; ArtHSD00; ArtHau+01; ArtSou+05; ArtKGW06a; ArtPar+06; ArtSVR11; ArtAC09; ArtSAV11; ArtSR11; ArtSen+16] (usually trilinear 8-node brick), which have two main disadvantages; the first one is the *hourglass* effect, which is called this way due to the characteristics shapes adopted in the proper modes and a stabilisation is required to reduce these problems; the second problematic is the meshing of the plane, due to the fact that meshing quadrilateral is less performant than triangles. For this reasons the triangular prisms (wedges) could be considered an interesting alternative, especially for the second problem mentioned, but this kind of geometry is not exempt of problematic, owing to the low order of interpolation of the geometry, when a linear triangular prism is considered. This last problem can be solved with the consideration of the neighbours elements³, in consequence the element becomes quadratic in the plane solving this last problem.

In addition to this, it is well known that low interpolation order elements in the standard displacement formulation (in contract to the mixed formulation) in the consideration of slender structures and incompressible materials suffer severe locking effects. The *transverse shear locking* provokes problems in the bending behaviour, especially when more slender is the element. The *membrane locking* appears especially in the initially curved shells when bending is preponderant without middle surface stretching. A *curvature thickness locking* can appear in problems with initially curved geometry due to artificial transverse strains and stresses under pure bending. Finally, the *volumetric locking* can appear when the material present an incompressible, nearly incompressible behaviour or elastic-plastic materials with isochoric plastic flow (typical in metals).

The element has been implemented into *Kratos* [OnlKra], the in-home **FEM-Multiphysics** open-source code, implemented in **C++** with parallelisation capabilities. The pre/post-process of all the presented examples haven been processed with *GiD* [OnlMel+16], the **CIMNE** software for pre and post processing.

3.4.2 Basic kinematics of the standard element

We can find the standard isoparametric interpolations for the linear 6-node triangular prism (or wedge) in (3.9), where \mathbf{X}^l , \mathbf{x}^l and \mathbf{u}^l correspond with the original coordinates (or undeformed configuration), the current coordinates (or deformed configuration) and the displacements in the node l respectively.

$$(3.9a) \quad \mathbf{X}(\xi) = \sum_{l=1}^6 N^l(\xi) \mathbf{X}^l$$

$$(3.9b) \quad \mathbf{x}(\xi) = \sum_{l=1}^6 N^l(\xi) \mathbf{x}^l = \sum_{l=1}^6 (\mathbf{X}^l + \mathbf{u}^l)$$

The shape function $N^l(\xi)$ from (3.10) are defined in function of the local coordinates (ξ, η, ζ) , where the tho first define the position in the plane of the triangular base and the third one corresponds with the coordinate along the prism axis.

$$(3.10a) \quad \begin{aligned} N^1 &= zL^1, & N^2 &= \xi L^1, & N^3 &= \eta L^1 \\ N^4 &= zL^2, & N^5 &= \xi L^2, & N^6 &= \eta L^2 \end{aligned}$$

In this definition the third triangular coordinate and the axis interpolation is defined as:

$$(3.10b) \quad z = 1 - \xi - \eta, \quad L^1 = \frac{1}{2}(1 - \zeta), \quad L^2 = \frac{1}{2}(1 + \zeta)$$

³Like in the **EBST** elements, see 3.3.Rotation-free shells.

We can define following the standard formulation the *Jacobian* matrix at each integration point as (3.15a), in consequence we can compute the *Cartesian* derivatives of the shape functions. At each element centre, a local *Cartesian* triad can be defined as in (3.11b), that allows to compute the *Cartesian* derivatives with respect to this local system following the orthotropic directions.

$$(3.11a) \quad \mathbf{J} = \frac{\partial \mathbf{X}}{\partial \xi} \rightarrow N'_X = \mathbf{J}^{-1} N'_\xi$$

$$(3.11b) \quad \mathbf{R} = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3] \rightarrow N'_Y = \mathbf{R}^T N'_X$$

As it will be shown in the following section, the left *Cauchy* tensor \mathbf{C} is modified into the $\bar{\mathbf{C}}$ using the assumed strain techniques that in one case include an additional internal degree of freedom α , leading to the improved tensor $\bar{\mathbf{C}}$. The balance equation to solve in the strong form for a \mathbf{TL}^4 formulation in **large strain** hypothesis is (3.12).

$$(3.12) \quad \begin{cases} g_1(\mathbf{u}, \alpha) = \int_{V_0} \frac{1}{2} \mathbf{S}(\bar{\mathbf{C}}) : \delta_u \bar{\mathbf{C}} dV_0 + g_{ext} = 0 \\ g_2(\mathbf{u}, \alpha) = \int_{V_0} \frac{1}{2} \mathbf{S}(\bar{\mathbf{C}}) : \delta_\alpha \bar{\mathbf{C}} dV_0 + g_{ext} = 0 \end{cases}$$

Where \mathbf{S} is the second *Piola-Kirchoff* stress tensor (**PK2**).

3.4.3 Modifications of the standard element

The prism presented in the previous lines corresponds with the standard, and it needs to be modified for the sake of large strain elastic-plastic analysis of shells. For this purpose different modifications are introduced in the metric tensor \mathbf{C} . The discretisation to be introduced in the prism solid-shell can be defined in first place with the discretisation of the triangular middle surface and in second place with the discretisation along the thickness. We will assume that the upper and lower face are almost parallel and thus the normal direction can be defined in function of ζ . In (3.13) the right *Cauchy-Green* tensor is decomposed according to the different behaviours that define classically the shell elements, this means we decompose in-plane (membrane and bending behaviour), transverse shear and normal components. The calculation of each one of the different components will be detailed in the next sections.

$$(3.13a) \quad \mathbf{C} = \begin{bmatrix} C_{11}^m & C_{12}^m & C_{13}^s \\ C_{21}^m & C_{22}^m & C_{23}^s \\ C_{31}^m & C_{32}^m & C_{33}^n \end{bmatrix}$$

Where the index m , s and n mean membrane, shear and normal behaviour, respectively. In consequence this tensor can be decomposed in three components as detailed in (3.13b).

$$(3.13b) \quad \mathbf{C} = \mathbf{C}_1 + \mathbf{C}_2 + \mathbf{C}_3 \rightarrow \begin{cases} \mathbf{C}_1 = C_{11} \mathbf{t}^1 \otimes \mathbf{t}^1 + C_{22} \mathbf{t}^2 \otimes \mathbf{t}^2 + C_{12} (\mathbf{t}^1 \otimes \mathbf{t}^2 + \mathbf{t}^2 \otimes \mathbf{t}^1) \text{ (tangent plane)} \\ \mathbf{C}_2 = C_{13} (\mathbf{t}^1 \otimes \mathbf{t}^3 + \mathbf{t}^3 \otimes \mathbf{t}^1) + C_{23} (\mathbf{t}^2 \otimes \mathbf{t}^3 + \mathbf{t}^3 \otimes \mathbf{t}^2) \text{ (transverse shear)} \\ \mathbf{C}_3 = C_{33} \mathbf{t}^3 \otimes \mathbf{t}^3 \text{ (thickness strain)} \end{cases}$$

3.4.3.1 In plane behaviour

The improvement in the in-plane behaviour is the same considered for the **EBST** element, a rotation-free shell element, where a four-element patch is considered (Figure 3.2), and the neighbour nodes allow us to work with a quadratic element in the in-plane behaviour; for more information about this element, we address the reader to [ArtOF05; ArtFO05]. The same computations from the **EBST** element can be considered in the upper and lower face as we can see in the quadratic shape functions from (3.14) and its derivatives in Table 3.2.

⁴The formulation can be extended to **UL** following the standard notation, for more information you can look among others [BookBel+14; BookBor+12].

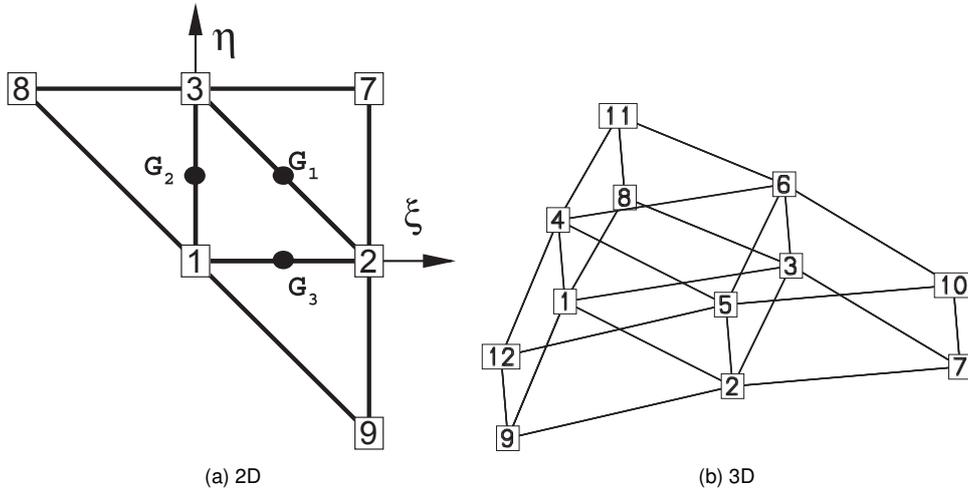


Figure 3.2: Patch performed in the element considering the neighbour elements

$$(3.14) \quad \begin{aligned} N^1 &= (z + \xi\eta), & N^7 &= \frac{z}{2}(z - 1), \\ N^2 &= (\xi + \eta z), & N^8 &= \frac{z}{2}(\xi - 1), \\ N^3 &= (\eta + z\xi), & N^9 &= \frac{z}{2}(\eta - 1) \end{aligned}$$

We define a local system of coordinates from (3.11b) taking as reference the components \mathbf{f}_1 and \mathbf{f}_2 in the tangent plane and \mathbf{f}_3 in the normal. In each mid-side point of the element, we compute the in-plane *Jacobian* as shown in (3.15a).

$$(3.15a) \quad \mathbf{J} = \begin{bmatrix} \mathbf{X}_\xi \cdot \mathbf{t}_1 & \mathbf{X}_\eta \cdot \mathbf{t}_1 \\ \mathbf{X}_\xi \cdot \mathbf{t}_2 & \mathbf{X}_\eta \cdot \mathbf{t}_2 \end{bmatrix}$$

And in combination of the shape function derivatives we can compute the *Cartesian* derivatives in (3.15b).

$$(3.15b) \quad \begin{bmatrix} N'_1 \\ N'_2 \end{bmatrix}^K = \mathbf{J}_K^{-1} \begin{bmatrix} N'_\xi \\ N'_\eta \end{bmatrix}^K$$

With the *Cartesian* derivatives, we can calculate the in-plane deformation gradient components \mathbf{f}_1^K and \mathbf{f}_2^K , and with it \bar{C}_{ij}^K which are averaged over each integration point along the thickness with (3.16a). As with the rotation-free shells, when a neighbour is missing the values from the central node of the face are used for the averaging.

$$(3.16a) \quad \bar{C}_{ij}(\xi) = L^1 \bar{C}_{ij}^1 + L^2 \bar{C}_{ij}^2$$

The variations are computed as (3.16b).

$$(3.16b) \quad \delta \begin{bmatrix} \frac{1}{2} \bar{C}_{11} \\ \frac{1}{2} \bar{C}_{22} \\ C_{12} \end{bmatrix} = \delta \begin{bmatrix} \frac{1}{2} \bar{C}_{11}^1 \\ \frac{1}{2} \bar{C}_{22}^1 \\ C_{12}^1 \end{bmatrix} L^1 + \delta \begin{bmatrix} \frac{1}{2} \bar{C}_{11}^2 \\ \frac{1}{2} \bar{C}_{22}^2 \\ C_{12}^2 \end{bmatrix} L^2 = \delta \begin{bmatrix} \bar{E}_{11} \\ \bar{E}_{22} \\ 2\bar{E}_{12} \end{bmatrix}$$

	1	2	3	7	8	9
N'_ξ	$-1 + \eta$	$1 - \eta$	$z - \xi$	$\frac{1}{2} - z$	$\xi - \frac{1}{2}$	0
N'_η	$-1 + \xi$	$z - \eta$	$1 - \xi$	$\frac{1}{2} - z$	0	$\eta - \frac{1}{2}$

Table 3.2: Derivatives of the shape functions from (3.14)

At each face a modified tangent matrix $\bar{\mathbf{B}}^f$ relating the incremental tensor components with the incremental displacements $\delta \mathbf{u}$ can be written as shown in (3.17), where just the nodes from the face and its opposite neighbour is

considered.

$$(3.17) \quad \delta \begin{bmatrix} \frac{1}{2} \bar{C}_{11}^f \\ \frac{1}{2} \bar{C}_{22}^f \\ \bar{C}_{12}^f \end{bmatrix} = \frac{1}{3} \sum_{K=1}^3 \delta \begin{bmatrix} \frac{1}{2} \bar{C}_{11}^K \\ \frac{1}{2} \bar{C}_{22}^K \\ \bar{C}_{12}^K \end{bmatrix} = \frac{1}{3} \sum_{K=1}^3 \sum_{J=1}^4 \begin{bmatrix} \mathbf{f}_1^K N_1^{J(K)} \\ \mathbf{f}_2^K N_2^{J(K)} \\ (\mathbf{f}_1^K N_2^{J(K)} + \mathbf{f}_2^K N_1^{J(K)}) \end{bmatrix} \delta \mathbf{u}^{J(K)} \\ = (\bar{\mathbf{B}}_m^f)_{3 \times 18} \delta \mathbf{u}^f \rightarrow [\bar{\mathbf{B}}_m]_{3 \times 36} = [L^1 \bar{\mathbf{B}}_m^1 \quad L^2 \bar{\mathbf{B}}_m^2]$$

We can obtain the equivalent nodal force vector from the integral (3.18), with $\int_{-1}^1 S_{ij} L^f J d\xi = \bar{S}_{ij}^f$.

$$(3.18) \quad \mathbf{r}_m^T \delta \mathbf{u} = \int_{-1}^1 \begin{bmatrix} S_{11} \\ S_{22} \\ S_{12} \end{bmatrix}^T [L^1 \bar{\mathbf{B}}_m^1 \quad L^2 \bar{\mathbf{B}}_m^2] J d\xi = \left\{ \begin{bmatrix} \bar{S}_{11}^1 \\ \bar{S}_{22}^1 \\ \bar{S}_{12}^1 \end{bmatrix}^T \bar{\mathbf{B}}_m^1, \begin{bmatrix} \bar{S}_{11}^2 \\ \bar{S}_{22}^2 \\ \bar{S}_{12}^2 \end{bmatrix}^T \bar{\mathbf{B}}_m^2 \right\} \delta \mathbf{u}$$

In (3.19) the calculation of geometric stiffness for the membrane behaviour is presented, where we sum the contribution of the n_G integration points along the direction ζ .

$$(3.19a) \quad \delta \mathbf{u}^T \mathbf{K}_{mG} \Delta \mathbf{u} = \int_V \frac{\partial}{\partial \mathbf{u}} \left(\delta \begin{bmatrix} \frac{1}{2} \bar{C}_{11} \\ \frac{1}{2} \bar{C}_{22} \\ \bar{C}_{12} \end{bmatrix} \right)^T \begin{bmatrix} S_{11} \\ S_{22} \\ S_{12} \end{bmatrix} \Delta \mathbf{u} dV \\ = \sum_{G=1}^{n_G} \frac{Vol_G}{3} \sum_{f=1}^2 L^f \sum_{K=1}^3 \sum_{J=1}^4 \left\{ \delta \mathbf{u}^f [N_1^f \quad N_2^f] \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} N_1^f \\ N_2^f \end{bmatrix} \right\}^K$$

$$(3.19b) \quad \delta \mathbf{u} \mathbf{K}_{mG} \Delta \mathbf{u} = \sum_{f=1}^2 \sum_{K=1}^3 \sum_{J=1}^4 \left\{ \delta \mathbf{u}^f [N_1^f \quad N_2^f] \begin{bmatrix} \sum_{G=1}^{n_G} \frac{Vol_G}{3} L^f \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} \\ \bar{S}_{21} & \bar{S}_{22} \end{bmatrix} \begin{bmatrix} N_1^f \\ N_2^f \end{bmatrix} \right\}^K \Delta \mathbf{u}^J \right\} \\ = \sum_{f=1}^2 \left\{ \sum_{K=1}^3 \sum_{J=1}^4 \left\{ \delta \mathbf{u}^f [N_1^f \quad N_2^f] \begin{bmatrix} \bar{S}_{11}^f & \bar{S}_{12}^f \\ \bar{S}_{21}^f & \bar{S}_{22}^f \end{bmatrix} \begin{bmatrix} N_1^f \\ N_2^f \end{bmatrix} \right\}^K \Delta \mathbf{u}^J \right\}^f$$

3.4.3.2 Transverse shear behaviour

To avoid the transverse shear locking we introduce an interpolation in natural coordinates of mixed tensorial components, a very common practice in the literature. We consider a linear variation of the transverse shear strain tangent to the side. Here we compute a mixed component of the right *Cauchy-Green* tensor as (3.20), where the components relative to the transverse shear $C_{\eta 3}$ and $C_{\xi 3}$ are written with respect to a mixed coordinate system that includes the in-plane natural coordinates (ξ, η) and the spatial local coordinate in the transverse direction (y_3) . The components are written in terms of the transverse shear strain tangent to the side computed in each mid-side point, as seen in Figure 3.3. Besides, the numerical integration is performed along the axis, which means $\mathbf{P}(\frac{1}{3}, \frac{1}{3})$.

$$(3.20) \quad \begin{bmatrix} C_{\xi 3} \\ C_{\eta 3} \end{bmatrix} = \begin{bmatrix} -\eta & -\eta & 1 - \eta \\ \xi & \xi - 1 & \xi \end{bmatrix} \begin{bmatrix} \sqrt{2} C_{t3}^1 \\ -C_{\eta 3}^2 \\ C_{\xi 3}^3 \end{bmatrix} = \mathbf{P}(\xi, \eta) \begin{bmatrix} \sqrt{2} \mathbf{f}_t^1 \cdot \mathbf{f}_3^1 \\ -\mathbf{f}_\eta^2 \cdot \mathbf{f}_3^2 \\ \mathbf{f}_\xi^3 \cdot \mathbf{f}_3^3 \end{bmatrix}$$

The deformation gradient components are \mathbf{f}_t (natural coordinate derivative) and \mathbf{f}_3 (local Cartesian coordinate derivative), these are expressed in (3.21a), where \mathbf{j}_3^{-T} are third column of the inverse of the transverse of the *Jacobian*.

$$(3.21a) \quad \begin{bmatrix} \bar{C}_{12} \\ \bar{C}_{23} \end{bmatrix} = \mathbf{J}_p^{-1} \begin{bmatrix} C_{\xi 3} \\ C_{\eta 3} \end{bmatrix} = \mathbf{J}_p^{-1} \mathbf{P}_a \begin{bmatrix} \mathbf{f}_t^1 \cdot \mathbf{f}_3^1 \\ -\mathbf{f}_\eta^2 \cdot \mathbf{f}_3^2 \\ \mathbf{f}_\xi^3 \cdot \mathbf{f}_3^3 \end{bmatrix}$$

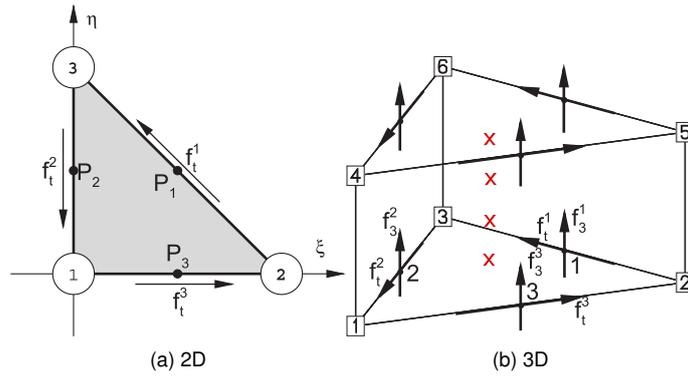


Figure 3.3: Nodes considered in the computation of the transverse shear strains

The modified transverse shear Cartesian components can be obtained from (3.21b), interpolating these values the components from the right *Cauchy-Green* tensor can be obtained (3.21c).

$$(3.21b) \quad \begin{bmatrix} \sqrt{2}\mathbf{f}_t^1 \\ -\mathbf{f}_\eta^2 \\ \mathbf{f}_\xi^3 \end{bmatrix}^1 = \begin{bmatrix} \mathbf{x}^3 - \mathbf{x}^2 \\ \mathbf{x}^1 - \mathbf{x}^3 \\ \mathbf{x}^2 - \mathbf{x}^1 \end{bmatrix}, \quad \begin{bmatrix} \sqrt{2}\mathbf{f}_t^1 \\ -\mathbf{f}_\eta^2 \\ \mathbf{f}_\xi^3 \end{bmatrix}^2 = \begin{bmatrix} \mathbf{x}^6 - \mathbf{x}^5 \\ \mathbf{x}^4 - \mathbf{x}^6 \\ \mathbf{x}^5 - \mathbf{x}^4 \end{bmatrix}, \quad \mathbf{f}_3 = \sum_{l=1}^6 N_3^l \mathbf{x}^l = \nabla_\xi(\mathbf{x})\mathbf{j}_3^{-T}$$

$$(3.21c) \quad \begin{bmatrix} \bar{\mathbf{C}}_{13} \\ \bar{\mathbf{C}}_{23} \end{bmatrix}(\zeta) = \begin{bmatrix} \bar{\mathbf{C}}_{12} \\ \bar{\mathbf{C}}_{23} \end{bmatrix}^1 L^1 + \begin{bmatrix} \bar{\mathbf{C}}_{13} \\ \bar{\mathbf{C}}_{23} \end{bmatrix}^2 L^2 \quad \text{and} \quad \begin{bmatrix} 2E_{13} \\ 2E_{23} \end{bmatrix}(\zeta) = \begin{bmatrix} \bar{\mathbf{C}}_{13} \\ \bar{\mathbf{C}}_{23} \end{bmatrix}(\zeta)$$

The tangent matrix $\bar{\mathbf{B}}_s$ is also obtained by interpolating from both faces (3.22a). In a similar way to the in-plane behaviour, the internal forces can be obtained from (3.22b). Finally, the geometric stiffness matrix can be obtained from (3.22c).

$$(3.22a) \quad \bar{\mathbf{B}}_s(\zeta) = \bar{\mathbf{B}}_s^1 L^1 + \bar{\mathbf{B}}_s^2 L^2 \quad \text{where} \quad \bar{\mathbf{B}}_s^f = (\mathbf{J}_p^f)^{-1} \mathbf{P}_a \bar{\mathbf{B}}_s^f \quad \text{and} \quad \bar{\mathbf{B}}_s \delta \mathbf{u}^e = \begin{bmatrix} \delta \mathbf{f}_t^1 \cdot \mathbf{f}_3^1 + \mathbf{f}_t^1 \cdot \delta \mathbf{f}_3^1 \\ -\delta \mathbf{f}_\eta^2 \cdot \mathbf{f}_3^2 - \mathbf{f}_\eta^2 \cdot \delta \mathbf{f}_3^2 \\ \delta \mathbf{f}_\xi^3 \cdot \mathbf{f}_3^3 + \mathbf{f}_\xi^3 \cdot \delta \mathbf{f}_3^3 \end{bmatrix}$$

$$(3.22b) \quad \mathbf{r}_s^T = \int_V \begin{bmatrix} \bar{\mathbf{S}}_{12} \\ \bar{\mathbf{S}}_{23} \end{bmatrix} [\bar{\mathbf{B}}_s]_{2 \times 18} dV = \bar{\mathbf{Q}}_{4 \times 1}^T \begin{bmatrix} \bar{\mathbf{B}}_s^1 \\ \bar{\mathbf{B}}_s^2 \end{bmatrix}_{4 \times 18} \quad \text{where} \quad \bar{\mathbf{Q}}_{4 \times 1}^T = \int_{-1}^1 \begin{bmatrix} \bar{\mathbf{S}}_{12} \\ \bar{\mathbf{S}}_{23} \\ \bar{\mathbf{S}}_{12} \\ \bar{\mathbf{S}}_{23} \end{bmatrix} \begin{bmatrix} L^1 \\ L^2 \end{bmatrix} J d\zeta$$

$$(3.22c) \quad \delta \mathbf{u}^T \mathbf{K}_{sG} \Delta \mathbf{u} = \Delta \left\{ \begin{bmatrix} \bar{\mathbf{B}}_s^1 \\ \bar{\mathbf{B}}_s^2 \end{bmatrix} \delta \mathbf{u} \right\}^T \bar{\mathbf{Q}} \quad \text{considering} \quad \begin{bmatrix} \bar{\mathbf{Q}}_1' & \bar{\mathbf{Q}}_2' \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{Q}}_1 & \bar{\mathbf{Q}}_2 \end{bmatrix} \mathbf{J}_p^{-1} \rightarrow$$

$$\begin{bmatrix} \bar{\mathbf{Q}}_1 & \bar{\mathbf{Q}}_2 \end{bmatrix} \Delta (\bar{\mathbf{B}}_s^1 \delta \mathbf{u}) = \frac{1}{3} \begin{pmatrix} (-\bar{\mathbf{Q}}_1' + \bar{\mathbf{Q}}_2') \left(\sqrt{2} \delta \mathbf{f}_t^1 \cdot \Delta \mathbf{f}_3^1 + \sqrt{2} \Delta \mathbf{f}_t^1 \cdot \delta \mathbf{f}_3^1 \right) \\ (-\bar{\mathbf{Q}}_1' - 2\bar{\mathbf{Q}}_2') \left(-\delta \mathbf{f}_\eta^2 \cdot \Delta \mathbf{f}_3^2 + \Delta \mathbf{f}_\eta^2 \cdot \delta \mathbf{f}_3^2 \right) \\ (2\bar{\mathbf{Q}}_1' + \bar{\mathbf{Q}}_2') \left(\delta \mathbf{f}_\xi^3 \cdot \Delta \mathbf{f}_3^3 + \Delta \mathbf{f}_\xi^3 \cdot \delta \mathbf{f}_3^3 \right) \end{pmatrix}$$

3.4.3.3 Transverse normal behaviour

As introduced previously, to avoid locking (volumetric locking) in quasi-incompressible problems due to the *Poisson* effect the **EAS** formulation is considered. With this formulation we obtain a modified \mathbf{C}_3 component.

3.4.3.3.1 EAS formulation :

In the standard **EAS** method the convective strain components are interpolated, in our case as we just want to improve \mathbf{C}_3 and some modifications as presented below are considered. At the element centre ($\xi = \eta = \frac{1}{3}$ and $\zeta = 0$) the *Cartesian* deformation gradient, and in consequence the right *Cauchy* tensor, can be enhanced as shown in (3.23). So, in this **EAS** the changes will affect just to the \mathbf{C}_{33} component, and the \mathbf{C}_{13} and \mathbf{C}_{23} are computed as presented in the previous section.

$$(3.23) \quad \mathbf{f}_3^C = \sum_{l=1}^6 N_3^{lC} \mathbf{X}^l \rightarrow \bar{\mathbf{f}}_3 = \mathbf{f}_3^C e^{\alpha\zeta} \rightarrow \bar{\mathbf{C}}_{33} = \bar{\mathbf{f}}_3 \cdot \bar{\mathbf{f}}_3 = \mathbf{C}_{33}^2 e^{2\alpha\zeta}$$

With this enhancement the deformation matrix, internal forces and geometric stiffness can be calculated as shown in (3.24).

$$(3.24a) \quad \delta \bar{E}_{33} = \frac{1}{2} \delta \bar{\mathbf{C}}_{33} = \delta \mathbf{f}_3^C \cdot \mathbf{f} - \frac{C}{3} e^{2\alpha\zeta} + \bar{\mathbf{C}}_{33} \zeta \delta \alpha = \left(\sum_{l=1}^6 N_3^{lC} \delta \mathbf{u}^l \right) \cdot \mathbf{f}_3^C e^{2\alpha\zeta} + \bar{\mathbf{C}}_{33} \zeta \delta \alpha = e^{2\alpha\zeta} \mathbf{B}_3^C \delta \mathbf{u}^e + \bar{\mathbf{C}}_{33} \zeta \delta \alpha$$

$$(3.24b) \quad \mathbf{r}_n^T = \int_V \mathbf{B}_3 \mathbf{S}_{33} dV = \int_{-1}^1 e^{2\alpha\zeta} \mathbf{B}_3^C \mathbf{S}_{33} J d\zeta = \mathbf{B}_3^C \bar{\mathbf{S}}_{33} \text{ with } \bar{\mathbf{S}}_{33} = \int_{-1}^1 e^{2\alpha\zeta} \mathbf{S}_{33} J d\zeta$$

$$(3.24c) \quad \delta \mathbf{u}^T \mathbf{K}_{Gn} \Delta \mathbf{u} = \delta \mathbf{f}_3^C \cdot \Delta \mathbf{f}_3^C \bar{\mathbf{S}}_{33} = \sum_{l=1}^6 (\delta \mathbf{u}^l)^T \sum_{j=1}^6 N_3^{lC} N_3^{jC} \bar{\mathbf{S}}_{33} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \Delta \mathbf{u}^j$$

3.4.3.3.2 Balance equation :

In the following lines, we will introduce the balance equation which allows us to obtain the implicit solution of the problem. Because we will focus in the implicit solution of the problem we will address the reader to [ArtFlo13c; ArtFlo13a; ArtFlo13b] where the explicit solution of the problem is presented too.

The balance equation related with α **DOF** from (3.12) can be expressed as (3.25a), where we can obtain the residue and approximate the solution with a *Newton-Rapson* technique the residual can be nullified (3.25b).

$$(3.25a) \quad \delta \alpha \int_{-1}^1 \mathbf{S}_{33} \bar{\mathbf{C}}_{33} \zeta J d\zeta = 0 \rightarrow r_\alpha = \int_{-1}^1 \mathbf{S}_{33} \bar{\mathbf{C}}_{33} \zeta J d\zeta$$

$$(3.25b) \quad \int_{-1}^1 \left[\frac{\partial \mathbf{S}_{33} \bar{\mathbf{C}}_{33}}{\partial \mathbf{u}} \Delta \mathbf{u} + \frac{\partial \mathbf{S}_{33} \bar{\mathbf{C}}_{33}}{\partial \alpha} \Delta \alpha \right] \zeta J d\zeta + r_\alpha = 0 \text{ with } \begin{cases} \frac{\partial \mathbf{S}_{33} \bar{\mathbf{C}}_{33}}{\partial \mathbf{u}} = \bar{\mathbf{C}}_{33} \mathbf{D}_3 \bar{\mathbf{B}} + 2 \mathbf{S}_{33} \bar{\mathbf{B}}_3 \\ \frac{\partial \mathbf{S}_{33} \bar{\mathbf{C}}_{33}}{\partial \alpha} = \bar{\mathbf{C}}_{33} (\mathbf{D}_{33} \bar{\mathbf{C}}_{33} \zeta + 2 \mathbf{S}_{33} \zeta) \end{cases}$$

The previous expression can be condensed in the following expression (3.25c), where after integrate the operators \mathbf{H} and k_α are obtained.

$$(3.25c) \quad \mathbf{H}_{1 \times 18} \Delta \mathbf{u} + k_\alpha \Delta \alpha + r_\alpha = 0 \text{ then } \Delta \alpha = \frac{r_\alpha}{k_\alpha} - \frac{1}{k_\alpha} \mathbf{H} \Delta \mathbf{u}$$

Where:

$$(3.25d) \quad \begin{cases} \mathbf{H}_{1 \times 18} = \int_{-1}^1 (\bar{\mathbf{C}}_{33} \mathbf{D}_3 \bar{\mathbf{B}} + 2 \mathbf{S}_{33} \bar{\mathbf{B}}_3) \zeta J d\zeta \\ k_\alpha = \int_{-1}^1 \bar{\mathbf{C}}_{33} (\bar{\mathbf{C}}_{33} \mathbf{D}_{33} + 2 \mathbf{S}_{33}) \zeta^2 J d\zeta \end{cases}$$

If the expression is introduced in the balance equation associated with the displacement in (3.12) we can obtain (3.25e), which can be linearised in the expression (3.25f).

$$(3.25e) \quad \delta \mathbf{u}^T \int_V \bar{\mathbf{B}}^T \mathbf{S} dV - \delta \mathbf{u}^T \mathbf{G}_{ext} = \delta \mathbf{u}^T \mathbf{r}(\mathbf{u}, \alpha)$$

$$(3.25f) \quad \int_{-1}^1 \left[\mathbf{B}^T \left(\frac{\partial \mathbf{S}}{\partial \mathbf{u}} \Delta \mathbf{u} + \frac{\partial \mathbf{S}}{\partial \alpha} \Delta \alpha \right) + \left(\frac{\partial \bar{\mathbf{B}}^T}{\partial \mathbf{u}} \Delta \mathbf{u} + \frac{\partial \bar{\mathbf{B}}^T}{\partial \alpha} \Delta \alpha \right) \mathbf{S} \right] J d\zeta + \mathbf{r}(\mathbf{u}) = \mathbf{0}$$

$$\int_{-1}^1 \left[\mathbf{B}^T (\mathbf{D}\bar{\mathbf{B}}\Delta\mathbf{u} + \mathbf{D}_3^T \bar{\mathbf{C}}_{33} \zeta \Delta\alpha) + \mathbf{S}\mathbf{G}\Delta\mathbf{u} + 2\mathbf{S}_{33}\bar{\mathbf{B}}_3^T \zeta \Delta\alpha \right] J d\zeta + \mathbf{r}(\mathbf{u}) = \mathbf{0}$$

$$\int_{-1}^1 \left[(\mathbf{B}^T \mathbf{D}\bar{\mathbf{B}}\mathbf{S}\mathbf{G}) \Delta\mathbf{u} + (\bar{\mathbf{B}}^T \mathbf{D}_3^T \bar{\mathbf{C}}_{33} + 2\mathbf{S}_{33}\bar{\mathbf{B}}_3^T) \zeta \Delta\alpha \right] J d\zeta + \mathbf{r}(\mathbf{u}) = \mathbf{0}$$

$$(\mathbf{K}_T \Delta \mathbf{u} + \mathbf{H}^T \Delta \alpha) + \mathbf{r}(\mathbf{u}) = \mathbf{0} \rightarrow \mathbf{K}_T \Delta \mathbf{u} - \mathbf{H}^T \left(\frac{r_\alpha}{k_\alpha} + \frac{1}{k_\alpha} \mathbf{H} \Delta \mathbf{u} \right) + \mathbf{r}(\mathbf{u}) = \mathbf{0}$$

Finally, all this can be expressed (3.25g) as a modification of the elemental tangent stiffness matrix and the internal forces.

$$(3.25g) \quad \left(\mathbf{K}_T - \mathbf{H}^T \frac{1}{k_\alpha} \mathbf{H} \right) \Delta \mathbf{u} - \mathbf{H}^T \frac{r_\alpha}{k_\alpha} + \mathbf{r}(\mathbf{u}) = \mathbf{0} \rightarrow \begin{cases} \bar{\mathbf{K}}_T = \mathbf{K}_T - \mathbf{H}^T \frac{1}{k_\alpha} \mathbf{H} \\ \bar{\mathbf{r}} = \mathbf{r} - \mathbf{H}^T \frac{r_\alpha}{k_\alpha} \end{cases}$$

3.4.3.3.3 Pull-Back and Push-Forward (Extension of the formulation) :

The formulation presented until now is broadly the formulation already presented in the works of *Fernando Flores* [ArtFlo13c; ArtFlo13a; ArtFlo13b], the main step forward of this work is the extension of the formulation. As we have seen, the formulation presented allows us to obtain a modified left *Cauchy* tensor $\bar{\mathbf{C}}$. With this tensor we are able to obtain strains in a traditional way, *g.e* with *Green-Lagrange*, but we are unable to work with the **Pull-Back** and **Push-Forward** operations, these are, furthermore, the underlying operations in the constitutive operations in *Kratos*. The fundamental concepts for this are presented in the Appendix section [A.1.Pull-Back, Push-Forward fundamental concepts](#).

In order to perform these operations, we need the deformation gradient \mathbf{F} , or in our case $\bar{\mathbf{F}}$ owing to we are working with a modified right *Cauchy* tensor $\bar{\mathbf{C}}$. Obtain one from another is not a trivial operation, and we must consider additional assumptions to obtain our modified deformation gradient. In a standard formulation to obtain the deformation gradient \mathbf{F} we compute (3.26) from the material displacement gradient tensor $\nabla_{\mathbf{x}} \mathbf{u}$, \mathbf{C} can be obtained easily from here.

$$(3.26) \quad \mathbf{F} = \nabla_{\mathbf{x}} \mathbf{u} + \mathbf{I} \rightarrow \mathbf{C} = \mathbf{F}^T \mathbf{F}$$

We will present now (3.27a) the polar decomposition of \mathbf{F} , which will be the key idea considered to obtain $\bar{\mathbf{F}}$. In this decomposition \mathbf{R} represents the *proper orthogonal tensor* and \mathbf{U} is the *right stretch tensor*. The *right stretch tensor* can be computed from the square root of the right *Cauchy* tensor 3.27b. The only remaining component needed to compute the $\bar{\mathbf{F}}$ will be the modified *proper orthogonal tensor* where we will take the assumption that the modified rotation tensor ($\bar{\mathbf{R}}$) equals the rotation tensor (\mathbf{R}), meaning $\bar{\mathbf{R}} = \mathbf{R}$. The computation of this $\bar{\mathbf{F}}$ is summarised in (3.27c).

$$(3.27a) \quad \mathbf{F} = \mathbf{R}\mathbf{U} \text{ thus } \bar{\mathbf{F}} = \bar{\mathbf{R}}\bar{\mathbf{U}}$$

$$(3.27b) \quad \mathbf{U} = \sqrt{\mathbf{C}} \text{ thus } \bar{\mathbf{U}} = \sqrt{\bar{\mathbf{C}}}$$

$$(3.27c) \quad \bar{\mathbf{F}} = \bar{\mathbf{R}}\bar{\mathbf{U}} \begin{cases} \mathbf{C} = \mathbf{F}^T \mathbf{F} \rightarrow \mathbf{U} = \sqrt{\mathbf{C}} \rightarrow \mathbf{R} = \mathbf{F} \cdot \mathbf{U}^{-1} \\ \bar{\mathbf{U}} = \sqrt{\bar{\mathbf{C}}} \end{cases}$$

3.5 Numerical examples

First of all, we consider as reference the results obtained with previous implementations of the element by Flores [ArtFlo13b; ArtFlo13a], as well as the results from [ArtSLL04; ArtKGW06b]. In the following, when "Ref." is displayed will refer to this; meanwhile when "Cal."⁵ is displayed the simulation will have been performed with the *in-home* implementation in *Kratos*.

3.5.1 Patch test

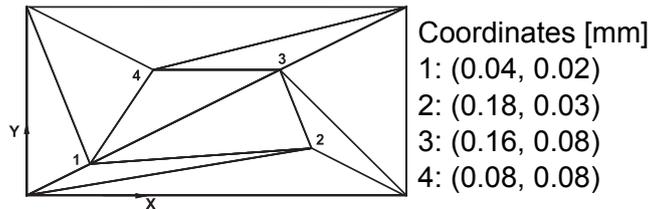


Figure 3.4: Patch test geometry

The first step in the implementation of an element into a **FEM** code is the verification of the kinematics, for this reason the patch test is understood as a necessary condition for the convergence of the element. In the case of solid elements it is expected that when nodal displacements corresponding to a constant strain gradient (membrane patch test) are imposed, constant efforts are obtained in all the elements; so in owing to the **SPRISM** is in fact a solid element this is the behaviour we should expect. In the case of a solid-shell element clearly, this must satisfy at least the membrane patch test and, although it may not be necessary, it is highly desirable that the element satisfies the bending patch test as this will lead to a more robust and reliable element.

Figure 3.4 shows a patch of elements that has been widely used to access quadrilateral shell elements and hexahedral solid shell elements, like where are working with prisms (wedges), we should split in two these elements. The size of the largest sides is $a = 0.24\text{mm}$ and the size of the shortest side is $b = 0.12\text{mm}$, while the thickness considered is $t = 0.001\text{mm}$. The lower surface has been located at coordinate $z = -t/2$. The mechanical properties of the material are: Youngs modulus $E = 10^6\text{MPa}$ and Poisson ratio $\nu = 0.25$. Because the problem considered is linear just 2 integration points across the thickness are used located in the usual Gauss quadrature positions ($\zeta = \pm 1/\sqrt{3}$).

3.5.1.1 Membrane patch test

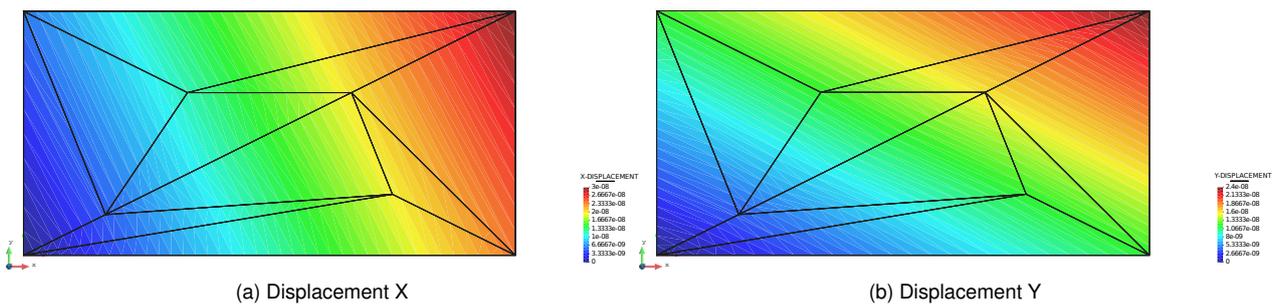


Figure 3.5: Solution for the displacement

The prescribed nodal displacements (on the boundary nodes) are defined by the linear functions (3.28), and $u_z = 0$ only on the nodes in the lower face to allow contraction due to *Poisson* effect. Due to the pure membrane strains developed in this test, the internal **DOF** α obtained is zero and constant in all the elements.

$$(3.28) \quad \begin{cases} u_x = \left(x + \frac{y}{2}\right) \cdot 10^{-3} \\ u_y = \left(y + \frac{x}{2}\right) \cdot 10^{-3} \end{cases}$$

⁵Or nothing more added.

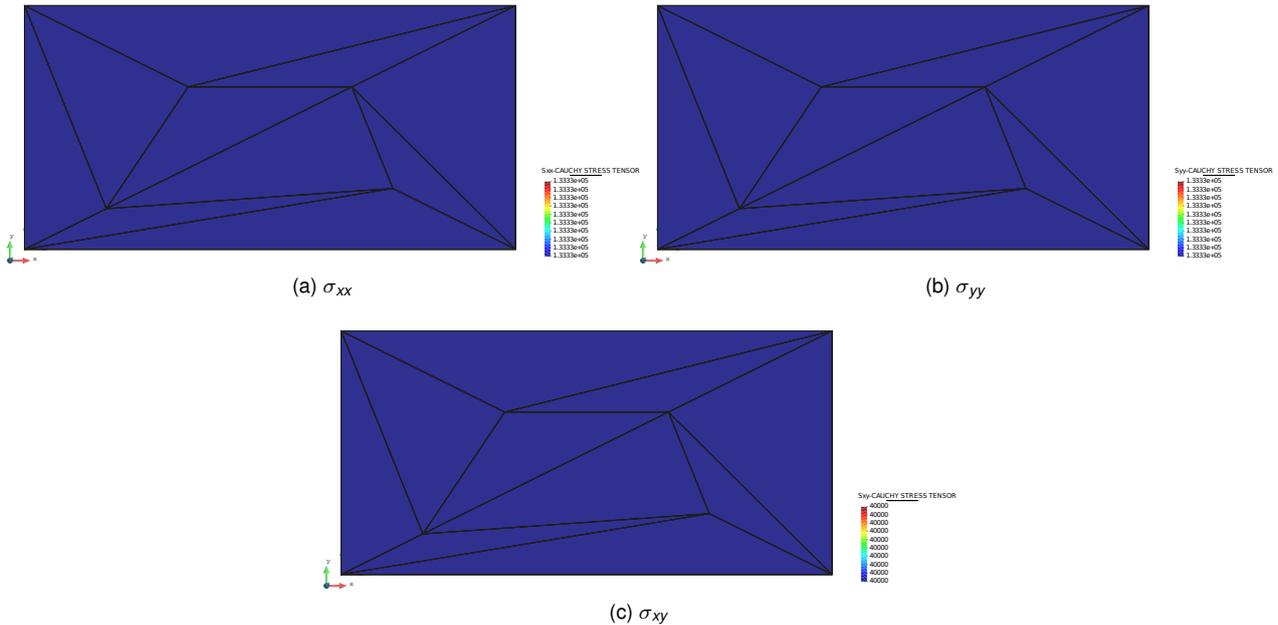


Figure 3.6: Solution for the stress

Using the present element **SPRISM** the correct results are obtained for both the displacements of the interior nodes according to (3.28) (Figure 3.5) and the element stresses ($\sigma_{xx} = \sigma_{yy} = 1333.3MPa$ and $\sigma = 400Mpa$, Figure 3.6).

3.5.1.2 Bending patch test

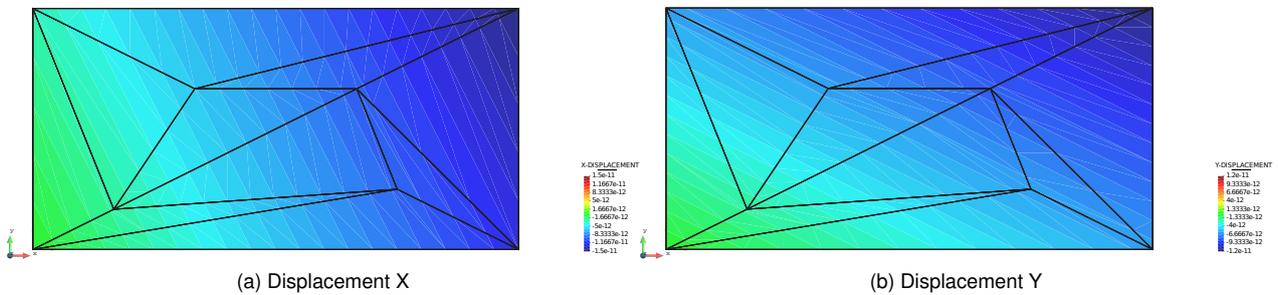


Figure 3.7: Solution for the displacement

In this case the displacement field associated with a constant bending stress state is given by (3.29), that is prescribed on the exterior nodes of both shell faces.

$$(3.29) \quad \begin{cases} u_x = \left(x + \frac{y}{2}\right) \cdot \frac{z}{2} \cdot 10^{-3} \\ u_y = \left(y + \frac{x}{2}\right) \cdot \frac{z}{2} \cdot 10^{-3} \\ u_z = \left(x^2 + xy + y^2\right) \cdot \frac{1}{2} \cdot 10^{-3} \end{cases}$$

We obtain a value for the internal **DOF** α constant in all elements and equal to: $\alpha = 0.3333 \cdot 10^{-8}$. The bending stresses at the integration points are $\sigma_{xx} = \sigma_{yy} = \pm 0.3849 Mpa^6$ and $\sigma_{xy} \pm = 0.1155 Mpa^7$ (Figure 3.8), while the displacements at the interior nodes correspond exactly with the expression (3.29) (Figure 3.7). So in consequence the element satisfies this test too.

⁶If we interpolate the values to the external faces we get $\sigma_{xx} = \sigma_{yy} = \pm 0.6666 Mpa$

⁷Interpolated to the most external faces $\sigma_{xy} \pm = 0.2000 Mpa$.

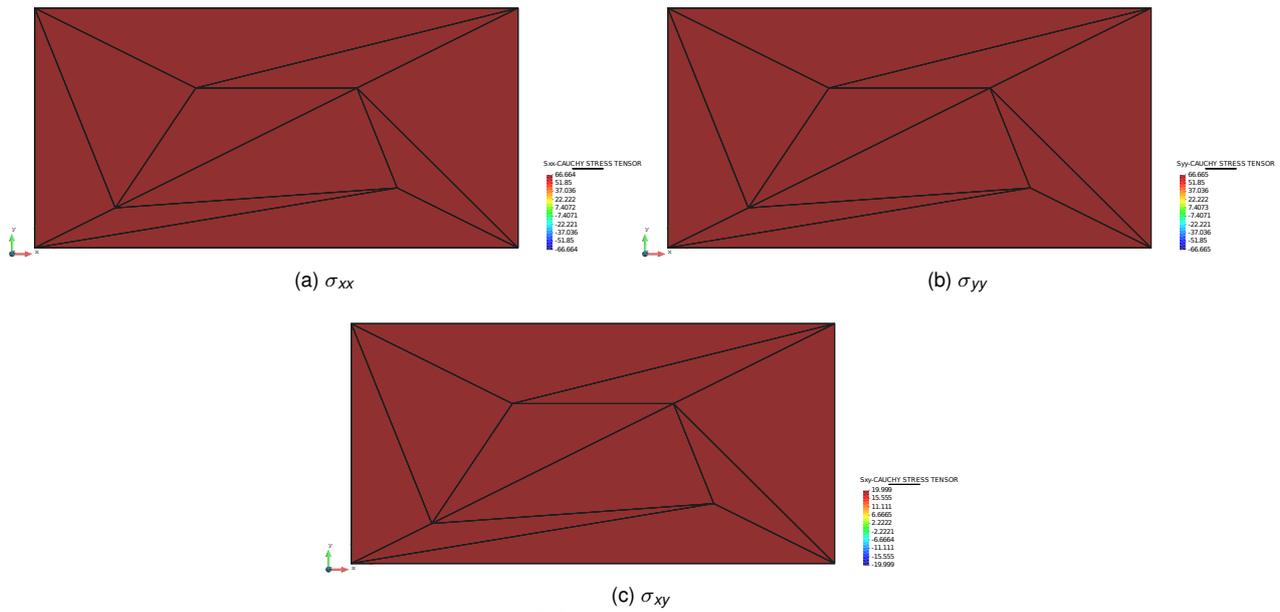


Figure 3.8: Solution for the stress

3.5.2 Cantilever

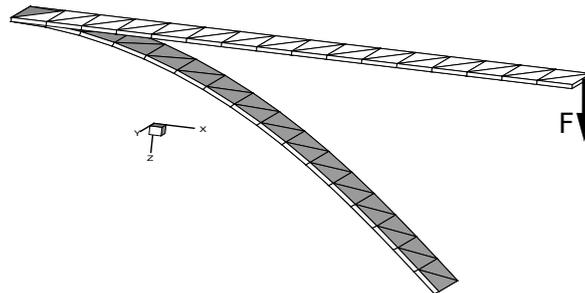


Figure 3.9: Cantilever with point load

A cantilever plate strip of length $L = 10 \text{ mm}$ width $b = 1 \text{ mm}$ and thickness $t_1 = 0.1 \text{ mm}$ is subjected to a transverse load $F = 40 \text{ N}$ (Figure 3.9). For the selected Youngs modulus $E = 106 \text{ MPa}$ the behaviour is one with large displacements but small strains. Using different values of Poisson ratio ($\nu = 0.0$, $\nu = 0.3$, $\nu = 0.49$, $\nu = 0.499$ and $\nu = 0.4999$ (quasi-incompressible)) it can be assessed if the proposed assumed strain techniques allow to avoid respectively the **transverse shear locking**, the **Poisson effect locking** and the **volumetric locking**. At the same time, different geometries, with different numbers of divisions in length have been considered, see Figure 3.10, one in the width and one across the thickness with two integration points. The final deformed configurations (vertical displacement is 70% of the length) is achieved considering small load steps.

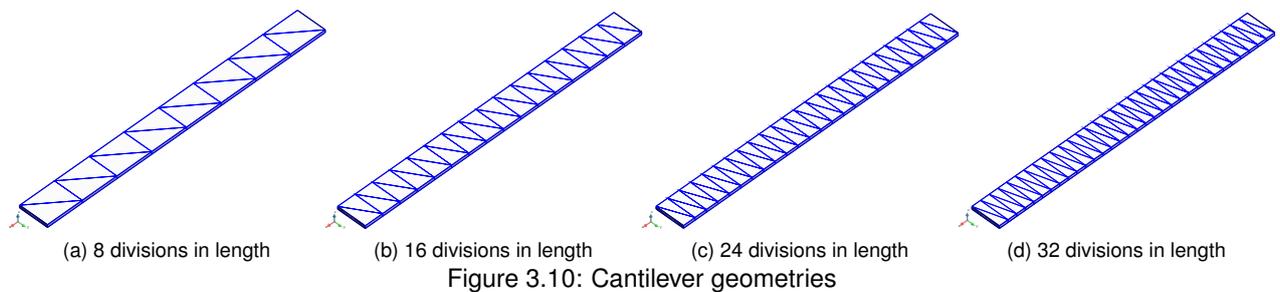


Figure 3.10: Cantilever geometries

The Figure 3.11b shows the maximum vertical displacement versus the load factor (from 0 to 1) for 5 different values of the Poisson ratio, for the geometry of the 32 divisions in length. The case $\nu = 0$ allows to compare with the

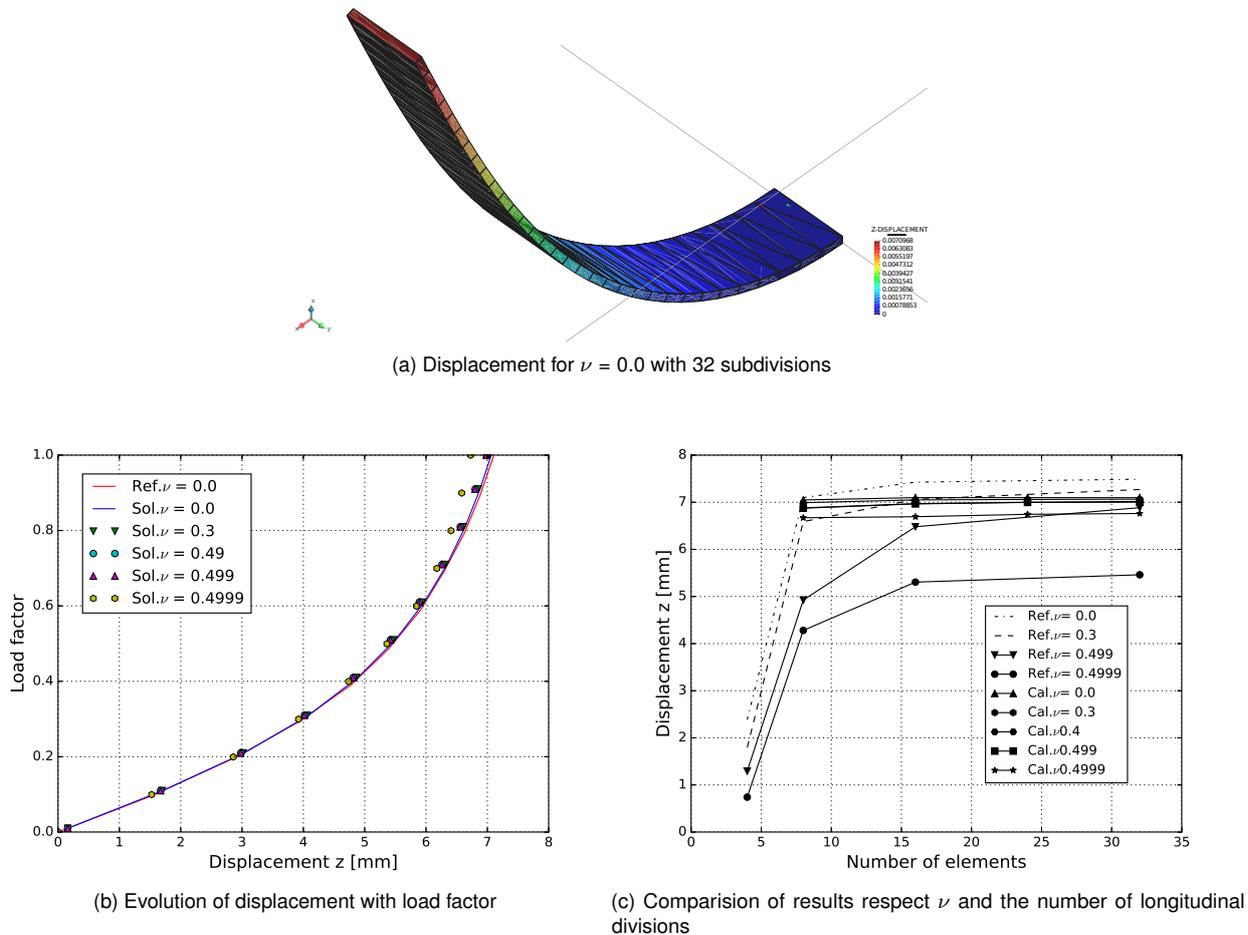


Figure 3.11: Solution

reference value ($u_z = 7.08mm$) and to see if the approach used to cure transverse shear locking is adequate. The result obtained $u_z = 7.06mm$ indicates that effectively the element is free of **transverse shear locking**. The second value of *Poisson* ratio ($\nu = 0.30$) is used to assess if the **EAS** technique avoids the appearance of **locking due to Poissons effect**. In this case the computed displacement is $u_z = 7.03mm$ that although it is not exactly the same value obtained for $\nu = 0$ shows that the proposed method avoids the **Poissons effect locking** allowing a proper gradation of the transverse normal strain. Finally, the last three values of *Poisson* ratio (0.49,0.499 and 0.4999) allow to observe if the performance of the element deteriorates significantly in the *quasi-incompressible* range. It can be seen that although differences grow with *Poisson* ratio, this is below 4% for the higher value considered.

Besides, Figure 3.11c plots the tip displacement as a function of the mesh density (number of divisions along the length) for four different *Poissons* ratio. In the reference results obtained by *Fernando G. Flores* it can be seen that convergence deteriorates for *Poissons* ratio larger than 0.499; as it happens in our *in-home* implementation, but slightly less significant in this case.

3.5.3 Cantilever subjected to end bending moment

Figure 3.12 shows a cantilever subjected to end moment M . The problem has been considered in from the reference[ArtSLL04]. The cantilever forms a circular arc with its radius R given by the classical flexural formula $R = EI = M$. Using the formula, the analytical normalised deflections can be derived to be (3.30) where $M_0 = EI = L$. The maximum end moment M_{max} is taken to be M_0 at which the beam will be bent into a circle.

$$(3.30) \quad \frac{U}{L} = \frac{M_0}{M} \sin \left(\frac{M}{M_0} - 1 \right), \quad \frac{W}{L} = \frac{M_0}{M} \left(1 - \cos \left(\frac{M}{M_0} \right) \right)$$

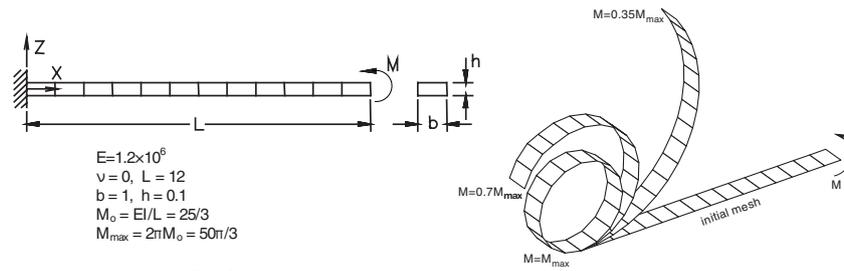
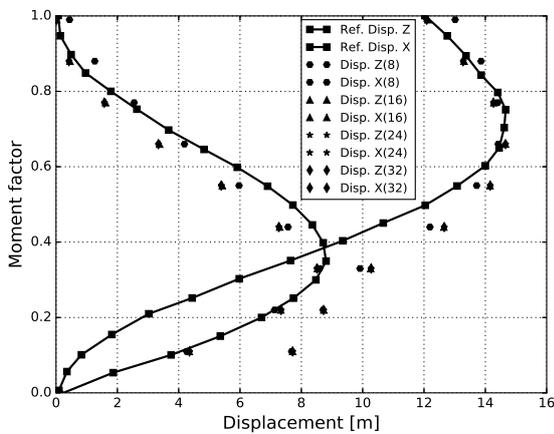
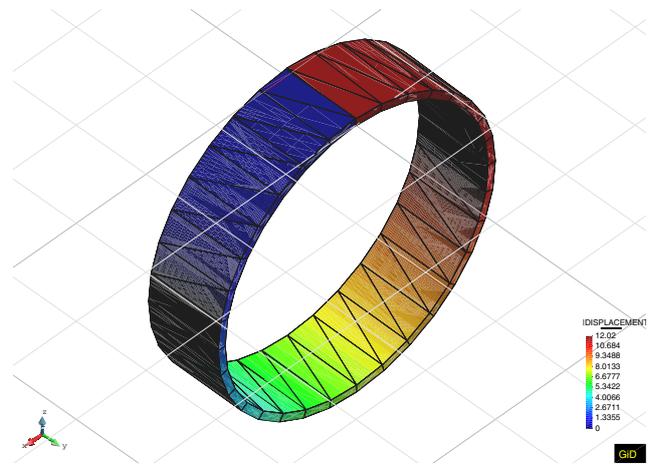


Figure 3.12: Cantilever subjected to end bending moment

Figure 3.13a plots the end moment against the vertical and horizontal tip deflections for different configurations of the geometry (8, 16 and 32 subdivisions) and Figure 3.13b portrays the deformed cantilever at $M = M_{max}$.



(a) Initial geometry



(b) Deformed geometry

Figure 3.13: Solution for a cantilever subjected to an end bending moment

3.5.4 Frequencies test

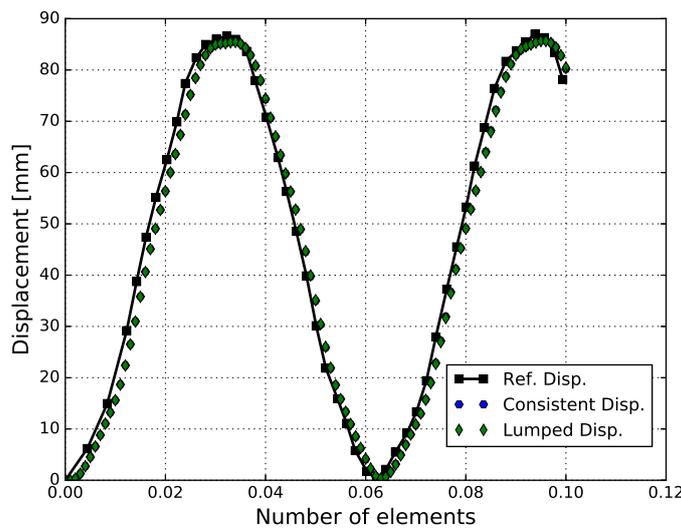


Figure 3.14: Frequencies behaviour test

This example (extracted from [ArtFlo13a; ArtOUS04]) considers the dynamic behaviour of a cantilever beam with length, width and thickness $L = 1$; $b = 0.1$ and $t = 0.01$ respectively. The mechanical properties are Young's modulus

$E = 100\text{GPa}$, *Poissons* ratio $\mu = 0$ and mass density $\rho = 1000\text{kg/m}^3$. The point load applied at the free side (Figure 3.9) has a value of 100N with a *Heaviside* step time function. As the problem is elastic with $\mu = 0$ there is no *Poissons* effect across the thickness nor volumetric locking. The behaviour is purely bending and it is useful to evaluate the shear locking and assess the proposed cure. The discretisation includes eight uniform divisions along the length, one in the width and one element through the thickness. The solution shown in Figure 3.14 a very close behaviour to the reference[ArtFlo13a], both for lumped and consistent mass matrix.

3.5.5 Cook membrane

This example (see Figure 3.15) involves a large amount of shear energy and is commonly used to assess in-plane bending performance.

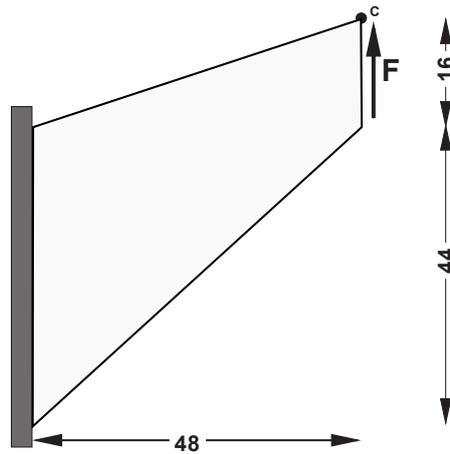
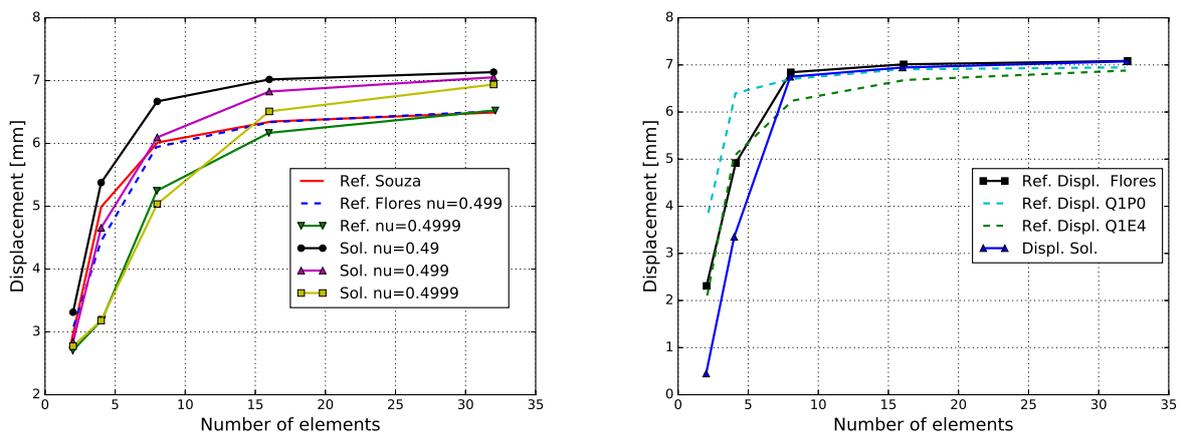


Figure 3.15: Cook's membrane geometry

Plane strain condition will be considered here with two different material behaviour: (a) a quasi-incompressible elastic material with $G = 80.1938\text{GPa}$ and $K = 40.1 \cdot 10^4\text{GPa}$ corresponding with a *Poisson* ratio $\mu = 0.4999$ and (b) an elasticplastic material with elastic properties $G = 80.1938\text{GPa}$ and $K = 164.21\text{GPa}$ implying a *Poisson* ratio $\mu = 0.29$ and J_2 plasticity with isotropic hardening as a function of the effective plastic strain e^p defined by (3.31) and the values of the Table 3.3.



(a) Displacement versus number of elements for the elastic case (b) Displacement versus number of elements for the elastic-plastic case

Figure 3.16: Cook's membrane convergence solution

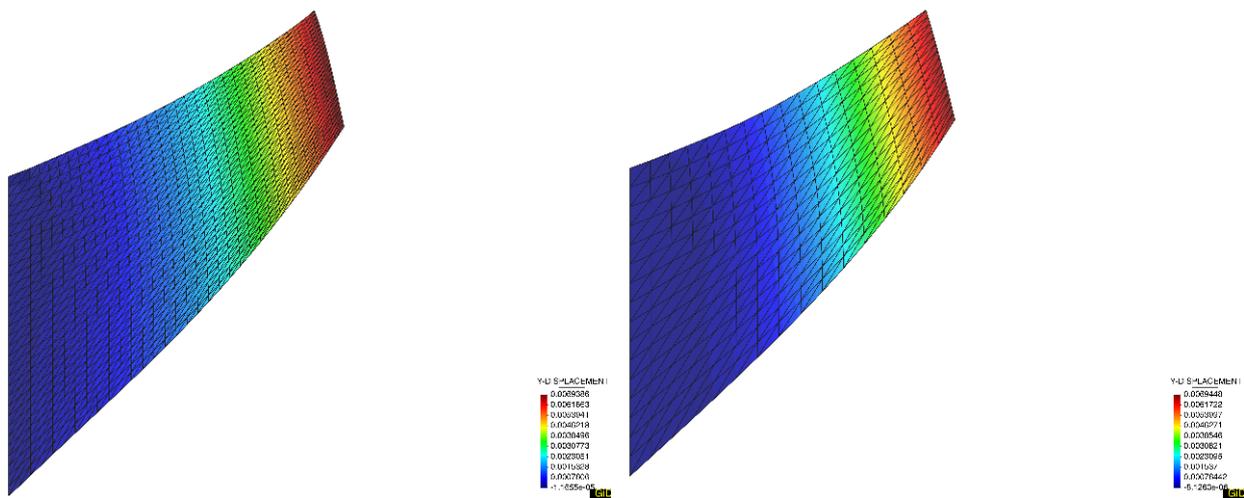
(3.31a) $\sigma_y^* = \text{Linear hardening} + \text{Exponential hardening}$

$$(3.31b) \quad \begin{cases} \text{Linear hardening} = \sigma_y + e^p \cdot \theta \cdot h \\ \text{Exponential hardening} = (\sigma_0 - \sigma_\infty) \cdot (1 - e^{-\eta \cdot e^p}) \end{cases}$$

Yielded stress σ_y	0.45 GPa	Kinematic hardening modulus h	0.12924 GPa
Reference hardening modulus σ_0	0.45 GPa	Infinity hardening modulus σ_∞	0.715 GPa
Hardening exponent η	16.93	Pure isotropic hardening θ	1

Table 3.3: Isotropic-kinematic hardening law constants

The applied load is 100kN for the elastic case and 5kN for the elasticplastic material. The plane strain condition implies coefficient $C_{33} = 1$ at all points ($\alpha = 0$), thus the version without **ANS** for the in-plane components locks due to the almost incompressibility constraint in the same way that a constant strain triangle does. Because of that this example is intended to assess how the improvement in the membrane field collaborates to cure the volumetric locking.



(a) Vertical displacement for the elastic case (32 elements $\nu = 0.4999$) (b) Vertical displacement for the elastic-plastic case(16 elements)
Figure 3.17: Cook's membrane deformed shape solution

Figures 3.16a and 3.16b show a convergence analysis as the mesh is refined, where the vertical displacement of the point **C** has been plotted versus the number of divisions per side. In Figures 3.17a and 3.17b the deformed shape for the elastic and elastic-plastic cases is shown. The results have been compared with the reference [ArtFlo13b], and other results presented in the same article, where the results that we have obtained present a very good convergence in comparison.

3.5.6 Scoordelis cylindrical roof test

In this example we proceed with a linear analysis of a cylindrical shell under self-weight, which is free along the one side as can be seen in the Figure 3.18 and it has been simplified in one quarter of the original size with additional consideration of the symmetry conditions of the problem. We have considered five different geometric configurations, with a different number of elements across the side, until the convergence of the solution have been archived. As we have mentioned previously, the problem is linear, therefore it is possible to solve the problem considering just two Gauss points across the thickness.

The problem is membrane dominant, that is why could be interesting to observe the relevance of the **ANS** for in-plane components in non-isochoric problems. The results obtained are shown in the Figure 3.19, where in Figure 3.19b the results are compared with our reference[ArtFlo13a], having a very close convergence to the reference solution. Besides the Figure 3.19a shows the deformed shape obtained for the maximum number of elements.

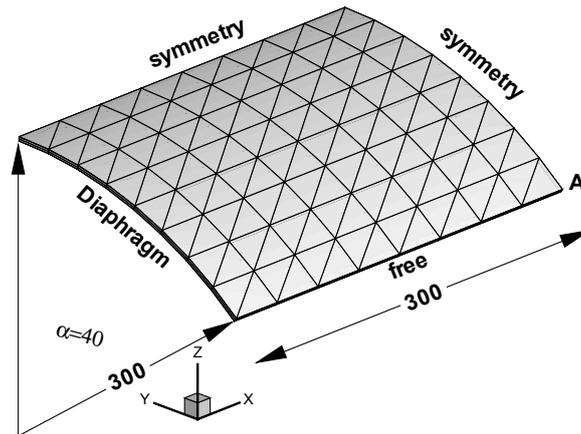
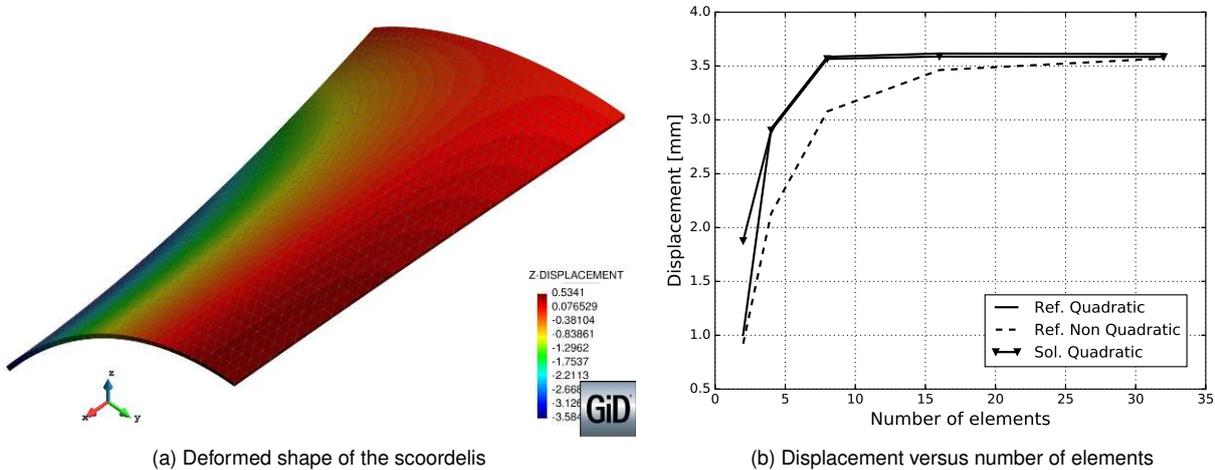


Figure 3.18: Geometry of the scoordelis cylindrical roof



(a) Deformed shape of the scoordelis (b) Displacement versus number of elements
Figure 3.19: Solution for the scoordelis cylindrical roof

3.5.7 Sphere test

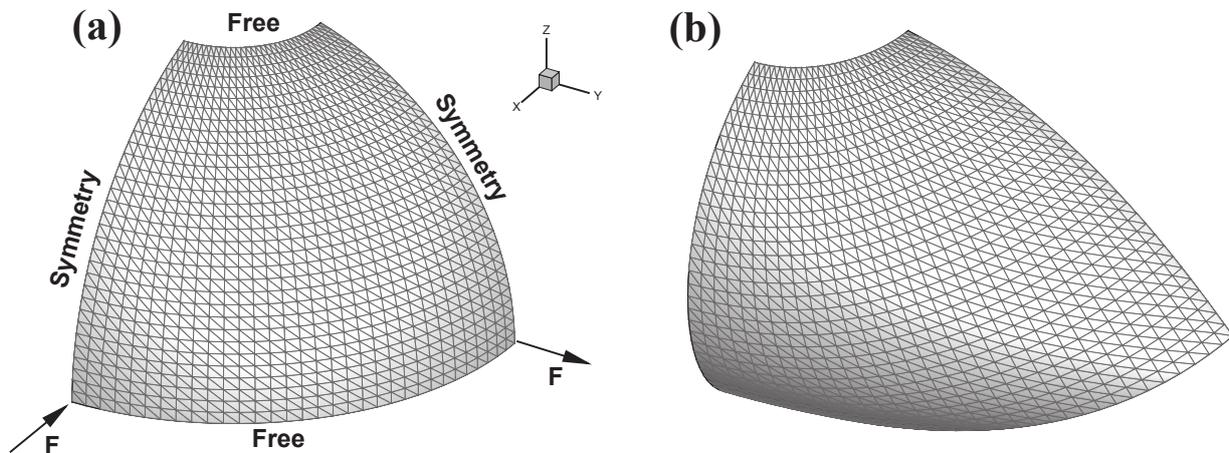
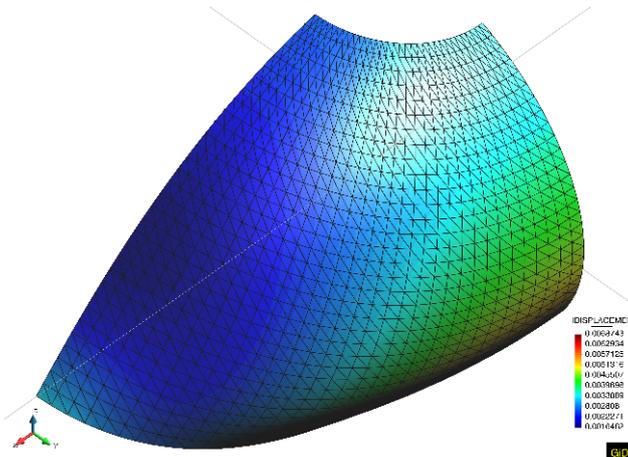


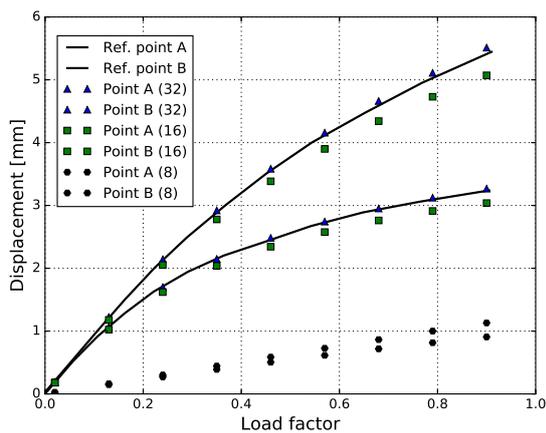
Figure 3.20: Geometry of a semi-spherical shell with a hole, both original and deformed geometry

Some interesting kind of problem to check the suitability of the element is the problems where an initially double curved geometry. This problem (Figure 3.20) is analysed and solved recurrently in the context of large elastic displacements. The Figure 3.20 presents the geometry considered in the resolution of the problem, where once

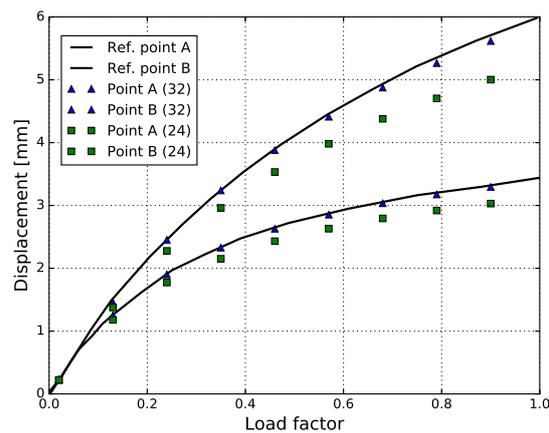
again the symmetry has been considered to simplify the resolution of the problem. This kind of problem is mainly an inextensional bending problem where *Poisson* effect has an important role in the behaviour of the structure, in contrast with the membrane effect that is less significant in this problem, on the other hand, the membrane locking and the curvature-thickness locking could appear.



(a) Deformed shape of the semi-spherical shell



(b) Displacement versus load for $R/t = 250$



(c) Displacement versus load for $R/t = 1000$

Figure 3.21: Geometry of a semi-spherical shell with a hole solution

Several meshes have been considered, with 8, 16, 24 and 32 elements by side respectively, with a middle radius of $R = 10\text{mm}$ and thickness of $t = 0.04\text{mm}$ ($R/t = 250$). The coarser is the element, the more it could suffer the locking effect due to the initial curvature, considering this when solutions more differ more than the 5% from the target values. We are considering the following mechanical properties, $E = 6.825 \cdot 10^4\text{GPa}$ and $\nu = 0.3$. See Figure 3.21.

3.5.8 Pull-out of an open-ended cylindrical shell

Figure 3.22 shows an open-ended cylinder being pulled by a pair of radial forces P . The problem has been considered in the Reference [ArtSLL04], among others. Owing to symmetry, just one-eighth of the shell is modelled.

The Figure 3.23a presents the convergence of the solution considering a different number of elements, having a converged solution very close to the reference one[ArtSLL04]. The final deformation of the one-eighth is shown in Figure 3.23b, showing that the problem involves very large displacements.

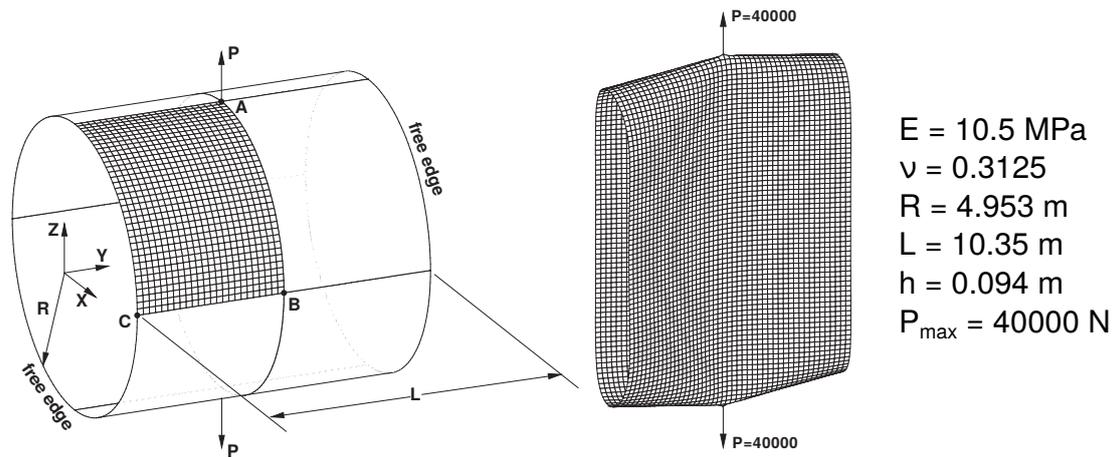
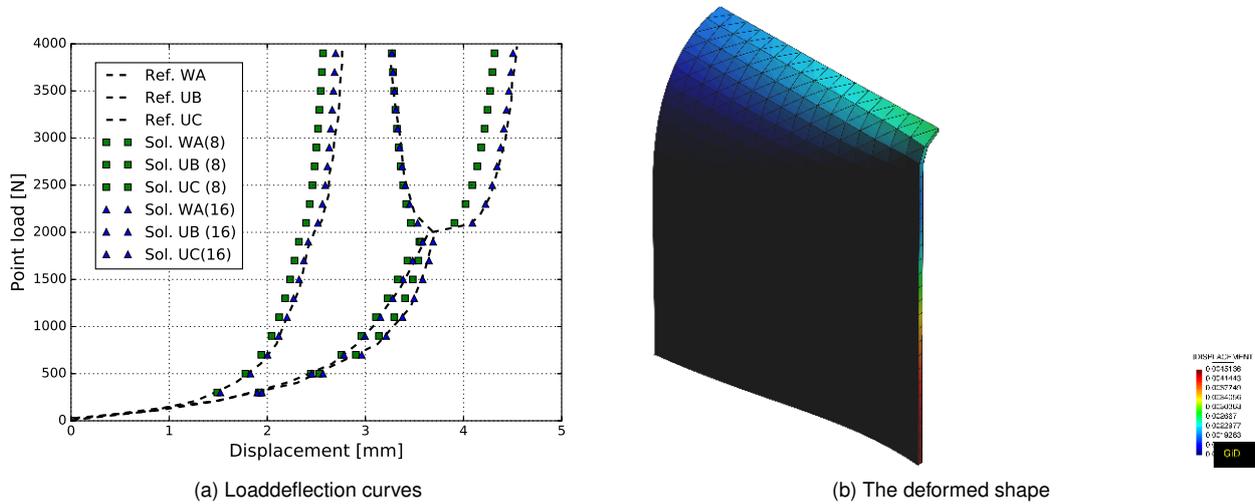


Figure 3.22: The open-end cylindrical shell subjected to radial pulling forces.



(a) Loaddeflection curves

(b) The deformed shape

Figure 3.23: Solution of the open-ended cylindrical shell

3.5.9 Pinched semi-cylindrical

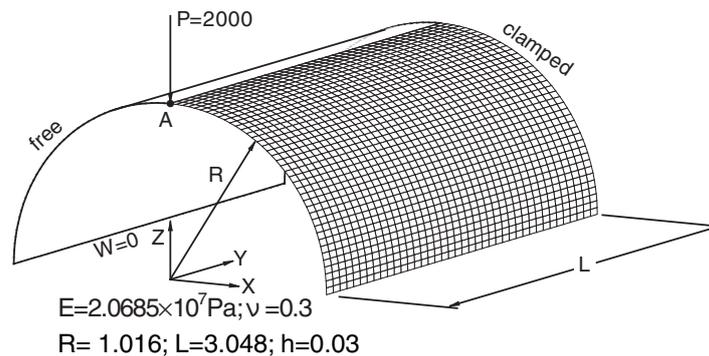
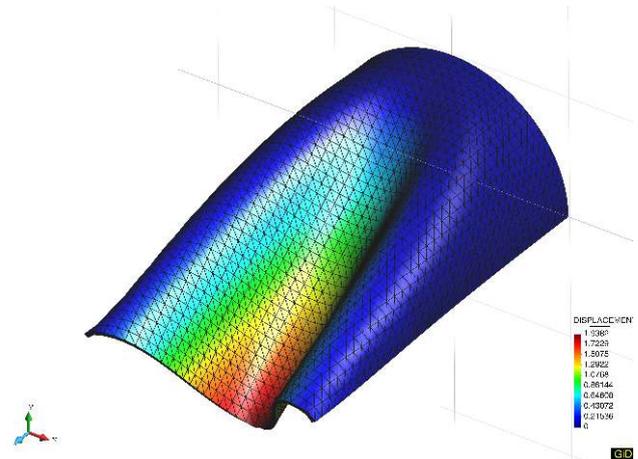
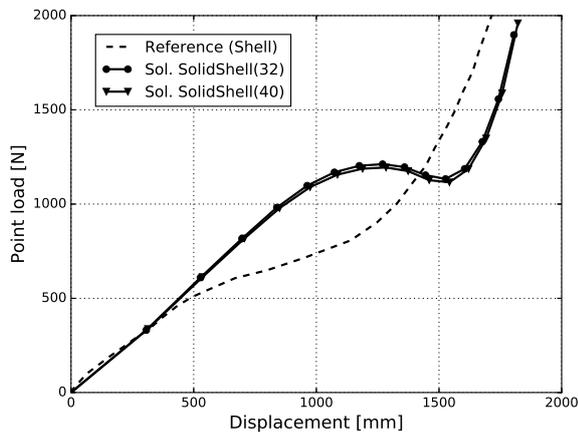


Figure 3.24: The semi-cylindrical shell subjected to an end pinching force

Figure 3.24 shows the semi-cylindrical shell [ArtSLL04] subjected to an end pinching force at the middle of the free-hanging circumferential periphery. The other circumferential periphery is fully clamped. Along its longitudinal edges, the vertical deflection and the rotation about the Y -axis are restrained.

The solution obtained is shown in Figure 3.25, where the Figure 3.25a presents the load-deflection curve, the behaviour in our obtained solution is slightly different from the one presented in the reference [ArtSLL04], this can be

due to the fact that we are taking the displacement in the point where the load is applied, meanwhile the reference solutions corresponds to a shell and the solution corresponds with the displacement in the middle surface; a possible way to improve our solution way to tackle this problematic is adding an additional layer and plot the solution in the point belonging to the middle surface. Figure 3.25b presents the deformed shape obtained for the maximum value of the load.



(a) Loaddeflection curves

(b) The deformed shape

Figure 3.25: Solution of the semi-cylindrical shell

3.5.10 Slit test

The Figure 3.26a shows the original geometry of an annular plate and it presents a large displacement (Figure 3.26b) due to a load applied in one face while the other face is constrained. This problem is a common *benchmark* considered to study the behaviour of shells under large rotations, in our case, as we are considering a **solid-shell** we do not have rotations in our element and it could be considered as large displacement problems.

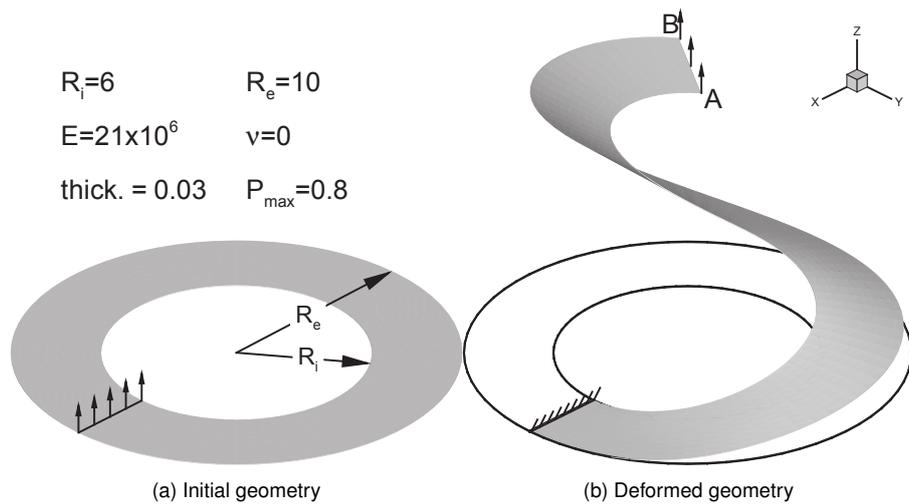


Figure 3.26: Slit annular plate

The solution looks like the solution expected Figure 3.27b. A comparison of the results for the coarse mesh (Figure 3.27a) allows to observe the influence of the **ANS** for membrane part that for the maximum load factor indicate a difference in displacements larger than 4%. The results for the fine mesh are in excellent agreement with those provided in[ArtFlo13b].

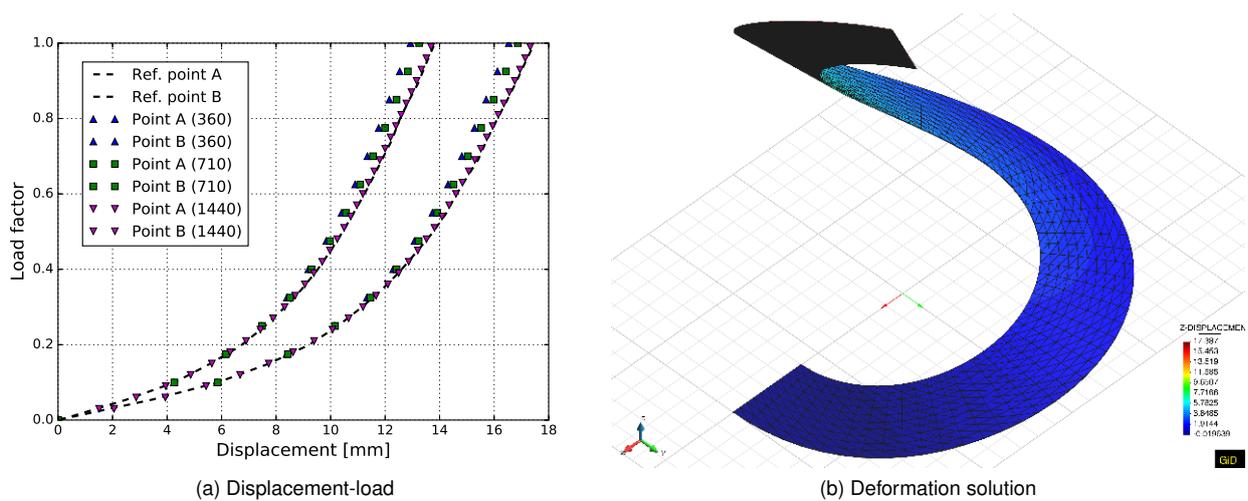


Figure 3.27: Slit annular plate solution

3.5.11 Cylindrical panel test

This example [ArtSLL04; ArtFlo13b] considers a rectangular cylindrical panel simple supported along the straight sides and free along the curved sides, that is subjected to a vertical point load in its centre (see Figure 3.28). The middle surface geometry is defined by the length of the panel $L = 508\text{mm}$, the radius of the cylinder $R = 2540\text{mm}$ and the half angle $\theta = 0.1\text{rad}$. The behaviour of the panel presents a limit point, followed by a strong loss of strength and a final stiffening once the curvature is inverted. Two different thicknesses for the same mid-surface geometry have been considered $t = 12.7\text{mm}$ and $t = 6.35\text{mm}$ that for the thin case leads to a snap back of the loaded point. This example has been widely used to assess the performance of shell elements and non-linear path-following techniques, like the arc-length.

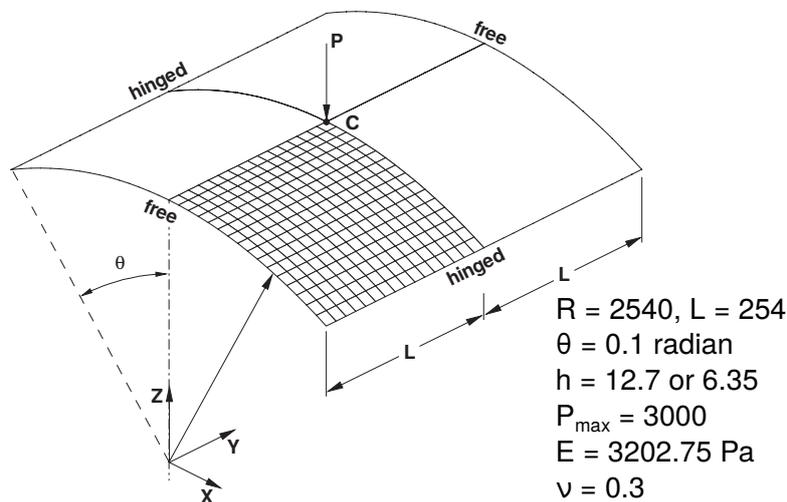
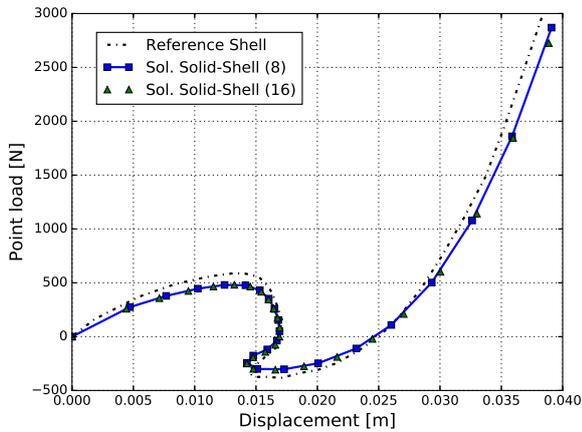


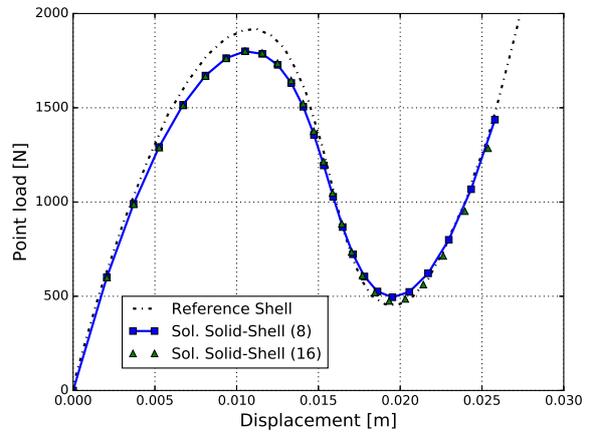
Figure 3.28: Geometry of the hinged cylindrical panel

For this problem two meshes have been considered with 8 and 16 elements per side. In this case 2 elements in the thickness direction have been used that allows to introduce the hinge in the middle surface and then to compare with solutions obtained with shell elements. The vertical displacement of the loaded point **C**.

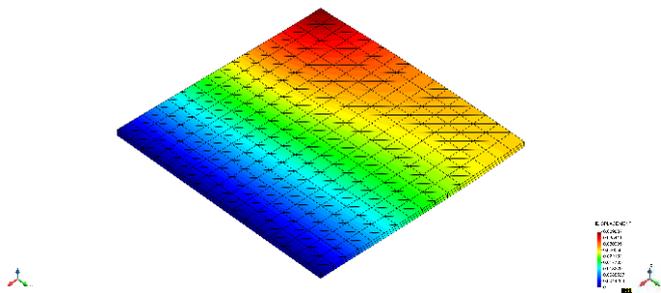
Figures 3.29a and 3.29b show the evolution of the load-deflection compared with the reference [ArtSLL04], besides the Figures 3.29c and 3.29d present the deformed shape of the problem.



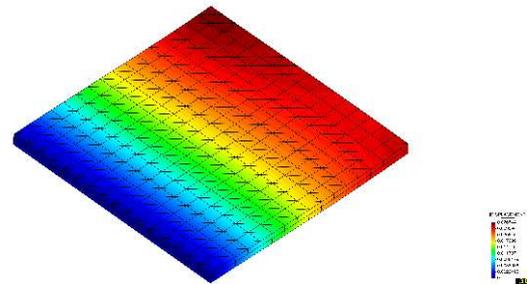
(a) Thin geometry displacement-load



(b) Thick geometry displacement-load



(c) Thin geometry displacement



(d) Thick geometry displacement

Figure 3.29: Solution of the cylindrical panel

3.5.12 Conical shell test

This example is selected to demonstrate the ability of the developed finite element to deal with strongly non-linear situations. The geometrical data are taken from [ArtKGW06b]. Here, elasticplastic material behaviour is assumed. All necessary material and geometrical data are depicted in Figure 3.30. The non-linear behaviour is computed using an arc-length algorithm with displacement control. The constitutive law considered is a J_2 hyperelastic-plastic model.

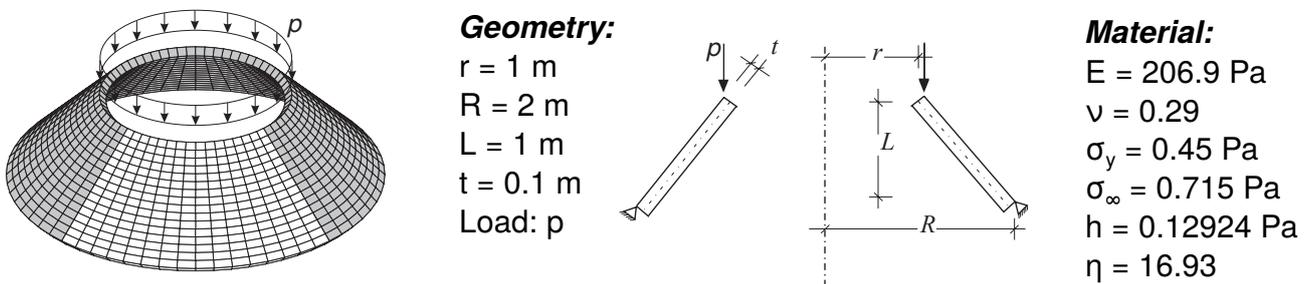


Figure 3.30: Geometry of the cone shell

The results are depicted in Figure 3.31a, where w denotes the vertical displacement of the upper edge. The load deflection diagram demonstrates that our result is close in the order of magnitude to the reference, thus some improvements are needed in the modelisation, in special the relative to the relative to the arc-length strategy. The Figures 3.31b and 3.31c present the deformation and plastic strain in the last stages of the simulation.

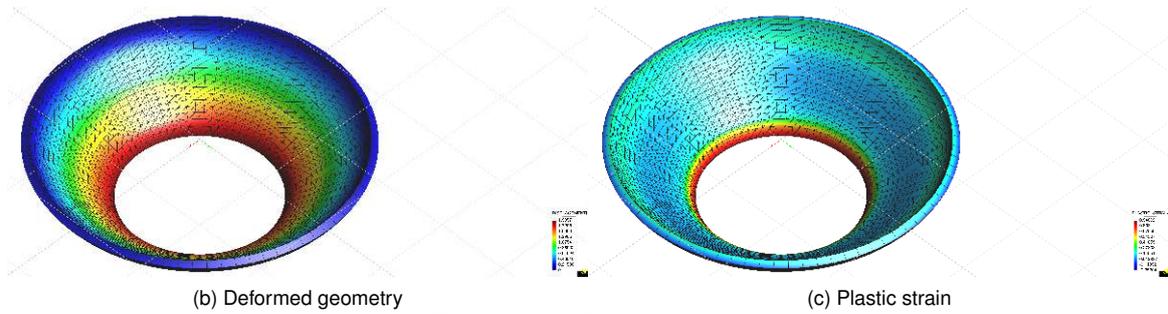
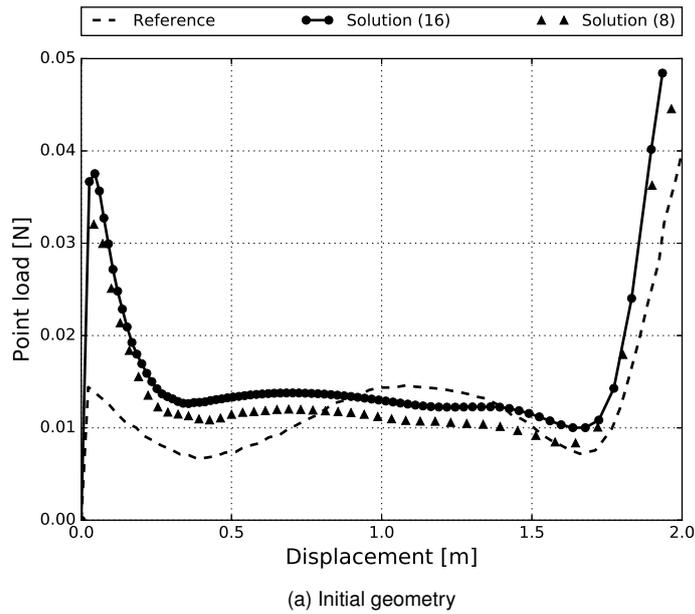


Figure 3.31: Cone-shell test

3.5.13 Wrinkling test

The problem consists [ArtTSW05; ArtFlo13a] of a square membrane (see Figure 3.32) with a side $a = 229\text{mm}$ made of a thin film of *Mylar* with a thickness $t = 0.0762\text{mm}$. The *Mylar* mechanical properties are $E = 3790\text{MPa}$ and $\mu = 0.38$. The top and bottom edges are clamped and the lateral edges are free. The top edge is subjected to a uniform horizontal displacement $\Delta = 1\text{mm}$ along the edge.

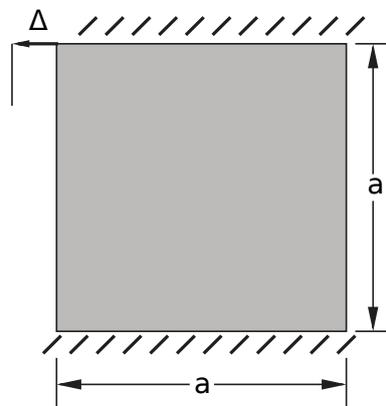
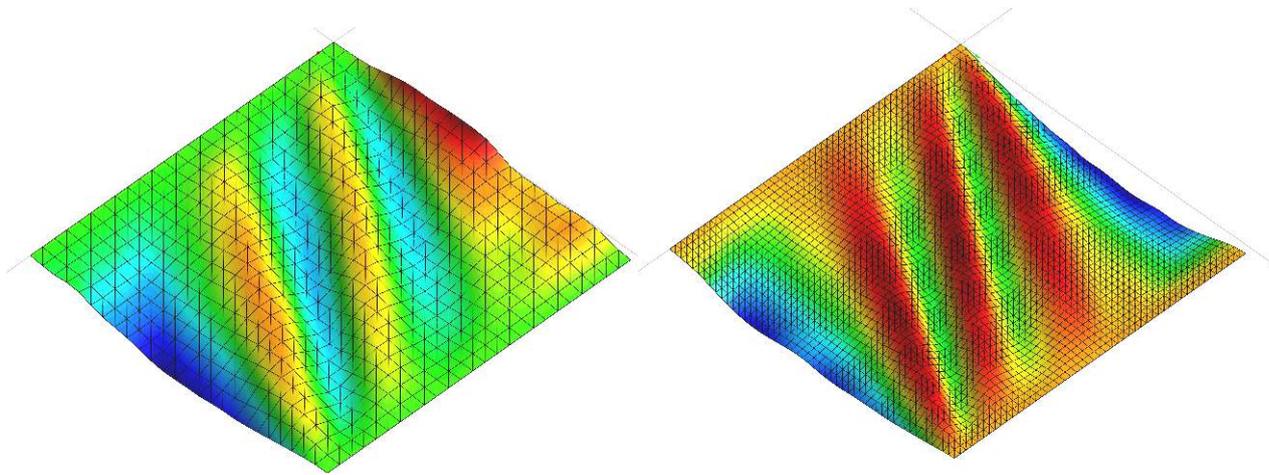


Figure 3.32: Square thin film under in-plane shear

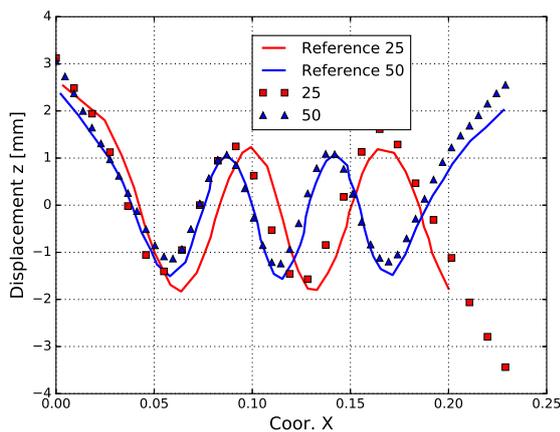
Two uniform structured meshes with $26 - 26$ and $51 - 51$ nodes, with 1250 and 5000 elements respectively have been considered. Figure 3.33 plots two out-of-plane displacement profiles along the centre of the square in

both Cartesian directions, as well as the deformation obtained. The solution that we obtain are very close to those obtained in the reference [ArtFlo13a].

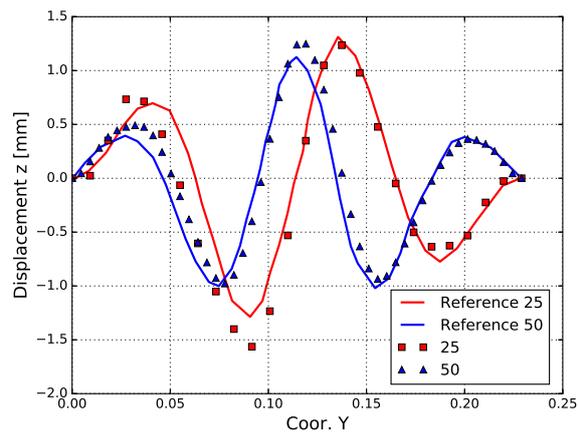


(a) Deformation with 25 elements per side

(b) Deformation with 50 elements per side



(c) Transverse displacement profiles along the centre of the square: $y = a/2$



(d) Transverse displacement profiles along the centre of the square: $x = a/2$

Figure 3.33: Solution of the wrinkling simulation

3.5.14 FSI-Vein test

An additional test that has been performed to check the robustness of the element, which consists in a simple **FSI** simulation of an elastic vein. This element has clear advantages from the common approaches performed nowadays, which consist normally in the consideration of multiple layers of solid elements (usually hexahedron), despite the vein is under *“shell stresses”*. This owing to the need of having a proper and correct definition of the irregular geometry and thickness that a real vein can show, besides a correct definition of the interface for a proper **FSI** simulation, which cannot be correctly modelled with a conventional shell element. So this element has advantages of the solid element, and like the element is clearly created to compute *“shell stresses”* there does not need to consider additional layers of solid to enrich the behaviour of the solid.

This problem was originally proposed in [ArtNob01] and later reproduced in [PhDVMO09]. Its aim is to simulate the **FSI** arising in the modelling of blood flow in the human cardiovascular system. As described in [PhDVMO09], the problem consists of a thin elastic vessel, which in this case has been modelled with the current element considering an hyperelastic constitutive law, conveying the blood flow, which is modelled as an incompressible fluid using the *Navier-Stokes* equations.

Regarding the geometry, it consists in a straight cylinder of radius $r_0 = 0.005m$ which length and thickness are $L = 0.05m$ and $t = 0.001m$. The blood physical parameters are $\rho_f = 1000kg/m^3$ and dynamic viscosity $\mu_f = 0.003kg/ms$, yielding a kinematic viscosity $\nu_f = 3e - 06m^2/s$. Regarding the solid parameters, the density is $\rho_s = 1200kg/m^3$ while the *Poisson* ratio and *Young* modulus are $\nu_s = 0.3$ and $E = 3e05Pa$. Regarding the boundary conditions, both sides of the vein are clamped (radial displacements allowed) and an overpressure of $p = 1333.2Pa$ (see (3.32)) is imposed at the inlet boundary for $3ms$.

$$(3.32) \quad p = \begin{cases} 1333.2 \sin(2\pi t) & \text{if } t \leq 0.25ms \\ 1333.2 & \text{if } 0.00025 < t \leq 0.275ms \\ 1333.2(1 - \sin(2\pi(t - 0.275))) & \text{if } 0.275ms < t \leq 0.3ms \\ 0.0 & \text{otherwise} \end{cases}$$

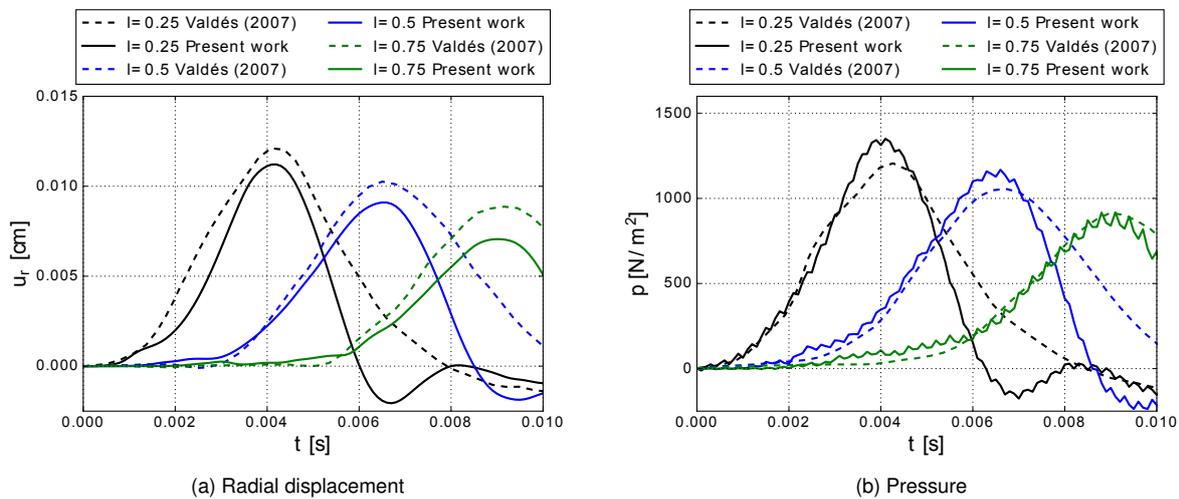


Figure 3.34: Solution of the **FSI-Vein** test

Figures 3.34a and 3.34b collect a comparison between the results in [PhDVM09] and the obtained ones for three control points placed at $0.25l$, $0.5l$ and $0.75l$, being l the tube length. Regarding the radial displacements (Figure 3.34a), it can be seen that the obtained results are similar to the reference ones. The major differences appear after the peak value when the vein section is recovering its shape. Besides, this vein retraction is much clear in the presented solution and can be clearly noted by the negative radial displacements. This behaviour is more similar to real hemodynamics and has been also observed in similar problems in the literature [PhDCal06].

On the other hand, the pressure evolution is also assessed in Figure 3.34b. As can be noted, the pressure trend matches the radial displacement evolution but some oscillations appear in the solution. Regarding the nature of these oscillations, it can be asserted that they are not numeric, since one oscillation is developed in several time steps, and this is done due to the fact that any non-reflecting boundary has been considered.

Bibliography

Books

- [BookBD12] *Sophie Germain: An essay in the history of the theory of elasticity.* Louis L Bucciarelli and Nancy Dworsky. Springer Science & Business Media. 2012.
- [BookBD80] *Sophie Germain: An Essay in the History of the Theory of Elasticity.* Louis L.. Bucciarelli and Nancy Dworsky. D. Reidel Publishing Company. 1980.
- [BookBD90] *Modélisation des structures par éléments finis: Solides élastiques.* Jean-Louis Batoz and Gouri Dhatt. Presses Université Laval. 1990.
- [BookBel+14] *Nonlinear Finite Elements for Continua and Structures.* Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. Wiley. 2nd ed. 2014.
- [BookBor+12] *Nonlinear Finite Element Analysis of Solids and Structures, 2nd edition.* René de Borst, Mike A. Crisfield, Joris J. C. Remmers, and Clemens V. Verhoosel. Wiley. 2nd Edition. 2012.
- [BookCal89] *Theory of shell structures.* Chris R Calladine. Cambridge university press. 1989.
- [BookFHK02] *Handbook of computer aided geometric design.* Gerald Farin, Josef Hoschek, and M-S Kim. Elsevier. 2002.
- [BookGou12] *Analysis of shells and plates.* Phillip L Gould. Springer Science & Business Media. 2012.
- [BookHBC09] *Isogeometric Analysis: Toward Integration of CAD and FEA.* T.J.R. Hughes, Yuri Bazilevs, and J. Austin Cottrell. Wiley. 1st ed. 2009.
- [BookHug+14] *Computer graphics: principles and practice.* John F Hughes, Andries Van Dam, James D Foley, Morgan McGuire, Steven K Feiner, and David F Sklar. Pearson Education. 2014.
- [BookKra67] *Thin elastic shells: an introduction to the theoretical foundations and the analysis of their static and dynamic behavior.* Harry Kraus. John Wiley & Sons. 1967.
- [BookNov59] *The theory of thin shells.* Valentin Valentinovich Novozhilov. P. Noordhoff. 1959.
- [BookOña13] *Structural Analysis with the Finite Element Method Linear Statics: Volume 2. Beams, Plates and Shells.* Eugenio Oñate. Springer Netherlands. Jointly published with CIMNE, Barcelona, Spain2013. 2013.
- [BookPT97] *The NURBS book: monograph in visual communication, 1997.* LA Piegl and W Tiller. Springer, New York. 1997.
- [BookRog00] *An introduction to NURBS: with historical perspective.* David F Rogers. Elsevier. 2000.
- [BookTW59] *Theory of plates and shells.* Stephen P Timoshenko and Sergius Woinowsky-Krieger. McGraw-hill. 1959.
- [BookWem82] *Mechanics of solids with applications to thin bodies.* Gerald Wempner. Springer Science & Business Media. 1982.
- [BookWri08] *Nonlinear finite element methods.* Peter Wriggers. Springer. 1st ed. 2008.

Articles

- [ArtAC09] “An improved assumed strain solid–shell element formulation with physical stabilization for geometric non-linear applications and elastic–plastic stability analysis”. Farid Abed-Meraim and Alain Combescure. In: *International Journal for Numerical Methods in Engineering*. No. 13, Vol. 80, 2009, pp. 1640–1686. Wiley Online Library.
- [ArtAIZ70] “Analysis of thick and thin shell structures by curved finite elements”. Sohrabuddin Ahmad, Bruce M. Irons, and O. C. Zienkiewicz. In: *International Journal for Numerical Methods in Engineering*. No. 3, Vol. 2, 1970, pp. 419–451. DOI: [10.1002/nme.1620020310](https://doi.org/10.1002/nme.1620020310).
- [ArtBen+10] “Isogeometric shell analysis: The Reissner–Mindlin shell”. D.J. Benson, Y. Bazilevs, M.C. Hsu, and T.J.R. Hughes. In: *Computer Methods in Applied Mechanics and Engineering*. No. 5–8, Vol. 199, 2010, pp. 276–289. DOI: [10.1016/j.cma.2009.05.011](https://doi.org/10.1016/j.cma.2009.05.011).
- [ArtBR92] “Comparison of Shell Theory and Degeneration”. N Büchter and E Ramm. In: . 1992, pp. 15–30. Springer. DOI: [10.1007/978-3-7091-2604-2_2](https://doi.org/10.1007/978-3-7091-2604-2_2).
- [ArtDB84] “A continuum mechanics based four-node shell element for general non-linear analysis”. Eduardo N. Dvorkin and Klaus-Jürgen Bathe. In: *Engineering computations*. No. 1, Vol. 1, 1984, pp. 77–88. MCB UP Ltd. DOI: [10.1108/eb023562](https://doi.org/10.1108/eb023562).
- [ArtDha69] “Numerical analysis of thin shells by curved triangular elements based on discrete Kirchhoff hypothesis”. G Dhatt. In: *Proc. ASCE, Symp. on Applications of FEM in Civil Engineering, Vanderbilt Univ., Nashville, Tenn., 1969*. 1969, pp. 13–14.
- [ArtDha70] “An efficient triangular shell element”. GS Dhatt. In: *AIAA Journal*. No. 11, Vol. 8, 1970, pp. 2100–2102. DOI: [10.2514/3.6068](https://doi.org/10.2514/3.6068).
- [ArtDMM86] “A new triangular discrete Kirchhoff plate/shell element”. G. Dhatt, L. Marcotte, and Y. Matte. In: *International Journal for Numerical Methods in Engineering*. No. 3, Vol. 23, 1986, pp. 453–470. DOI: [10.1002/nme.1620230310](https://doi.org/10.1002/nme.1620230310).
- [ArtFlo13a] “A “Prism” solid element for large strain shell analysis”. Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 253, 2013, pp. 274–286. Elsevier. DOI: [10.1016/j.cma.2012.10.001](https://doi.org/10.1016/j.cma.2012.10.001).
- [ArtFlo13b] “Development of a non-linear triangular prism solid-shell element using ANS and EAS techniques”. Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 266, 2013, pp. 81–97. Elsevier. DOI: [10.1016/j.cma.2013.07.014](https://doi.org/10.1016/j.cma.2013.07.014).
- [ArtFlo13c] “Un Elemento Prisma Triangular De Sólido-lámina Para El Análisis Con Grandes Deformaciones”. Fernando G. Flores. In: *Mecánica Computacional*. Vol. Xxxii, 2013, pp. 63–87.
- [ArtFO01] “A basic thin shell triangle with only translational DOFs for large strain plasticity”. Fernando G. Flores and Eugenio Oñate. In: *International Journal for Numerical Methods in Engineering*. No. 1, Vol. 51, 2001, pp. 57–83. John Wiley & Sons, Ltd. DOI: [10.1002/nme.147](https://doi.org/10.1002/nme.147).
- [ArtFO05] “Improvements in the membrane behaviour of the three node rotation-free {BST} shell triangle using an assumed strain approach”. Fernando G. Flores and Eugenio Oñate. In: *Computer Methods in Applied Mechanics and Engineering*. No. 6–8, Vol. 194, 2005, pp. 907–932. DOI: [10.1016/j.cma.2003.08.012](https://doi.org/10.1016/j.cma.2003.08.012).
- [ArtFO11] “Wrinkling and folding analysis of elastic membranes using an enhanced rotation-free thin shell triangular element”. Fernando G. Flores and Eugenio Oñate. In: *Finite Elements in Analysis and Design*. No. 9, Vol. 47, 2011, pp. 982–990. Elsevier. DOI: [10.1016/j.finel.2011.03.014](https://doi.org/10.1016/j.finel.2011.03.014).
- [ArtFV82] “Some results on the behaviour of Degenerated Shell (D.S.) elements”. G. Fezans and G. Verchery. In: *Nuclear Engineering and Design*. No. 1, Vol. 70, 1982, pp. 27–35. DOI: [10.1016/0029-5493\(82\)90264-3](https://doi.org/10.1016/0029-5493(82)90264-3).
- [ArtHau+01] “Solid-shell elements with linear and quadratic shape functions at large deformations with nearly incompressible materials”. R. Hauptmann, S. Doll, M. Harnau, and K. Schweizerhof. In: *Computers & Structures*. No. 18, Vol. 79, 2001, pp. 1671–1685. Elsevier.

- [ArtHCB05] “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement”. T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. In: *Computer Methods in Applied Mechanics and Engineering*. No. 39, Vol. 194, 2005, pp. 4135–4195. DOI: [10.1016/j.cma.2004.10.008](https://doi.org/10.1016/j.cma.2004.10.008).
- [ArtHL81] “Nonlinear finite element analysis of shells: Part I. three-dimensional shells”. Thomas J.R. Hughes and Wing Kam Liu. In: *Computer Methods in Applied Mechanics and Engineering*. No. 3, Vol. 26, 1981, pp. 331–362. DOI: [10.1016/0045-7825\(81\)90121-3](https://doi.org/10.1016/0045-7825(81)90121-3).
- [ArtHS98] “A systematic development of ‘solid-shell’ element formulations for linear and non-linear analyses employing only displacement degrees of freedom”. R. Hauptmann and K. Schweizerhof. In: *International Journal for Numerical Methods in Engineering*. No. 1, Vol. 42, 1998, pp. 49–69. Wiley Online Library.
- [ArtHSD00] “Extension of the ‘solid-shell’ concept for application to large elastic and large elastoplastic deformations”. R. Hauptmann, K. Schweizerhof, and S. Doll. In: *International Journal for Numerical Methods in Engineering*. No. 9, Vol. 49, 2000, pp. 1121–1141.
- [ArtKan79] “A simple and efficient finite element for general shell analysis”. Worsak Kanok-nukulchai. In: *International Journal for Numerical Methods in Engineering*. No. 2, Vol. 14, 1979, pp. 179–200. DOI: [10.1002/nme.1620140204](https://doi.org/10.1002/nme.1620140204).
- [ArtKGW06a] “A robust non-linear solid shell element based on a mixed variational formulation”. Sven Klinkel, Friedrich Gruttmann, and Werner Wagner. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1, Vol. 195, 2006, pp. 179–201. Elsevier. DOI: [10.1016/j.cma.2005.01.013](https://doi.org/10.1016/j.cma.2005.01.013).
- [ArtKGW06b] “A robust non-linear solid shell element based on a mixed variational formulation”. Sven Klinkel, Friedrich Gruttmann, and Werner Wagner. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1–3, Vol. 195, 2006, pp. 179–201. Elsevier. DOI: [10.1016/j.cma.2005.01.013](https://doi.org/10.1016/j.cma.2005.01.013).
- [ArtKir50] “Über das Gleichgewicht und die Bewegung einer elastischen Scheibe.” Gustav Kirchhoff. In: *Journal für die reine und angewandte Mathematik*. No. 40, Vol. 1850, 1850, pp. 51–88. De Gruyter.
- [ArtLev99] “Sur l’équilibre élastique d’une plaque rectangulaire”. Maurice Levy. In: *Comptes Rendus Acad. Sci. Paris*. Vol. 129, 1899, pp. 535–539.
- [ArtLov88] “XVI. The small free vibrations and deformation of a thin elastic shell”. Augustus Edward Hough Love. In: *Philosophical Transactions of the Royal Society of London. (A.)*. No. 179, 1888, pp. 491–546. The Royal Society London.
- [ArtMG86] “A triangular thin-shell finite element based on discrete Kirchhoff theory”. S.S. Murthy and R.H. Gallagher. In: *Computer Methods in Applied Mechanics and Engineering*. No. 2, Vol. 54, 1986, pp. 197–222. DOI: [10.1016/0045-7825\(86\)90126-x](https://doi.org/10.1016/0045-7825(86)90126-x).
- [ArtMin51] “Influence of rotatory inertia and shear flexural motions of isotropic elastic plates”. Raymond D Mindlin. In: . 1951, DOI: [10.1007/978-1-4613-8865-4_29](https://doi.org/10.1007/978-1-4613-8865-4_29).
- [ArtNio85] “Shell theory, vol. xiv”. Frithiof I Niordson. In: . 1985, Amsterdam, New York, NY, North-Holland: Elsevier Science Pub Co.
- [ArtNob01] “Numerical approximation of fluid-structure interaction problems with application to haemodynamics”. Fabio Nobile. In: . 2001, Epfl.
- [ArtNoo90a] “Bibliography of monographs and surveys on shells”. Ahmed K Noor. In: *Applied Mechanics Reviews; (United States)*. No. 9, Vol. 43, 1990, DOI: [10.1115/1.3119170](https://doi.org/10.1115/1.3119170).
- [ArtNoo90b] “Bibliography of Monographs and Surveys on Shells”. Ahmed K. Noor. In: *Applied Mechanics Reviews*. No. 9, Vol. 43, 1990, pp. 223–234. DOI: [10.1115/1.3119170](https://doi.org/10.1115/1.3119170).
- [ArtOF05] “Advances in the formulation of the rotation-free basic shell triangle”. Eugenio Oñate and Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. No. 21–24, Vol. 194, 2005, pp. 2406–2443. DOI: [10.1016/j.cma.2004.07.039](https://doi.org/10.1016/j.cma.2004.07.039).
- [ArtOUS04] “Selective mass scaling for thin walled structures modeled with tri-linear solid elements”. L. Olovsson, M. Unosson, and K. Simonsson. In: *Computational Mechanics*. No. 2, Vol. 34, 2004, pp. 134–136. DOI: [10.1007/s00466-004-0560-6](https://doi.org/10.1007/s00466-004-0560-6).

- [ArtOZ00] "Rotation-free triangular plate and shell elements". Eugenio Oñate and Francisco Zarate. In: *International Journal for Numerical Methods in Engineering*. No. 1-3, Vol. 47, 2000, pp. 557–603. DOI: [10.1002/\(sici\)1097-0207\(20000110/30\)47:1/3<557::aid-nme784>3.0.co;2-9](https://doi.org/10.1002/(sici)1097-0207(20000110/30)47:1/3<557::aid-nme784>3.0.co;2-9).
- [ArtPar+06] "Sheet metal forming simulation using {EAS} solid-shell finite elements". M.P.L. Parente, R.A. Fontes Valente, R.M. Natal Jorge, R.P.R. Cardoso, and Ricardo J. Alves de Sousa. In: *Finite Elements in Analysis and Design*. No. 13, Vol. 42, 2006, pp. 1137–1149. DOI: [10.1016/j.finel.2006.04.005](https://doi.org/10.1016/j.finel.2006.04.005).
- [ArtPar79] "A critical survey of the 9-node degenerated shell element with special emphasis on thin shell application and reduced integration". H. Parisch. In: *Computer Methods in Applied Mechanics and Engineering*. No. 3, Vol. 20, 1979, pp. 323–350. DOI: [10.1016/0045-7825\(79\)90007-0](https://doi.org/10.1016/0045-7825(79)90007-0).
- [ArtRei45] "The effect of transverse shear deformation on the bending of elastic plates". Eric Reissner. In: *J. appl. Mech.*. 1945, A69–a77.
- [ArtSAV11] "On the use of EAS solid-shell formulations in the numerical simulation of incremental forming processes". JIV Sena, RJ Alves de Sousa, and RAF Valente. In: *Engineering Computations*. No. 3, Vol. 28, 2011, pp. 287–313. Emerald Group Publishing Limited. DOI: [10.1108/02644401111118150](https://doi.org/10.1108/02644401111118150).
- [ArtSen+16] "Single point incremental forming simulation with adaptive remeshing technique using solid-shell elements". José IV Sena, Cedric Lequesne, Laurent Duchene, Anne-Marie Habraken, Robertt AF Valente, and Ricardo J Alves de Sousa. In: *Engineering Computations*. No. 5, Vol. 33, 2016, pp. 1388–1421. Emerald Group Publishing Limited. DOI: [10.1108/ec-06-2015-0172](https://doi.org/10.1108/ec-06-2015-0172).
- [ArtSF89] "On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parametrization". J.C. Simo and D.D. Fox. In: *Computer Methods in Applied Mechanics and Engineering*. No. 3, Vol. 72, 1989, pp. 267–304. DOI: [10.1016/0045-7825\(89\)90002-9](https://doi.org/10.1016/0045-7825(89)90002-9).
- [ArtSFR89] "On a stress resultant geometrically exact shell model. Part II: The linear theory; Computational aspects". J.C. Simo, D.D. Fox, and M.S. Rifai. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1, Vol. 73, 1989, pp. 53–92. DOI: [10.1016/0045-7825\(89\)90098-4](https://doi.org/10.1016/0045-7825(89)90098-4).
- [ArtSFR90] "On a stress resultant geometrically exact shell model. Part III: Computational aspects of the nonlinear theory". J.C. Simo, D.D. Fox, and M.S. Rifai. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1, Vol. 79, 1990, pp. 21–70. DOI: [10.1016/0045-7825\(90\)90094-3](https://doi.org/10.1016/0045-7825(90)90094-3).
- [ArtSim93] "On a stress resultant geometrically exact shell model. Part VII: Shell intersections with 56-DOF finite element formulations". J.C. Simo. In: *Computer Methods in Applied Mechanics and Engineering*. No. 3, Vol. 108, 1993, pp. 319–339. DOI: [10.1016/0045-7825\(93\)90008-1](https://doi.org/10.1016/0045-7825(93)90008-1).
- [ArtSK92] "On a stress resultant geometrically exact shell model. Part V. Nonlinear plasticity: formulation and integration algorithms". J.C. Simo and J.G. Kennedy. In: *Computer Methods in Applied Mechanics and Engineering*. No. 2, Vol. 96, 1992, pp. 133–171. DOI: [10.1016/0045-7825\(92\)90129-8](https://doi.org/10.1016/0045-7825(92)90129-8).
- [ArtSLL04] "Popular benchmark problems for geometric nonlinear analysis of shells". K.Y. Sze, X.H. Liu, and S.H. Lo. In: *Finite Elements in Analysis and Design*. No. 11, Vol. 40, 2004, pp. 1551–1569. DOI: [10.1016/j.finel.2003.11.001](https://doi.org/10.1016/j.finel.2003.11.001).
- [ArtSou+05] "A new one-point quadrature enhanced assumed strain (EAS) solid-shell element with multiple integration points along thickness: Part I—geometrically linear applications". Ricardo J. Alves de Sousa, Rui P. R. Cardoso, Robertt A. Fontes Valente, Jeong-Whan Yoon, José J. Grácio, and Renato M. Natal Jorge. In: *International journal for numerical methods in engineering*. No. 7, Vol. 62, 2005, pp. 952–977. Wiley Online Library.
- [ArtSR11] "A reduced integration solid-shell finite element based on the EAS and the ANS concept—Large deformation problems". Marco Schwarze and Stefanie Reese. In: *International Journal for Numerical Methods in Engineering*. No. 3, Vol. 85, 2011, pp. 289–329. Wiley Online Library. DOI: [10.1002/nme.2966](https://doi.org/10.1002/nme.2966).
- [ArtSRF90] "On a stress resultant geometrically exact shell model. Part IV: Variable thickness shells with through-the-thickness stretching". J.C. Simo, M.S. Rifai, and D.D. Fox. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1, Vol. 81, 1990, pp. 91–126. DOI: [10.1016/0045-7825\(90\)90143-a](https://doi.org/10.1016/0045-7825(90)90143-a).

- [ArtSRF92] “On a stress resultant geometrically exact shell model. Part VI: Conserving algorithms for non-linear dynamics”. J.C. Simo, M. S. Rifai, and D. D. Fox. In: *International Journal for Numerical Methods in Engineering*. No. 1, Vol. 34, 1992, pp. 117–164. DOI: [10.1002/nme.1620340108](https://doi.org/10.1002/nme.1620340108).
- [ArtSVR11] “Sheet metal forming and springback simulation by means of a new reduced integration solid-shell finite element technology”. Marco Schwarze, Ivaylo N Vladimirov, and Stefanie Reese. In: *Computer Methods in Applied Mechanics and Engineering*. No. 5, Vol. 200, 2011, pp. 454–476. Elsevier. DOI: [10.1016/j.cma.2010.07.020](https://doi.org/10.1016/j.cma.2010.07.020).
- [ArtTSW05] “Effective Modeling and Nonlinear Shell Analysis of Thin Membranes Exhibiting Structural Wrinkling”. Tessler Alexander, Sleight David W., and Wang John T. In: *Journal of Spacecraft and Rockets*. No. 2, Vol. 42, 2005, pp. 287–298. American Institute of Aeronautics and Astronautics. DOI: [10.2514/1.3915](https://doi.org/10.2514/1.3915).
- [ArtWas82] “Variational methods in elasticity and plasticity Pergamon Press”. K Washizu. In: . 1982, Oxford, England.
- [ArtWem69] “Finite elements, finite rotations and small strains of flexible shells”. Gerald Wempner. In: *International Journal of Solids and Structures*. No. 2, Vol. 5, 1969, pp. 117–153. Elsevier. DOI: [10.1016/0020-7683\(69\)90025-0](https://doi.org/10.1016/0020-7683(69)90025-0).
- [ArtWOK68] “Finite-element analysis of thin shells”. Gerald A Wempner, John T Oden, and Dennis A Kross. In: *Journal of the Engineering Mechanics Division*. No. 6, Vol. 94, 1968, pp. 1273–1294. Asce.

Ph.D.'s thesis

- [PhDCal06] “Simulación del flujo sanguíneo y su interacción con la pared arterial mediante modelos de elementos finitos”. Francisco José Calvo Plaza. 2006.
- [PhDVMO09] “Nonlinear Finite Element Analysis of Orthotropic and Prestressed Membrane Structures”. J. G. Valdés, J. Miquel, and Eugenio Oñate. Elsevier Science Publishers B. V. 2009.

Online resources

- [OnIKra] *Kratos Homepage*. URL: <http://www.cimne.com/kratos/>.
- [OnIMel+16] *GiD Homepage*. URL: <http://www.gidhome.com/%20https://www.gidhome.com/support/brand-resources/citing-gid/>.

Chapter 4

Contact mechanics

“ Alan Turing is reported as saying that PDEs are made by God, the boundary conditions by the Devil! The situation has changed, Devil has changed places... We can say that the main challenges are in the interfaces, with Devil not far away from them... ”

Jacques-Louis Lions
(1928 - 2001 AD, French mathematician)

4.1 Introduction

When speaking about **CCM** we need to know that the phenomenon covers an extended range of problems, and depending of the problem we want to solve we will need to consider a different numerical technique[BookWri06]. The most general one, and the one we are interest on is the resolution employing **FEM**, which can be applied in a large range of problem, from small to large deformations and from linear to **NL** material behaviour. The second methods are the **Discrete Element Method (DEM)** which study the interaction between a large number of particles coming into contact. Finally, *multibody systems*, which describe rigid bodies interacting between them, creating a mechanism, and where contact may influence the behaviour of the system.

Despite the large number of topics covered by **CCM**, as we already mention, we will focus on the application of **FEM** for the resolution of contact problems. This is consistent with the developments of the previous chapters are classified on this type of method.

4.1.1 Historical outline

History of the contact mechanics is probably as long as the history of civilization. Practically any physical interaction between object involves contact and friction, making then the invention of the wheel probably the first human invention involving this problematic. What we know for sure[BookPop10] is that the ancient Egyptians already employed oil reducing the friction between the wheels and the floor in order to facilitate the transportation of the pyramid blocks (Figure 4.1).

We then can jump to the modern contributions of *Leonardo Da Vinci* in the *Codex-Madrid I*, who experimented with the friction phenomenon. He studied the influence of the contact area on the friction, testing objects of the same mass but different contact area, coming to the conclusion that the frictional force is proportional to the weight

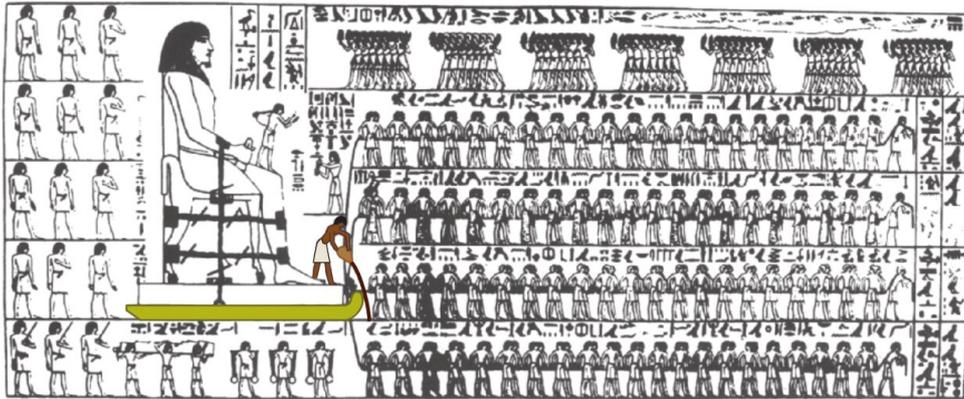


Figure 4.1: Representation of moving blocks during ancient Egypt. Source[BookPop10]

and independent of the contact area. This was a crucial influence to Charles Augustin Coulomb, French engineer, to which we must thank many of the expressions used in computational mechanics[ArtPP15], like the extended expression $F_\tau = \mu N$, denominated *Coulomb* friction law. It is in 1785[ArtCou85], when in *The theory of simple machines* differentiated for first time between *kinetic* and *static* friction.

The first mathematical contributions we have records off are attributed to *Euler*[ArtEul50a; ArtEul50b]¹, who studied the friction problem assuming that the roughness of the contact surfaces can be represented with a series of triangles. This leads him to the conclusion that the static friction coefficient is larger than the dynamic one, which means we need to apply a larger force in order to starting to move an object than once that the object is moving. In the same way that we own the use of the Greek letter π to represent the circumference ratio, among other Greek letters used as mathematical symbols, we own to *Euler* the use of μ in order to represent the friction coefficient.

During a long time, the contact conditions were modelled in a very experimental manner, as the lack of analytical results did not allow consider an alternative approach. It was *Hertz* with his² *On the contact of elastic solids*[ArtHer82] who presented the first analytical solution for a contact problem. In 1882, *Hertz* solved the contact problem of two elastic bodies with curved surfaces (Figure 4.2). This problem is so relevant that still today, this classical solution provides a foundation for modern problems in contact mechanics, and it is extensively used as the main *benchmark* for contact mechanics today[ArtSpe], as we will see in following sections.

Ueber die Berührung fester elastischer Körper.

(Von Herrn *Heinrich Hertz*.)

In der Theorie der Elasticität werden als Ursachen der Deformationen theils Kräfte, welche auf das Innere der Körper wirken, theils auf die Oberfläche wirkende Druckkräfte angenommen. Für beide Arten von Kräften kann der Fall eintreten, dass dieselben in einzelnen unendlich kleinen Theilen der Körper unendlich gross werden, so zwar, dass die Integrale der Kräfte über diese Theile genommen einen endlichen Werth behalten. Beschreiben wir alsdann um den Unstetigkeitspunkt eine geschlossene Fläche, deren Dimensionen sehr klein gegen die Dimensionen des ganzen Körpers

Figure 4.2: Original *Hertz* paper in 1882. Source[ArtHer82]

The developments that happened just after that were motivated for its application to railways, reduction gears and to rolling contact bearing industry when Industrial Revolution was at full steam at the beginning of the XXth century. The analytical solutions that appeared during this time only allowed to solve a very limited rank of problems, limited to

¹Being widely considered the most prolific mathematician of all time, there was almost no field that he did not study. As *Pierre-Simon Laplace* said: "Read Euler, read Euler, he is the master of us all".

²In German in the original version: *Über die berührung fester elastische Körper und über die Harte*

only certain simple geometries, linear material mainly, and not much more. This contrasted with the industrial needs, where the solution of complex geometries, not trivial **BC** and **NL** materials. Not only that, industrial needs implied even more complex behaviours, with friction, wear, adhesion, large deformations and large sliding[BookYas13].

In 1933 *Signorini*[ArtSig33] formulated the general problem of the equilibrium of a linear elastic body in frictionless contact with a rigid foundation, what is also denominated *unilateral* contact problem. Later more mathematical developments came, like the contributions of *Fichera*[ArtFic63] on the uniqueness of *variational inequalities*. Later contributions like the book of *Kikuchi* and *Oden*[BookKO88] who extended the prove of *Fichera* to the *Signorini* problem.

In any case, until the existence of modern computation, for industrial applications contact was modelled as a local problem using the stress and strain field obtained directly from the analysis of a complete structure[BookWri06]. Once the computational power raised in the second half of XXth century the whole **NL** constraint of the contact problem could be considered. At the beginning only some semi-analytical problems could be solved, but after the appearance of modern **FEM** with **NASA STRucture ANALysis (NASTRAN)** more and more developments have emerged. In a first stage, only *Signorini*'s problem was solved, limiting the resolution of *unilateral* contact, in further steps friction was included, later large deformations and finally *bilateral* multibody contact.

We can also mention some additional problematic that emerged in order to treat the *frictional* problem. While the *frictionless* problem can be easily formulated as minimisation problem with *inequality constraint* following standard approaches (*barrier*, **PM**, **LMM**, **ALM**, ...), see the respective deductions from the annexe **D.Constrained optimisation problems**. There is not such associated minimisation principle for the *frictional* contact problem, as *Kikuchi* and *Oden*[BookKO88] proved. This is due to the dependence of the frictional status with the normal contact pressure, which at the same time induces additional second order dependencies (such geometrical configuration). Since then the *frictional* problem has been proved as a complex problem with many challenges.

In order to tackle these problematic, several approaches to work on these problematic. Some developments came from the replacement of a *variational inequality* with a *variational equality* with a modified contact term, which allowed to consider the classical optimisation techniques. More techniques have been proposed, *Wriggers*[BookWri06] list a wide list of alternatives emerged, like *simplex* method, *parametric quadratic programming*, the *flexibility* method, *Nitsche* method, direct elimination, *cross constraint*, among others.

Additionally to classic contact mechanics developments on the **Tribology** theory was made. We can mention as examples the contributions related with the adhesive contact by *Johnson et al.*[ArtJKR71] and *Derjaguin et al.*[BookMau13]. These contributions were in conflict between them, leading to the two most representative models of adhesive elastic contact, the **Johnson, Kendall and Roberts (JKR)** and **Derjaguin, Muller and Toporov (DMT)** respectively, these two methods are studied and compared in *Barthel*[ArtBar08]. More recent developments by *Talon and Curnier*[ArtTC03] couple the adhesive model with frictional contact.

Furthermore, on the **Tribology** domain is important to mention the contributions from *Bowden* and *Tabor*[ArtBT39; BookBBT01] in the mid-twentieth century. They were the first to emphasise the importance of surface roughness for bodies in contact, due to the microstructure of the surfaces in contact, the true contact area between friction partners is found to be less than the apparent contact area. Additionally we can mention the work of *Kragelsky*[ArtKra68], who was one of the first on doing developments related to the wear phenomenon.

4.1.2 Contact problem

Has we already said, we can find contact and friction in almost all kinds of movements, both in nature and in human-origin ones. We can find this problem[BookWri06] in many engineering applications, p.e. on civil engineering the foundation problem requires to take into account the contact³, but traditionally, restricting the solution to small deformations, several simplifications were taken into account. Other problems related to this field of engineering, more industrial-related, can be the study of bearing and connections on metallic structures. The applications in mechanical engineering may even be applicable in more problematic, like gears and bearings, metal forming, cutting processes, rolling contact of car tyres, crashing among other problems. Extending this field to the domain of the biology, we can find in biomechanics problematic such as human joints, teeth implants, stents, etc.

³It is extensively known the solution of *Boussinesq*[BookBou85], which considers several simplifications in order to represent the elastic support.

The physics of the contact interaction is particularly rich and complicated [PhDYas11], due to the multiscale and multiphysical nature of the phenomenon. This kind of mechanical problems are classically formulated as a boundary value problems, where the contact constraints are formulated as sets of inequalities, and the problem becomes even more complex when the frictional effect is assumed. A whole discussion about the constraints and optimisation can be found in the Appendix D. [Constrained optimisation problems](#). In fact, a proper treatment of the contact constraint requires the consideration of the micromechanical approach, not only for the proper consideration of the frictional behaviour, but also in the normal contact [BookWri06].

These boundary conditions are solution dependent, which naturally leads to difficulties in the formulation of the frictional contact problem, for example considering the *Coulomb's* friction law yields to a non-smooth [BookAnd01] energy functional. Additionally the contacting bodies may penetrate each other or be separated, where with the **FE** discretisation leads to mathematical and numerical difficulties. Another challenge is the boundary discretisation and methods to surpass this problematic. These points are discussed on the state-of-the-art section (4.2. [State of the Art in computational contact mechanics](#)) and the formulation section (4.3. [Formulation](#)).

Finally, the algorithm considered for the detection phase can suppose a significant bottleneck in terms of efficiency. This is discussed and presented in deeper detail in the corresponding section 4.4. [Contact detection. Search techniques](#).

First we should define the basic contact problem, Figure 4.3 illustrates a common contact problem. Every contact problem is defined between two entities (not necessarily different, e.g. *self-contact*) here noted as Ω^i for $i = 1, 2$. The contact problem occurs on the interface, where we want to link the displacements of the first domain (\mathbf{x}^1) with its projection over second domain ($\hat{\mathbf{x}}^2$) in a local reference frame (\mathbf{n} , $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$). The contact problem will consist in avoid the penetration in the normal direction between domains, and respect the tangent movement restriction imposed by the frictional component of the contact.

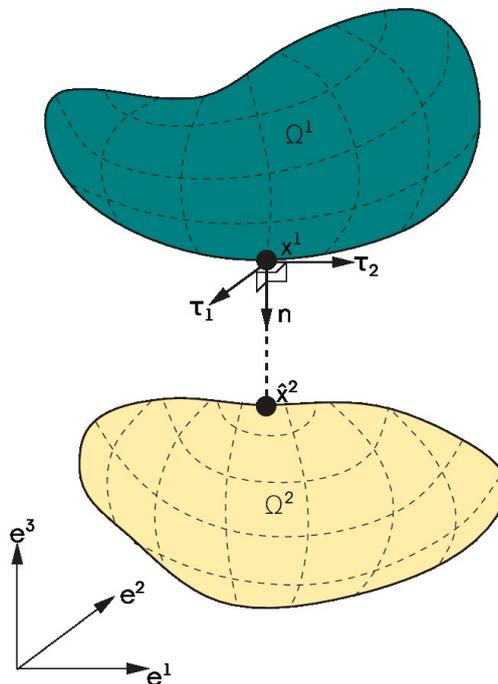


Figure 4.3: Basic definition of the contact problem

It is important to distinguish that we can divide two types of contacts, the *bilateral* and *unilateral*, where the first one supposes the contact between two or more deformable bodies, in contrast to the *unilateral* contact, where the contact occurs between a rigid solid and a deformable solid. *A priori* for the problems of metal forming the second one is enough to formulate the problem, but in order to consider more advanced problems in this field *a priori* is preferable to consider the *bilateral* case for being more general.

On Figure 4.4 the different states of the contact problem are presented. In here we can differentiate each one of

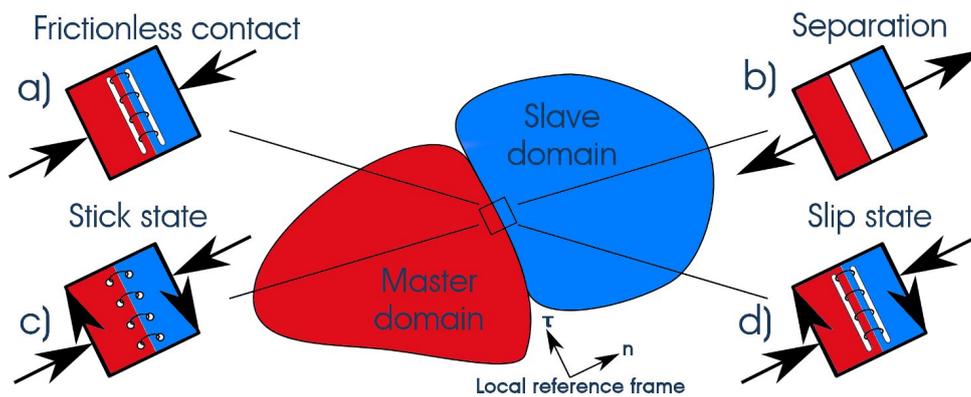


Figure 4.4: Contact states. Based on[PhDYas11]

them, with its corresponding idealised simplification. The illustration shows two different bodies⁴, with its corresponding denominations as master and slave. In the interface between the two bodies there is a local reference frame, defined by the normal and tangent directions on this interface. The states listed on this image can be listed as:

- **a) Frictionless contact:** This contact allows the movement in the tangent direction of the local reference frame, but does not allow the movement toward the opposite body. The idealisation will lead to a perfect rail, no friction, between the two bodies. This is what is usually denominated as **Karush-Kuhn-Tucker (KKT)** condition.
- **b) Separation:** This state is the absence of contact, meaning that the two bodies are not in contact any more and therefore there is not interaction between them any more.
- **c) Stick state:** In the case of the frictional contact, before the threshold value is surpassed, whatever model of friction is considered, the two bodies will be fully tied, acting as one body. The idealisation will be this case will correspond with a sewing in the interface.
- **d) Slip state:** Once this threshold is surpassed, the bodies can move freely in the tangent direction. Will be the same as the case **a)**, but in this case the rail which idealises the interface will not be ideal, and therefore hindering the movement due to the present friction.

4.2 State of the Art in computational contact mechanics

4.2.1 Introduction

In this section we compare the existing methods, we can consider in order to solve the contact problem. Due to the huge industrial and engineering interest in this type of problem, many developments exist, and therefore we will focus on the most relevant ones on the sake of brevity. First, we will introduce the discretisation methods available, following with the optimisation procedures necessities to fulfil the contact inequality constraint. Finally, a brief introduction to existing techniques available to model the frictional behaviours will be presented. We address to some additional monographs, which will provide a deeper outline of the State of the Art in **CCM**. We found particularly relevant to highlight the book of *Wriggers*[BookWri06], *Laursen*[BookLau10], *Schweizerhof*[BookSK13] and *Yastrebov*[BookYas13].

4.2.2 Discretisation

In order to solve the contact problem, several approaches can be considered from the point of view of the discretisation. When we mention discretisation, we refer the procedure followed for the integration of the interface. The following ones are the main methods that can be considered from this perspective.

⁴As previously stated, the contact may occur between more bodies, but here for the sake of simplicity, and in order to present the problem we will consider only two bodies.

4.2.2.1 NTN

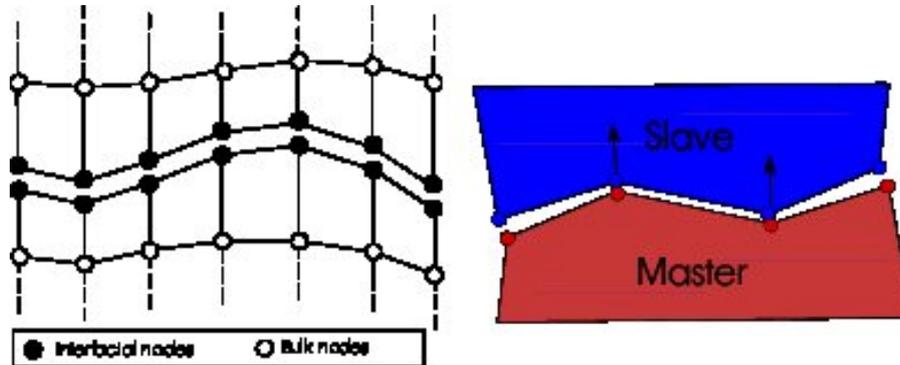


Figure 4.5: NTN. Inspired[PhDYas11]

The discretisation of the contact interface is between a node of the slave domain to a node of the master domain (Figure 4.5), reason why the method is denominated this way. This method, originally proposed by *Franca* and *Zienkiewicz*[ArtFZ75] in 1975 is the oldest and the simplest of all the discretisation methods available. We can list the pros and cons of the method as follows:

- **Pros:**

- Passes the *Taylor test*[ArtTP] when the mesh is conforming
- Simple conceptually and to implement. As it is node to node, which means that relate directly **DOF**, it is possible to consider all kinds of constraint enforcement methodologies.

- **Cons:**

- Small slip and small deformations. The method loses precision when the nodes move across the interface and the nodes are not coincident any more.
- Related with previous method, the mesh must be conforming between interfaces. This constraint introduces a restriction in mesh generation.

4.2.2.2 NTS

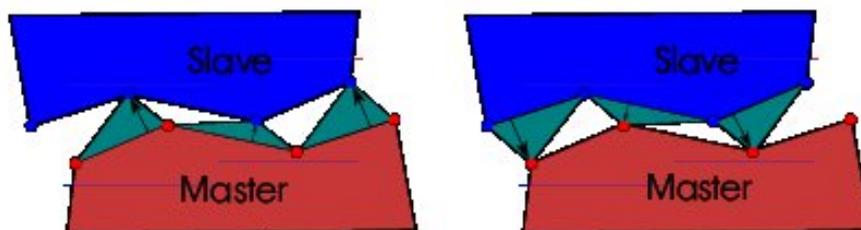


Figure 4.6: NTS. Inspired[PhDYas11]

The discretisation of the contact interface is between a node of the slave domain to the surface of the master domain (Figure 4.6). Despite of being quite close in time to the **NTN**, in 1977 *Hughes et al.*[ArtHTK77] proposed the **NTS**, which already allowed to solve problems in large deformations. We can summarise the pros and cons of the method in:

- **Pros:**

- Simple and robust. The implementation is probably the most extended in **FEA** software.
- Large deformations and large slip can be considered
- Mesh independent. The meshes are not required to be conforming in the interface like in the **NTN**

- **Cons:**

- Fails *Taylor* test[ArtTP] for nonconforming meshes, except if considering the *double pass LMM*, solving the problem twice, swapping *master* and *slave*. This problem is solved with the modification suggested by Zavarise[ArtZL09].
- It has several difficulties to calculate the gap in order to compute the contact. In the work of Yastrebov[PhDYas11] several alternatives to solve this issue are proposed.

4.2.2.3 CDM

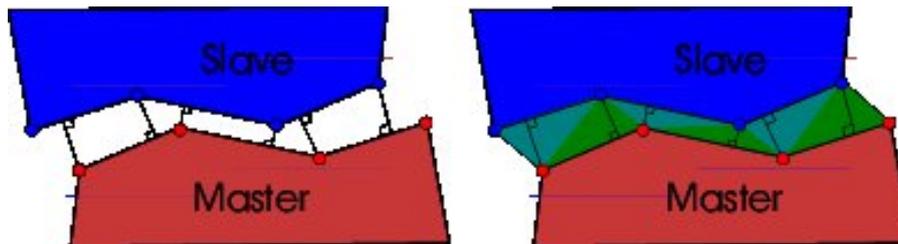


Figure 4.7: **CDM**. Inspired[PhDYas11]

The discretisation of the contact interface is based on a full triangulation of the zone between contacting surfaces based on surface nodes (Figure 4.7). This method is in fact a full symmetric **NTS** discretisation proposed originally by Oliver[ArtOli+09a; ArtOli+09b] and Hartmann[ArtHar+09]. We can list the main advantages and disadvantages as:

- **Pros:**

- Passes *Taylor* test[ArtTP]
- Large deformations and large slip can be considered like in the **NTS**
- In-house developed in **CIMNE**[ArtOli+09a; ArtHar+09]

- **Cons:**

- Mesh dependent, in part related by next point
- Triangulation problems may happen for 3D cases[ArtHar+10]

4.2.2.4 STS (*Mortar methods*)

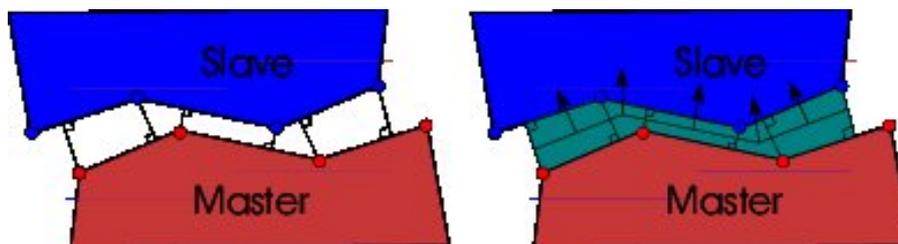


Figure 4.8: **STS**. Inspired[PhDYas11]

In the **STS** discretisation of the contact interface is between a surface of the slave domain to a node of the master domain (Figure 4.8), it is also denominated *Mortar* method as a metaphor to the strong union between bricks in a wall. As a note of curiosity, the method was originally developed on the field of **Domain Decomposition Method (DDM)**[BookWoh01; BookTos05], but originally proposed for **CCM** by Simo[ArtSWT85]. The method can be summarised as:

- **Pros:**

- Passes *Taylor* test[ArtTP], even with different types of mesh combinations, *i.e.* tetrahedra with hexahedra meshes as shown in the section for numerical examples 4.5.Numerical examples

- Correct integration of contact forces. The method is consistent as both meshes are fully integrated.
 - Large deformations and large slip can be considered[ArtPL04]. The *Mortar* based formulation leads to a consistent formulation of the frictional contact problem for large sliding and large deformations.
 - Additionally the *Mortar* formulation has been shown as being general enough. As a method coming from **DDM** it can be considered in order to couple different types of problems in a Multiphysics way. The method can be considered on mapping techniques (see our appendix [E.Mortar mapper](#)) or to strongly couple problems with mesh tying techniques[PhDPop12]. The work of Seitz[ArtSWP18] shows a fully integrated Multiphysics with *thermo-elasto-plastic frictional contact*, or the **FSI** implementation in Popp[PhDPop12] thesis. The consideration of **DLMM** (4.2.3.4.DLMM) extends the application range of the formulation.
- **Cons:**
 - Complex implementation in 3D. We experienced great advances since *Puso and Laursen*[ArtPL04] and later thanks to *Brunssen*[PhDBru08], *Popp*[PhDPop12] and *Gitterle*[PhDGit12]. The complexity comes from the 3D intersection between two flat geometries, Figure 4.9, which requires some additional considerations, especially when the *Gateaux* derivatives (see from appendixes [4.6.Derivatives for contact mechanics linearization](#)).

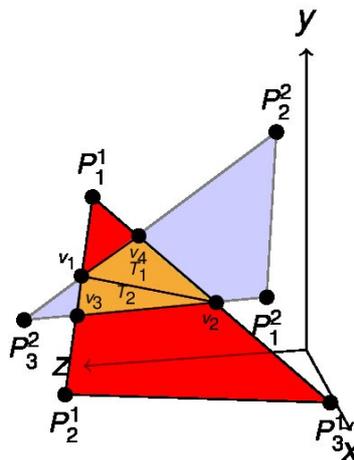


Figure 4.9: 3D segmentation between two triangles

4.2.2.5 Other alternative methods

The following section introduces some additional integration approaches existing in the state of art of the **CCM**. Many of these approaches cannot be directly applied in a standard **FEM** formulation, but we consider them as interesting to mention.

4.2.2.5.1 Isogeometric :

In the recent years, the isogeometric analysis has been on the rise. The key concept underlying on this method is the meshless integration. The integration is done directly on the **NURB**, avoiding them the necessity of the mesh step during the preprocess. This method originally developed by *Hughes*[BookHBC09] has grown significantly on last years due to the continuous workflow integration between **FEA** and **CAD**. Additionally **NURB** provide an exact representation of the surfaces and high order of integration during the **FE** computations.

The method has been considered particularly in order to represent surfaces, which suits to thin objects such as the ones present during the *forming* simulations. Additionally to the corresponding shell formulation[ArtBen+11] contact formulations are needed. We can mention respect this, the works from *Lorenzis*[ArtLWZ12a; ArtLWZ12b].

4.2.2.5.2 Smooth surface approximation :

In order to overcome the problematic from the discontinuity in the standard **FEM** interface, we may define a smooth approach of the boundary[BookBHS07]. In order to do so, we can consider different types of discretisations, such as *Hermite*, *splines*, *Bézier* surfaces or **NURB**.

Sometimes this last approach can also be denominated **Segment-To-Analytical-Surface (STAS)**[BookSK13], and can be found between the previous method (4.2.2.5.1.*Isogeometric*) and the standard **FEM**. There are different ways to approach analytically the surface. We can define this surface at the beginning of the simulation and then move it, which implies that the surface is rigid and then limits the contact to unilateral cases[ArtWI93]. This is not necessarily a limit, if we think that in the case of forming processes the tools considered are *de facto* rigid. The second approach can consist in consider standard **FEM**, with two deformable bodies and update the analytical surface continuously, this is for example done with *Nagata*[ArtNag05] patches by *Neto*[ArtNet+14; ArtNOM17].

4.2.2.6 Conclusion

After all the methods presented, and due to our consideration of standard **FEM**, we will consider as discretisation method the **STS** or *Mortar* approach (4.2.2.4.**STS (Mortar methods)**). This method will provide us the best standard **FEM** integration possible, despite its technical problems related to implementation details.

4.2.3 Optimisation method

The assumption of a known *a priori* contact surface allows to replace the variational inequality by a variational equality with an additional contact term. The form of this contact term depends upon the choice of the optimisation method. Owing to the fact that we have decided to implement a *Mortar* approach discretisation based in *Popp*[PhDPop12], our initial thought could be to consider as optimisation method the **DLMM**. In the following section, we will show the most common approaches that could be taken into consideration in order to solve the **CCM** problem, justifying this way our final choice.

The main optimisation methods are studied in detail in the corresponding appendix, we directly address to [D.Constrained optimisation problems](#) for further detail about these methods. In the mentioned appendix not only the different methods are introduced, but also compared in detail between them, and its suitability for solving different types of constrained problems.

4.2.3.1 Penalty Method (PM)

This is probably the most extended optimisation method considered on **CCM**, particularly in explicit approaches. The contact conditions are fulfilled exactly only in case of the infinite penalty parameter (ε), which results in *ill-conditioning*. For more details check [D.2.Penalty method](#).

- **Pros:**

- Very simple and robust. The simplicity of the method can be seen in the corresponding section in the optimisation appendix [D.2.Penalty method](#), and its robustness in the last numerical example shown ([D.7.3.Over-constrained optimisation problem](#)).
- Pure displacement based formulation, no change in the system size.

- **Cons:**

- The solution is inexact, which happens only in case of an infinite value of ε , making the system unsolvable due to the deterioration on the condition number.
- Choice of penalty parameter. If the value is chosen to small the penetration is large, if large as we have stated will ill-condition the system.

4.2.3.2 LMM

Contact conditions are exactly satisfied by the introduction of an extra **DOF** called **LM**, and usually represented by the Greek letter λ . This and more is analysed in detail at [D.3.Lagrange Multiplier method](#).

- **Pros:**
 - Exact solution is obtained when solving the system of equations. This is the main advantage of the **LMM**, and the reason of its extensive use in optimisation problems.
 - User independent. It does not depend, as in the case of the **PM** of a user-driven decision to choose a parameter.
- **Cons:**
 - Additional **DOFs**. The **LHS** of the system grows in n^{contact} additional **DOF** for the case of frictionless contact, and in $3 \times n^{\text{contact}}$ in the case of frictionless with full λ components or for frictional problems. Here n^{contact} is the number of nodes in contact.
 - Moderate convergence rate in the **NR**. Additionally the condition of the system is affected due to the zero diagonal terms introduced by the *Lagrange* multipliers.

4.2.3.3 ALM

It is a **LMM** regularised by a **PM**. It yields a smooth energy functional and fully unconstrained problem, resulting in exact fulfilment of contact constraints with finite value of the penalty parameter (ε). For more details check [D.4.Augmented Lagrange Multiplier method](#). The method is successfully considered with a *Mortar* approach by *Cavaleri and Cardona* [[ArtCFC12](#); [ArtCC12](#); [ArtCC13a](#); [ArtCC15](#); [ArtCC13b](#)].

The method is often considered as a synonym of the *Uzawa* iteration (see [D.4.2.2.Uzawa iteration](#)). Indeed the *Uzawa* iteration is always applied to the **ALM**, but this is just one of the possible approaches possible to be considered in order to solve the system. In the following when we mention the **ALM** we will refer to the standard approach without *Uzawa* iteration. This is due to the reason that the convergence order of the *Lagrange* multiplier becomes linear.

- **Pros:**
 - Exact solution as in the case of the **LMM**. The result is not influenced by the penalty terms (ε).
 - No additional **DOFs** with *Uzawa* iteration algorithms. Otherwise the system of equations is increased in the same way it is done in the standard **LMM**.
 - Smooth functional is obtained. See Figure [D.4](#) from the appendix, compared with the Figure [D.3](#).
 - Less sensitive to penalty choice compared with the **PM**. The penalty (ε) only contributes in smooth \mathcal{L} , but does not affect the solution. Once the system has converged, the contribution of ε becomes zero.
- **Cons:**
 - In the case of an *Uzawa* iteration algorithm coming comes at the price of increased computational costs due to an additional nonlinear iteration loop (augmentation), typically only showing a linear convergence rate.

4.2.3.4 DLMM

This concept of dual *Lagrange* multiplier was introduced first by *Wohlmuth* [[ArtWoh02](#)]. The method is an extension of the **LMM**, but the *Lagrange* multiplier (λ) are interpolated considering a different set of shape functions. These shape functions are locally supported and continuous dual basis functions.

- **Pros:**
 - Exact solution as in the **LMM** case

- System can be condensed, reducing the **DOF** respect the original, the system then becomes pure displacement.
- With regard to linear solvers, **DLMM** allows for an *out-of-the-box* application of *state-of-the-art* iterative solution and preconditioning techniques, such as the **Generalized Minimal RESidual (GMRES)** iterative solver. It can be used with **Algebraic MultiGrid (AMG)** methods if the proper modifications are taken into account, like recent works of *Wiesner*[ArtWie+18].
- **Cons:** As the disadvantages of the **LMM** disappear, there is none. We can mention the fact that is slightly more difficult to implement in comparison to the standard **LMM**.

4.2.3.5 Other alternative methods

The following are some other additional approaches possible to solve the optimisation problem related with contact. Here we will add the ones we have considered as most promising after the already presented, more methods are mentioned in the optimisation appendix (D.1.Introduction).

4.2.3.5.1 Perturbed Lagrangian :

This special formulation can be used to combine both **PM** and **LMM** in a mixed formulation, therefore, similar to the already mentioned **ALM**. The method was considered originally by *Oden*[ArtOde81] in 1981. In this method the contribution of the *Lagrange* multiplier is regularised with the inclusion of a complementary term. This method is discarded due to its limitations in order to consider the slip case in frictional simulations, where this is usually done with the inclusion of an incremental constitutive equation for the frictional behaviour considered[BookWri06]. In addition to that we can mention as an advantage that the method does not deteriorate the conditioning of the system.

4.2.3.5.2 Nitsche :

Nitsche[ArtNit71] methods are based on a different concept respect the **LMM**. In here the stress vector on the interface is computed from the stress field inside the solid body. The formulation usually includes a penalty term to avoid *ill-conditioning*, but in the same way as the **ALM** the constraint is enforced exactly and this term does not have effect in the final solution. The resulting formulation does not introduce additional **DOF**, but as it formulates the stresses from the displacement field, this method becomes complex to formulate on **NL** cases.

An extended overview of recent advances of the method applied in **CCM** can be found in the work of *Chouly*[ArtCho+17]. Here additional references can be found.

4.2.3.5.3 Other minor mentions :

We can mention additionally, some of them came directly from techniques applied on the **DDM**[BookWoh01] field. *i.e.* the **Finite Element Tearing and Interconnect (FETI)** methods[ArtDFS98] or monotone methods[ArtKor94; PhDKra01].

Another recent relevant contribution is the work of *Hiermeier*[ArtHWP18], who introduces a gradient based *Mortar* formulation, still based on **LMM** and **ALM**, which provides a symmetric system of equations to respect the *active-set* contributions. A symmetric **LHS** has advantages as it allows to consider certain types of iterative solvers such as the **Conjugate Gradient (CG)**.

4.2.4 Conclusion

After all the methods here introduced, we conclude that we will combine two methodologies. In the *Popp's* work[PhD-Pop12] and derived ones, the **DLMM** is considered in combination of a **Non-Linear Complementary Function (NCP)**. The resulting system is close to a **ALM** solution. We will take the dual shape functions considered on the **DLMM** in combination with the **ALM** approach used by *Cavaliere and Cardona*[ArtCFC12]. This means that we will work with **ALM** displacement only approach due to the consideration of **DLMM**, but without using the *Uzawa* iteration algorithm.

Additionally we will consider the **Adapted Augmented Lagrangian Method (AALM)** by *Busseta*[PhDBus; ArtBMP] in order to enhance the method. Check D.4.3.1. Adapted Augmented Lagrangian Method for further detail.

4.2.5 Frictional models

The frictional problem is quite complex by itself, but in this work we will focus on simple models that will allow to introduce the phenomenon in our simulations, not taking into account the dynamic effects of the friction [ArtPPS03]. Basically, the frictional behaviours are addressed in this work are limited to simple frictional laws as *Coulomb* and *Tresca*. The basic definitions of these frictional laws are such as the follow [BookWP99], Equation (4.1). In here we can see as the *Tresca* law depends only in a constant threshold parameter g , meanwhile in the case of *Coulomb* it depends on the normal reaction and the friction coefficient (μ).

$$(4.1) \quad \begin{cases} \text{Tresca: } \|F_T\| \leq g \\ \text{Coulomb: } \|F_T\| \leq \mu |F_N| \end{cases}$$

In addition to these frictional models, there exist more advanced frictional models, like the ones in Figure 4.10. These laws express that the friction no longer depends on the normal force when the latter surpasses a certain threshold. It is a kind of saturation of the friction threshold, which is often considered in metal forming. The *Coulomb* cone can become a truncated cone like in the *Coulomb-Orowan* and *Shaw* or a cylinder (*Tresca*). It is also possible to regularise the *Coulomb* law [BookWP99], this smoothes the law reducing part of the numerical problematics of the original piecewise function. *i.e.* *square root* regularisation, *hyperbolic tangent* or a *piecewise polynomial*.

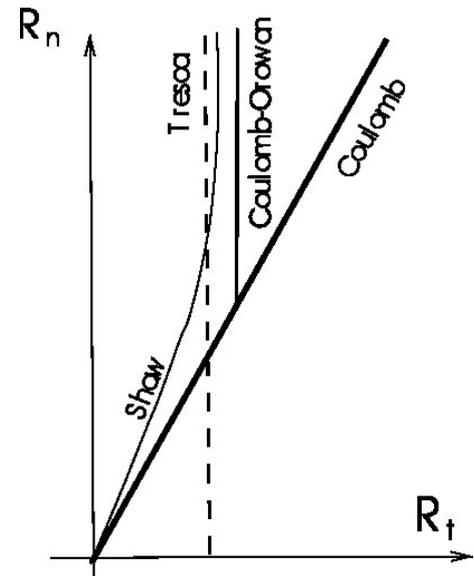


Figure 4.10: Friction cone for variants of the *Coulomb* law. Inspired [ArtRao16]

Besides of the pure frictional behaviour, the frictional models can incorporate additional effects, such as the *wear*, *adhesion* or variational evolutions in the friction coefficients [ArtRao16]. For more detailed frictional behaviours, we will address literature at the end of the section.

It is commonly admitted in the literature an analogy between plasticity and friction. In here the question of applicability of plasticity principles to frictional contact problems remains open. Indeed, the formulations of several frictional laws and its respective numerical procedures have been derived, many times taking into account on this analogy. The work of *Antoni* [ArtAnt17] analyses this phenomenon in detail.

Finally, we want to address some relevant works in the field of frictional behaviour in order to extend the short summary presented here. The book of *Popov* [BookPop10] is a very complete work about the friction in contact. We can also mention to more forming oriented manuscripts as *Boisse* [BookBAL03].

4.3 Formulation

4.3.1 Introduction

The following section presents the formulation employed in the derivation of the frictional *Mortar* contact condition formulation with **Augmented Dual Lagrange Multiplier (ADLM)**, based mainly in the work of *Alexander Popp* [PhDPop12; ArtPop+10], but also on *Cavaleri* and *Cardona* [ArtCC12; ArtCC13b] and the work of *Yastrebov* [PhDYas11; BookYas13]. In addition to the mentioned work, this author has incorporated its own consideration, which will be highlighted.

The contact mechanics problems are based on the **Initial Boundary Value Problem (IBVP)** of *non-linear* solid mechanics and the *unilateral* contact constraints. After recapitulating some basic notation and the strong formulation, a weak formulation of the contact mechanics problems with two subdomains will be introduced. In contrast to the

mesh tying case, see A.3.Mesh tying, *unilateral* contact leads to a constrained minimisation problem with inequality constraints, or more generally to so-called variational inequalities. It should be mentioned that both frictionless and frictional contact can either be formulated as variational inequalities with a constrained solution or as saddle point problems based on *Lagrange* multipliers, where the focus will be on the latter approach here.

The motivation for dual *Lagrange* multipliers[BookWoh01; ArtWoh02] lies in the fact that an extension of the master side basis functions to the slave side of the interface has a global support for standard *Lagrange* multipliers.

4.3.2 Definition of the problem

The basic fundamentals of a contact problem, featuring potentially large deformations and large sliding[ArtYL08; PhDPop12; BookWri06] will be formulated in the following lines (Figure 4.11). Reference configurations of two containing bodies are denoted by open sets Ω^1 and Ω^2 , and the deformations ϕ^1 and ϕ^2 of these two bodies are to be found. We call Γ_u where the *Dirichlet* boundary conditions are prescribed, Γ_σ where the *Neumann* boundary conditions are prescribed, and Γ_c where the contact constraints will be defined and enforced. The spatial counterparts are denoted as γ_u, γ_σ , and γ_c .

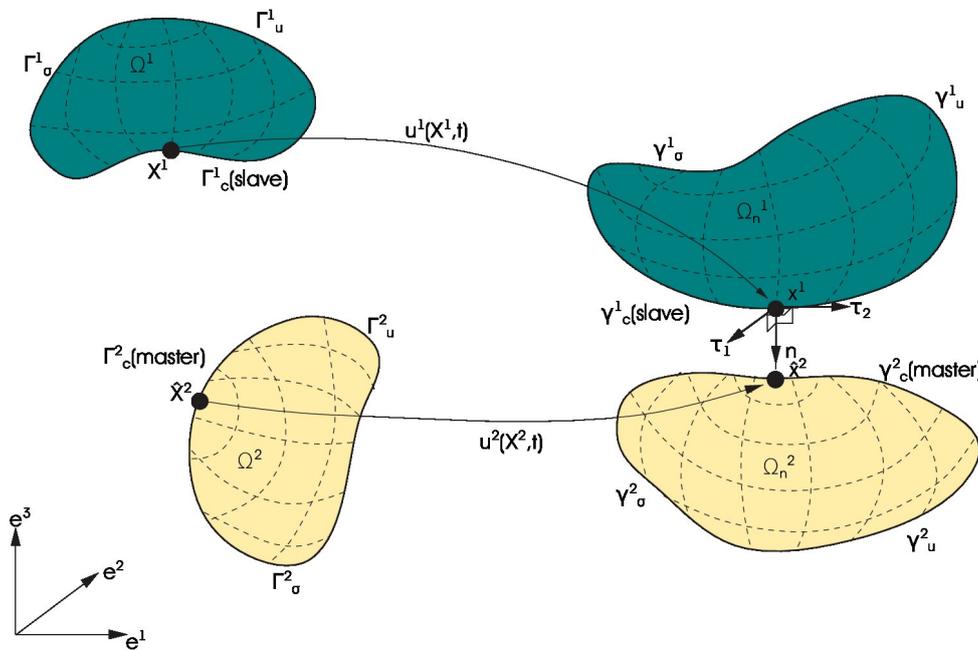


Figure 4.11: Definition of the contact problem

4.3.3 Frictionless contact

Before introduce the formulation employed for the frictional contact, the formulation for the frictionless will be introduced. Many terms are reused and common in both formulation, so the theoretical introduction for the frictional case will require a smaller effort.

4.3.3.1 Strong formulation

On each subdomain Ω_0^i , the **IBVP** of finite deformation elastodynamics needs to be satisfied, viz (4.2). In here we are adding the **BC** terms to the linear form from the **FE** chapter, 2.3.1.Linear form, and splitting in two different domains ($i = 1, 2$) the problem. This includes the *Dirichlet BC* in (4.2b), the *Neumann BC* from (4.2c) and the initial **BC** defined in (4.2d) and (4.2e).

$$\begin{aligned}
(4.2a) \quad & \nabla \cdot \boldsymbol{\sigma}^i + \mathbf{b}^i = \rho^i \ddot{\mathbf{u}}^i \text{ in } \Omega^i \times [0, T] \\
(4.2b) \quad & \mathbf{u}^i = \hat{\mathbf{u}}^i \text{ on } \Gamma_u^i \times [0, T] \\
(4.2c) \quad & \boldsymbol{\sigma}^i \cdot \mathbf{n}^i = \mathbf{t}^i \text{ on } \Gamma_\sigma^i \times [0, T] \\
(4.2d) \quad & \mathbf{u}^i(\mathbf{X}^i, 0) = \mathbf{u}_0^i(\mathbf{X}^i) \text{ in } \Omega_0^i \\
(4.2e) \quad & \dot{\mathbf{u}}^i(\mathbf{X}^i, 0) = \dot{\mathbf{u}}_0^i(\mathbf{X}^i) \text{ in } \Omega_0^i
\end{aligned}$$

The contact constraints in normal direction are typically given in form of **Hertz-Signorini-Moreau (HSM)** conditions as given in (4.3), and Figure 4.12. In optimisation theory these conditions are denominated usually as **KKT**⁵.

In the course of deriving a weak formulation, the balance of linear momentum at the unilateral contact problem for the interface Γ_c^i is typically exploited and a *Lagrange* multiplier vector field λ_n is introduced, thus setting the basis for a mixed variational approach. Unilateral contact constraints are typically formulated (and later also numerically evaluated) in the current configuration.

$$(4.3) \quad g_n \geq 0, p_n \leq 0, p_n g_n = 0 \text{ on } \Gamma_c^i \times [0, T]$$

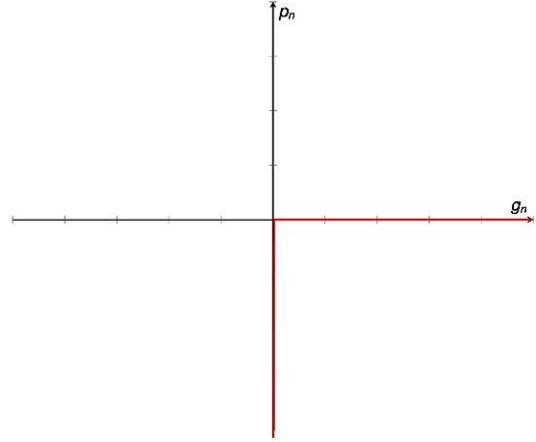


Figure 4.12: **KKT** conditions of non-penetration.

4.3.3.2 Weak formulation

We will distinguish two different cases for the weak formulation, the first one considering the **LM** as a scalar variable. The second one considering the **LM** decomposed in its *Cartesian* components, which means that the contact pressure can be represented as $\mathbf{n} \cdot \boldsymbol{\lambda}$.

4.3.3.2.1 Scalar *Lagrange* multiplier :

In this case the **LM** corresponds directly with the contact pressure, which is the complementary form of the normal gap (g_n). We present this form first, as it is simpler than the components form, where we will simply present the parts of the formulation changing respect the scalar form.

4.3.3.2.1.1 LMM :

The general theory for the **LMM** is presented and detailed in **D.3.Lagrange Multiplier method**. In the relative to the resolution of the contact problem, to start the derivation of a weak formulation of (4.2), appropriate solution spaces \mathcal{U}^i and weighting spaces \mathcal{V}^i need to be defined as (4.4).

$$(4.4) \quad \begin{cases} \mathcal{U}^i = \{ \mathbf{u}^i \in H^1(\Omega) \mid \mathbf{u}^i = \hat{\mathbf{u}}^i \text{ on } \Gamma_u^i \}, \\ \mathcal{V}^i = \{ \delta \mathbf{u}^i \in H^1(\Omega) \mid \delta \mathbf{u}^i = \mathbf{0} \text{ on } \Gamma_u^i \} \end{cases}$$

Additionally the *Lagrange* multiplier vector $\lambda_n = \lambda_n \cdot \mathbf{n} = -\mathbf{t}_c^1$, which enforce the unilateral contact constraint(4.3), represents the negative slave side contact traction \mathbf{t}_c^1 , is chosen from a corresponding solution space denoted as \mathcal{M} .

In terms of its classification in functional analysis, this space represents the dual space of the trace space \mathcal{L}^1 of \mathcal{V}^1 . In the given context, this means that $\mathcal{M} = H^{1/2}(\Gamma_c)$ and $\mathcal{L}^1 = H^{1/2}(\Gamma_c)$, where \mathcal{M} and \mathcal{L}^1 denote single scalar components of the corresponding vector-valued spaces \mathcal{M} and \mathcal{W} .

⁵Moreau[ArtMor73] formulated this on 1974 expressed as sub-gradients. The expression from Karush[ArtKar39] is previous, from 1939.

Based on these considerations, a saddle point type weak formulation is derived next. This can be done by extending the standard weak formulation of non-linear solid mechanics as defined to two subdomains and combining it with the *Lagrange* multiplier coupling terms introduced in generic form. Find $\mathbf{u}^i \in \mathcal{U}^i$ and $\lambda_n \in \mathcal{M}$ such that we obtain (4.5a), then once derived (4.6).

$$(4.5a) \quad \mathcal{L}_{co}(\mathbf{u}, \lambda_n) = \int_{\Gamma_c^1} \lambda_n \cdot \mathbf{g}_n d\Gamma_{co}^i$$

Where \mathbf{g}_n is the continuous normal gap, that can be defined as (4.5b).

$$(4.5b) \quad \mathbf{g}_n = \mathbf{n}^1 \cdot (\mathbf{u}^1 - \mathbf{u}^2)$$

$$(4.6a) \quad \delta \mathcal{L}(\mathbf{u}, \lambda_n) = \delta \mathcal{L}_V + \delta \mathcal{L}_M$$

$$(4.6b) \quad \delta \mathcal{L}_V = -\delta \mathcal{L}_{kin}(\mathbf{u}^i, \delta \mathbf{u}^i) - \delta \mathcal{L}_{int,ext}(\mathbf{u}^i, \delta \mathbf{u}^i) - \delta \mathcal{L}_{co}(\lambda^i, \delta \mathbf{u}^i) = 0 \quad \forall \delta \mathbf{u}^i \in \mathcal{V}$$

$$(4.6c) \quad \delta \mathcal{L}_M = -\delta \mathcal{L}_\lambda(\mathbf{u}^i, \delta \lambda^i) \geq 0 \quad \forall \delta \lambda^i \in \mathcal{M}$$

Herein, the kinetic contribution $\delta \mathcal{L}_{kin}$, the internal and external contributions $\delta \mathcal{L}_{int,ext}$ and the unilateral contact contribution $\delta \mathcal{L}_{co}$ to the overall virtual work on the two subdomains, as well as the weak form of the unilateral contact constraint $\delta \mathcal{L}_\lambda$, have been abbreviated as (4.7).

$$(4.7a) \quad -\delta \mathcal{L}_{kin}(\mathbf{u}) = \sum_{i=1}^2 \left[\int_{\Omega^i} \rho^i \dot{\mathbf{u}}^i \cdot \delta \mathbf{u}^i d\Omega^i \right]$$

$$(4.7b) \quad -\delta \mathcal{L}_{int,ext}(\mathbf{u}) = \sum_{i=1}^2 \left[\int_{\Omega^i} \left(\frac{\partial \sigma^i}{\partial x_j} : \delta \dot{\mathbf{u}}^i + \sigma^i : \frac{\partial \delta \dot{\mathbf{u}}^i}{\partial x_j} - \mathbf{b} \cdot \delta \mathbf{u}^i \right) d\Omega^i - \int_{\Gamma_\sigma^i} \mathbf{t}^i \cdot \delta \mathbf{u}^i d\Gamma_\sigma^i \right]$$

$$(4.7c) \quad -\delta \mathcal{L}_{co}(\mathbf{u}, \lambda_n) = \int_{\Gamma_c^1} \lambda_n \cdot \delta \mathbf{g}_n d\Gamma_{co}^1$$

$$(4.7d) \quad -\delta \mathcal{L}_\lambda(\mathbf{u}, \lambda_n) = \int_{\Gamma_c^1} \delta \lambda_n \cdot \mathbf{g}_n d\Gamma_{co}^1$$

The coupling terms on Γ_c also allow for a direct interpretation in terms of variational formulations and the principle of virtual work. Whereas the contribution in (4.7c) represents the virtual work of the unknown interface tractions $\lambda = \mathbf{t}_c^1 = \mathbf{t}_c^2$, the contribution in (4.7d) ensures a weak, variationally consistent enforcement of the unilateral contact constraint (4.3). Nevertheless, the concrete choice of the discrete *Lagrange* multiplier space \mathcal{M}_h in the context of mortar finite element discretisations is decisive for the stability of the method and for optimal a priori error bounds. Finally, it is pointed out that the weak formulation (4.6b) and (4.6c) possesses all characteristics of saddle point problems and *Lagrange* multiplier methods.

In contrast to the mesh tying case, where this mapping only came into play in the discrete setting, γ_c^1 and γ_c^2 cannot even be guaranteed to be identical in the continuum framework for unilateral contact, because they not only comprise the actual contact surfaces but the potential contact surfaces.

As compared with the mesh tying case, it is noticeable that the weak formulation contains inequality (4.6c) conditions for unilateral contact. These require a particular numerical treatment based on active set strategies.

4.3.3.2.1.2 Penalty :

This optimisation method is better detailed in the corresponding section of the appendix, see D.2.Penalty method. The main advantages of this method is that does not require to add additional **DOF** to the system of equations, reducing the system of equations to be solved and that it does not introduce *ill-conditioning* problems to the system

of equations resolution due to the introduction of *saddle point* in the formulation. The main disadvantage is that it never achieves the exact solution as this would require the consideration of an infinite penalty, which will induce ill-conditioning in the system. Therefore the main problem of this method is the chosen of the proper penalty value, enough to solve approximately the problem without the cited problematic.

The main difference from the formulation presented in (4.7), is the lack of (4.7d), as our formulation will no deal with **LM**. Additionally (4.7c) will be rewritten as in (4.8b), which comes from (4.8a). In here the ε_n is a positive penalty parameter corresponding to the normal contact.

$$(4.8a) \quad -\mathcal{L}_{co}(\mathbf{u}) = \frac{1}{2} \int_{\Gamma_c^1} \varepsilon_n \cdot \delta g_n^2 d\Gamma_{co}^1$$

$$(4.8b) \quad -\delta \mathcal{L}_{co}(\mathbf{u}) = \int_{\Gamma_c^1} \varepsilon_n \cdot \delta g_n d\Gamma_{co}^1$$

4.3.3.2.1.3 ALM :

One of the main disadvantages of the standard *Lagrange* multiplier is the *saddle point* problem that appears in the formulation. For solving that, an **Augmented Lagrangian** method to solve contact problems with friction was proposed by *Alart and Curnier*[ArtAC91] based on a reformulation of the contact and friction laws into a system of equations without inequalities. More details about **ALM** can be found in the optimisation appendix D.4. **Augmented Lagrange Multiplier method**. The resulting *Lagrangian* (\mathcal{L}) can be seen as a combination of the standard **LM Lagrangian** (4.3.3.2.1.1.LMM) and the penalty *Lagrangian* (4.3.3.2.1.2.Penalty).

Focusing in the functional relative to the contact ($\mathcal{L}_{co}(\mathbf{u}, \lambda_n) = \mathcal{L}_{\mathcal{V}co} + \mathcal{L}_{\mathcal{M}}$), we can rewrite (4.5a) as (4.9).

$$(4.9) \quad \mathcal{L}_{co}(\mathbf{u}, \lambda_n) = \int_{\Gamma_c^1} k \lambda_n \cdot g_n + \frac{\varepsilon}{2} g_n^2 - \frac{1}{2\varepsilon} \langle k \lambda_n + \varepsilon g_n \rangle^2 d\Gamma_{co}^1$$

Where ε is a positive penalty parameter, k is a positive scale factor, and $\langle \cdot \rangle$ is the *Macauley* bracket operator, that is (4.10).

$$(4.10) \quad \langle x \rangle \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

This functional is \mathcal{C}^1 differentiable *saddle-point*, as shown in Figure D.4, from D.4.3. **Applicability on contact problems**. The solution is obtained as the set of values that render this functional stationary.

The solution does not depend on the value of parameters ε and k . Nevertheless, the convergence rate does depend on their value. In numerical computations, default values of ε and k are selected in terms of a mean value of the *Young* modulus (E) of the bodies in contact and of a mean value of mesh size, as (4.11). Numerical examples show that this choice gives a better condition number of the iteration matrix than other choices. This is proven later at 4.3.3.3. **Augmented Lagrange multiplier parameters calibration**, where a numerical experiment is provided, to illustrate on the influence of these parameters on the convergence properties.

$$(4.11) \quad \varepsilon = k \approx 10 \frac{E_{mean}}{h_{mean}}$$

The functional (4.9) can be separated in two different parts, as can be seen in (4.12).

$$(4.12) \quad \mathcal{L}_{co}(\mathbf{u}, \lambda_n) = \int_{\Gamma_c^1} \begin{cases} k \lambda_n \cdot g_n + \frac{\varepsilon}{2} g_n^2 d\Gamma_{co}^1 & \text{if } k \lambda_n + \varepsilon g_n \leq 0 \text{ (Contact zone)} \\ -\frac{k}{2\varepsilon} \lambda_n^2 & \text{if } k \lambda_n + \varepsilon g_n > 0 \text{ (Gap zone)} \end{cases} d\Gamma_{co}^1$$

Finally, we can derive (4.12) to obtain the variational form from (4.13), where to simplify we define the augmented normal pressure $\bar{\lambda}_n = k\lambda_n + \varepsilon g_n$.

$$(4.13) \quad \delta \mathcal{L}_{co}(\mathbf{u}, \lambda_n) = \int_{\Gamma_c^1} \begin{cases} \bar{\lambda}_n \cdot \delta \mathbf{g}_n + k g_n \delta \lambda_n & \text{if } \bar{\lambda}_n \leq 0 \text{ (Contact zone)} \\ -\frac{k^2}{\varepsilon} \lambda_n \delta \lambda_n & \text{if } \bar{\lambda}_n > 0 \text{ (Gap zone)} \end{cases} d\Gamma_{co}^1$$

The functional from (4.13) makes that the system obtained varies in function if the nodes are present in the contact or the gap zone, so the system is not a priori known and in the following to present the numerical discretisation will focus in the solution obtained in the gap zone. Once this is derived the solution in the gap zone can be obtained in a straightforward way.

4.3.3.2.2 Vector Lagrange multiplier :

Here the vectorised, or by components, of frictionless contact formulation is presented. The main reason in order to define it by this form is that the resulting system of equations can be statically condensed, therefore removing the corresponding **LM DOF** from the system of equations, and by this solving a system of equations purely based on displacement **DOF**. This is done taking in consideration properties from the dual *Lagrange* multiplier which will be presented later on 4.3.3.4.1. *Dual Lagrange multipliers*. Of course the solution for the penalty method, 4.3.3.2.1.2. *Penalty*, remains identical, as there is not any **LM** taking roles on the formulation.

The main modification which takes place on this formulation is the replacement of the contact pressure **LM**, or λ_n , by the consideration of a **LM** defined in the *Cartesian* components, represented as λ , where the normal components (λ_n) are different or equal to zero, and the tangential ones (λ_τ) are always zero. This can be summarised in (4.14).

$$(4.14) \quad \begin{cases} \lambda_n = \mathbf{n} \cdot \lambda \\ \lambda_\tau = \lambda - \mathbf{n} \cdot (\mathbf{n} \cdot \lambda) = \mathbf{0} \end{cases}$$

4.3.3.2.2.1 LMM :

From the formulation stated at 4.3.3.2.1.1. **LMM**, we modify the terms from (4.7) related to the **LM**, resulting (4.15). (4.7c) will be reformulated as (4.15a) and (4.7d) into (4.15b).

$$(4.15a) \quad \delta \mathcal{L}_{co}(\mathbf{u}, \lambda) = \int_{\Gamma_c^1} \lambda \cdot (\delta \mathbf{u}^1 - \delta \mathbf{u}^2) d\Gamma_{co}^1$$

Where $\hat{\mathbf{u}}^2$ are the displacement from the master side projected on the slave side.

$$(4.15b) \quad \delta \mathcal{L}_\lambda(\mathbf{u}, \lambda) = \int_{\Gamma_c^1} \delta(\mathbf{n} \cdot \lambda) \cdot g_n - (\lambda - \mathbf{n} \cdot \lambda) (\delta \lambda - \mathbf{n} \cdot \delta \lambda) d\Gamma_{co}^1$$

4.3.3.2.2.2 ALM :

Taking as base the solution presented on 4.3.3.2.1.3. **ALM**, and the modifications introduced in the previous section, we will need to define the augmented **LM** by components, as defined in (4.16).

$$(4.16) \quad \bar{\lambda} = k\lambda + \varepsilon \mathbf{n} g_n$$

With this consideration, for the contact contributions of the potential, (4.12) will be reformulated as (4.17a) and (4.13) into (4.17b) respectively. Additionally, (4.7c), which defines the **LM** contribution will be rewritten as (4.17c). On these equations we will consider $\lambda_n = k(\mathbf{n} \cdot \boldsymbol{\lambda}) + \varepsilon g_n$ to simplify the expressions.

$$(4.17a) \quad \mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = \int_{\Gamma_c^1} \begin{cases} \bar{\lambda}_n \cdot (\mathbf{u}^1 - \mathbf{u}^2) + \frac{\varepsilon}{2} g_n^2 d\Gamma_{co}^i & \text{if } \bar{\lambda}_n \leq 0 \text{ (Contact zone)} \\ -\frac{k}{2\varepsilon} \boldsymbol{\lambda}^2 & \text{if } \bar{\lambda}_n > 0 \text{ (Gap zone)} \end{cases} d\Gamma_{co}^1$$

$$(4.17b) \quad \delta \mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = \int_{\Gamma_c^1} \begin{cases} \bar{\lambda}_n \cdot (\delta \mathbf{u}^1 - \delta \mathbf{u}^2) + k g_n \delta \boldsymbol{\lambda} \cdot \mathbf{n} & \text{if } \bar{\lambda}_n \leq 0 \text{ (Contact zone)} \\ -\frac{k^2}{\varepsilon} \boldsymbol{\lambda} \delta \boldsymbol{\lambda} & \text{if } \bar{\lambda}_n > 0 \text{ (Gap zone)} \end{cases} d\Gamma_{co}^1$$

$$(4.17c) \quad \delta \mathcal{L}_{\lambda}(\mathbf{u}, \boldsymbol{\lambda}) = \int_{\Gamma_c^1} k(\mathbf{n} \cdot \delta \boldsymbol{\lambda}) \cdot g_n - \frac{k^2}{\varepsilon} (\boldsymbol{\lambda} - \mathbf{n} \cdot \boldsymbol{\lambda})(\delta \boldsymbol{\lambda} - \mathbf{n} \cdot \delta \boldsymbol{\lambda}) d\Gamma_{co}^1$$

4.3.3.3 Augmented Lagrange multiplier parameters calibration

The expression from Equation (4.11) is taken directly from literature [ArtCC12]. In here we present a simple patch test [ArtTP] example, in order to show the influence of k and ε in the condition number (κ) of the **LHS**. We will see as the ε affects always negatively to κ , this is proven in the optimisation appendix at D.7.2.2.3. Augmented Lagrange multiplier method, meanwhile k can improve or worsen depending on the range value considered.

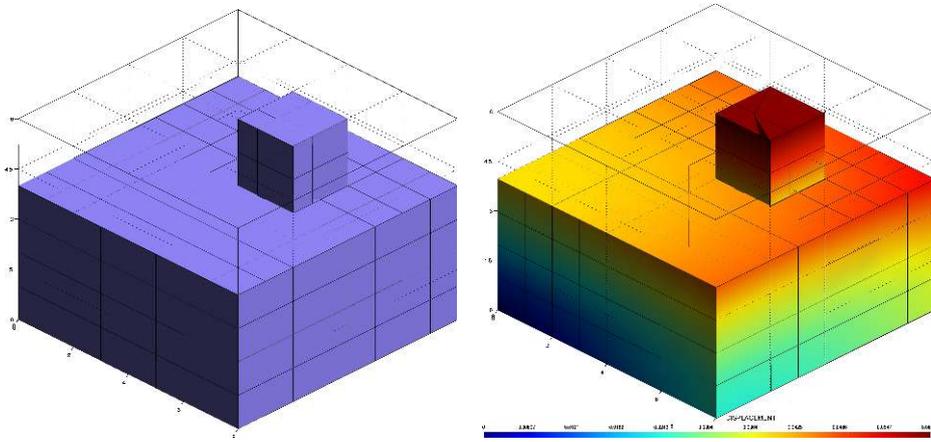
E Solid 1	ν Solid 1	E Solid 2	ν Solid 2
100Pa	0.3	100Pa	0.3

Table 4.1: Parameters considered for **ALM** parameters calibration

The properties of the materials considered are listed on Table 4.1, these properties are quite simple numbers, this will allow later to compute easily the **ALM** parameters. Taking (4.11) and with the corresponding $h \approx 10$, our reference values will correspond with $\varepsilon = k = 100$. In addition to the load considered in the top face of the punch block is equal to 1Pa.

$$(4.18a) \quad \kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

$$(4.18b) \quad \kappa(A) = \frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}$$



(a) Mesh of the patch test

(b) Displacement solution for the

Figure 4.13: Condition number study for the **ALM**

The condition number (κ) of a function measures how much the output value of the function can change for a small change in the input argument. A problem with a low condition number is said to be **well-conditioned**, while a problem with a high condition number is said to be **ill-conditioned**. In order to compute the corresponding κ we should evaluate the **Singular Value Decomposition (SVD)**, in order to compute the maximal and minimal singular values required in (4.18a). This expression can be simplified if A is **normal** ($A^*A = AA^*$), in that case we can simply compute κ considering the maximal and minimal eigenvalues as shown in (4.18b)⁶. This last approach simplifies considerably the numerical effort required, as we dispose of optimal methods for the extreme eigenvalues, like the *power iteration* for the greatest eigenvalue and the *inverse power iteration* for the smallest one respectively.

In Figure 4.13 we can see the proposed mesh for the 3D *Taylor patch test* (Figure 4.13a), as well as the displacement solution of the problem (Figure 4.13b).

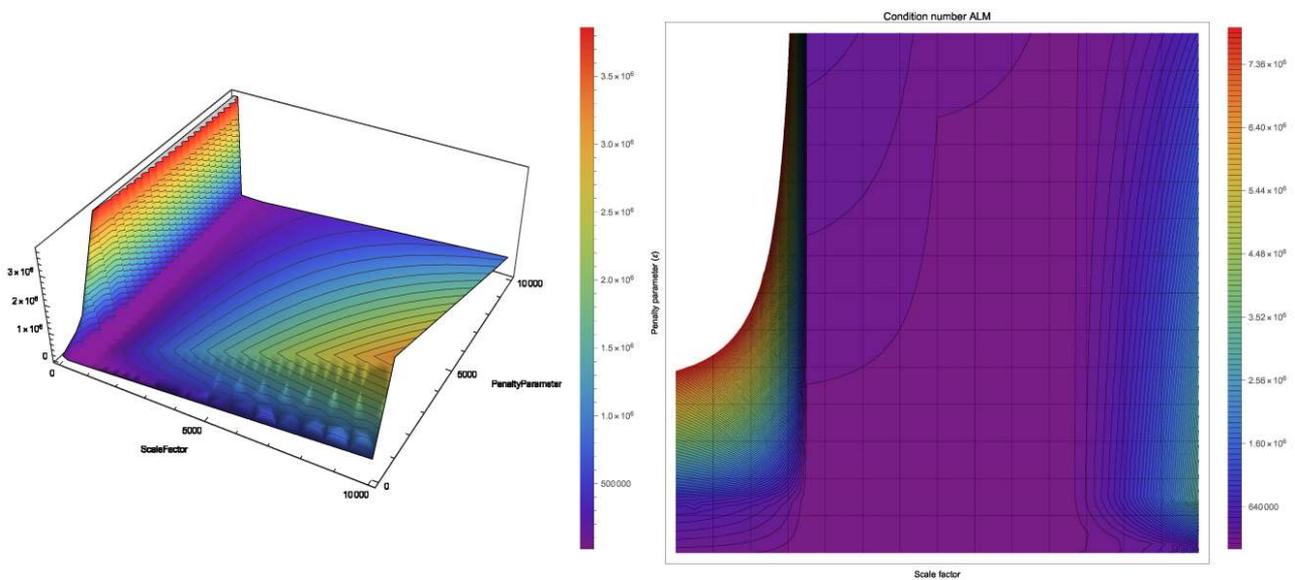


Figure 4.14: Condition number study graphic representation

Finally, the Figure 4.14 represents graphically the results from Table 4.2. We represent in the left the surface plot without any additional consideration, on the other hand, on the contour plot the k axis considers a logarithmic scale. The latter is due to the fact that the ε value is always increasing κ and therefore we do not require to consider a more detailed scale for this variable, but in the case of k it provides us relevant information taking into account that we modify the values in powers of 10 (Table 4.2). In this continuous representation, we can appreciate several things. First of all, as previously stated, is that the ε increases always the condition number, and the k may improve or not the κ depending of the value. Second of all we can see, particularly in the right figure like the value estimated from Equation (4.11), 100 – 100, provides the best conditioning in overall.

4.3.3.4 Discretisation and numerical integration

4.3.3.4.1 Dual *Lagrange* multipliers :

4.3.3.4.1.1 Definition :

The discretisations of the displacements correspond with the standard ones in the finite element formulation, for more information check the literature[BookZZT13]. In addition, an adequate discretisation of the *Lagrange* multiplier vector λ is needed, and will be based on a discrete *Lagrange* multiplier space \mathcal{M}_h being an approximation of \mathcal{M} . Thus, we can define the discrete *Lagrange* multiplier as (4.19), with the shape functions Φ_j and the discrete nodal

⁶We are considering **SVD** as it is the proper definition of the condition number, but in fact this is a very expensive operation and can be applied only in small systems. Another alternative if A is symmetric, then it is possible to compute the ratio between the max and min eigenvalues of A . In the case that A is not symmetric it is possible to compute the max and min eigenvalues of $\sqrt{A^T A}$.

k	ε	κ	k	ε	κ	k	ε	κ
1	1.00E-12	1.13E+05	10	1000	1.74E+04	1000	10	1.92E+04
1	1.00E-02	1.13E+05	10	10000	2.69E+05	1000	100	1.81E+04
1	1	1.13E+05	100	1.00E-12	1.74E+04	1000	1000	3.14E+04
1	10	1.18E+05	100	1.00E-02	1.74E+04	1000	10000	6.43E+04
1	100	1.63E+05	100	1	1.74E+04	10000	1.00E-12	6.64E+05
1	1000	6.15E+05	100	10	1.74E+04	10000	1.00E-02	6.63E+05
1	10000	2.69E+07	100	100	1.74E+04	10000	1	6.63E+05
10	1.00E-12	1.74E+04	100	1000	1.74E+04	10000	10	6.63E+05
10	1.00E-02	1.74E+04	100	10000	9.38E+04	10000	100	3.53E+05
10	1	1.74E+04	1000	1.00E-12	6.60E+04	10000	1000	3.26E+06
10	10	1.74E+04	1000	1.00E-02	6.60E+04	10000	10000	7.38E+05
10	100	1.74E+04	1000	1	5.75E+05			

Table 4.2: Results of numerical experiment for **ALM** parameters

Lagrange multipliers λ_h .

$$(4.19) \quad \lambda_h = \sum_{i=1}^{m^1} \Phi_j(\xi^1, \eta^1) \lambda_j$$

Details on how to define dual *Lagrange* multiplier shape functions Φ_j using the so-called biorthogonality relationship with the standard displacement shape functions N_k have first been presented in *Wohlmuth*[ArtWoh02]. A common notation of the biorthogonality condition is (4.20), where $\Gamma_{co,h}^1$ represents the discrete contact interface.

$$(4.20) \quad \int_{\Gamma_{co,h}^1} \Phi_j N_k^1 d\Gamma_{co}^i = \delta_{jk} \int_{\Gamma_{co,h}^1} N_k^1 d\Gamma_{co}^i, j, k = 1, \dots, m^1$$

Herein, δ_{jk} is the *Kronecker* delta, and the most common choice $m^1 = n^1$ is assumed. For practical reasons, the biorthogonality condition is typically applied locally on each slave element, represented with the index e , yielding (4.21), where m_e^1 represents the number of *Lagrange* multiplier nodes of the considered slave element.

$$(4.21) \quad \int_e \Phi_j N_k^1 de = \delta_{jk} \int_e N_k^1 de, j, k = 1, \dots, m_e^1$$

Combining the biorthogonality condition in (4.21) and the partition of unity property of the dual shape functions, it follows that (4.22).

$$(4.22) \quad \int_e \Phi_j de = \int_e N_j^1 de, j = 1, \dots, m_e^1$$

It is important to point out that the element-wise biorthogonality condition in (4.21) must be satisfied in the physical space, and not simply in the finite element parameter space. Consequently, a matrix system of size $m_e^1 \times m_e^1$ must be solved on each slave element. The first step for doing this is to introduce unknown linear coefficients a_{jk} such that (4.23).

$$(4.23) \quad \Phi_j(\xi, \eta) = a_{jk} N_k^1(\xi, \eta), \mathbf{A}_e = [a_{jk}] \in \mathbb{R}^{m_e^1 \times m_e^1}$$

It can easily be verified that, as a second step, insertion of (4.23) into (4.21) yields the unknown coefficient matrix \mathbf{A}_e as (4.24), where $J(\xi, \eta)$ is the slave *Jacobian* determinant.

$$(4.24) \quad \begin{aligned} \mathbf{A}_e &= \mathbf{D}_e \mathbf{M}_e^{-1} \\ \mathbf{D}_e &= [d_{jk}] \in \mathbb{R}^{m_e^1 \times m_e^1}, d_{jk} = \delta_{jk} \int_e N_k^1(\xi, \eta) J(\xi, \eta) d\mathbf{e} \\ \mathbf{M}_e &= [m_{jk}] \in \mathbb{R}^{m_e^1 \times m_e^1}, m_{jk} = \int_e N_j^1(\xi, \eta) N_k^1(\xi, \eta) J(\xi, \eta) d\mathbf{e} \end{aligned}$$

4.3.3.4.1.2 Graphical representation :

The dual shape functions presented on the Equation (4.25) for the linear line is presented on the Figure 4.15. In the case of the linear triangles the dual shape functions from (4.26) is represented on Figure 4.16. Finally, the bilinear quadrilateral shape functions from (4.27) is shown in Figure 4.17. These graphic representations can be compared with the standard shape functions, these are presented at the end of this chapter, which defines the shape functions derivatives, see 4.6. [Derivatives for contact mechanics linearization](#).

$$(4.25) \quad \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1 - \xi) \\ \frac{1}{2}(1 + \xi) \end{bmatrix}$$

$$(4.26) \quad \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} = \begin{bmatrix} 1 - \xi - \eta \\ \xi \\ \eta \end{bmatrix}$$

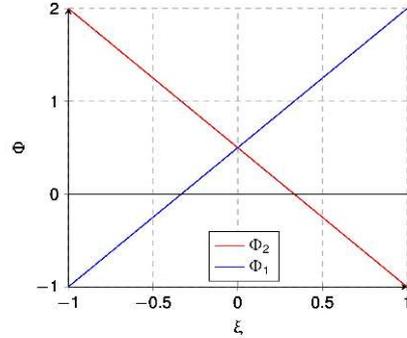


Figure 4.15: Dual shape functions for the 2D linear line

$$(4.27) \quad \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4}((1 - \xi)(1 - \eta)) \\ \frac{1}{4}((1 + \xi)(1 - \eta)) \\ \frac{1}{4}((1 + \xi)(1 + \eta)) \\ \frac{1}{4}((1 - \xi)(1 + \eta)) \end{bmatrix}$$

4.3.3.4.1.3 Derivatives :

To define $\Delta\phi$ it is necessary to define the derivatives from (4.24), which can be obtained with (4.99), which can be found in the corresponding section of the derivatives section of this chapter, see 4.6.1.3. [Dual shape functions](#).

4.3.3.4.2 Mortar operators :

4.3.3.4.2.1 Definition :

Considering the discrete *Lagrange* multiplier (4.19) in (4.6b) we obtain (4.28), where χ is the interface mapping. This mapping, or projection, between domains is shown in the derivatives section, 4.6. [Derivatives for contact mechanics linearization](#), in the 2D case in the Figure 4.95 and in the 3D case in the Figure 4.100.

$$(4.28) \quad -\delta \mathcal{L}_{co,h} = \sum_{j=1}^{m^1} \sum_{k=1}^{n^1} \lambda_{nj}^T \left(\int_{\Gamma_{c,h}^1} \Phi_j N_k^1 d\Gamma_{co}^i \right) \delta \mathbf{d}_{nk}^1 - \sum_{j=1}^{m^1} \sum_{l=1}^{n^2} \lambda_{lj}^T \left(\int_{\Gamma_{c,h}^1} \Phi_j (N_l^2 \circ \chi_h) d\Gamma_{co}^i \right) \delta \mathbf{d}_{nl}^2$$

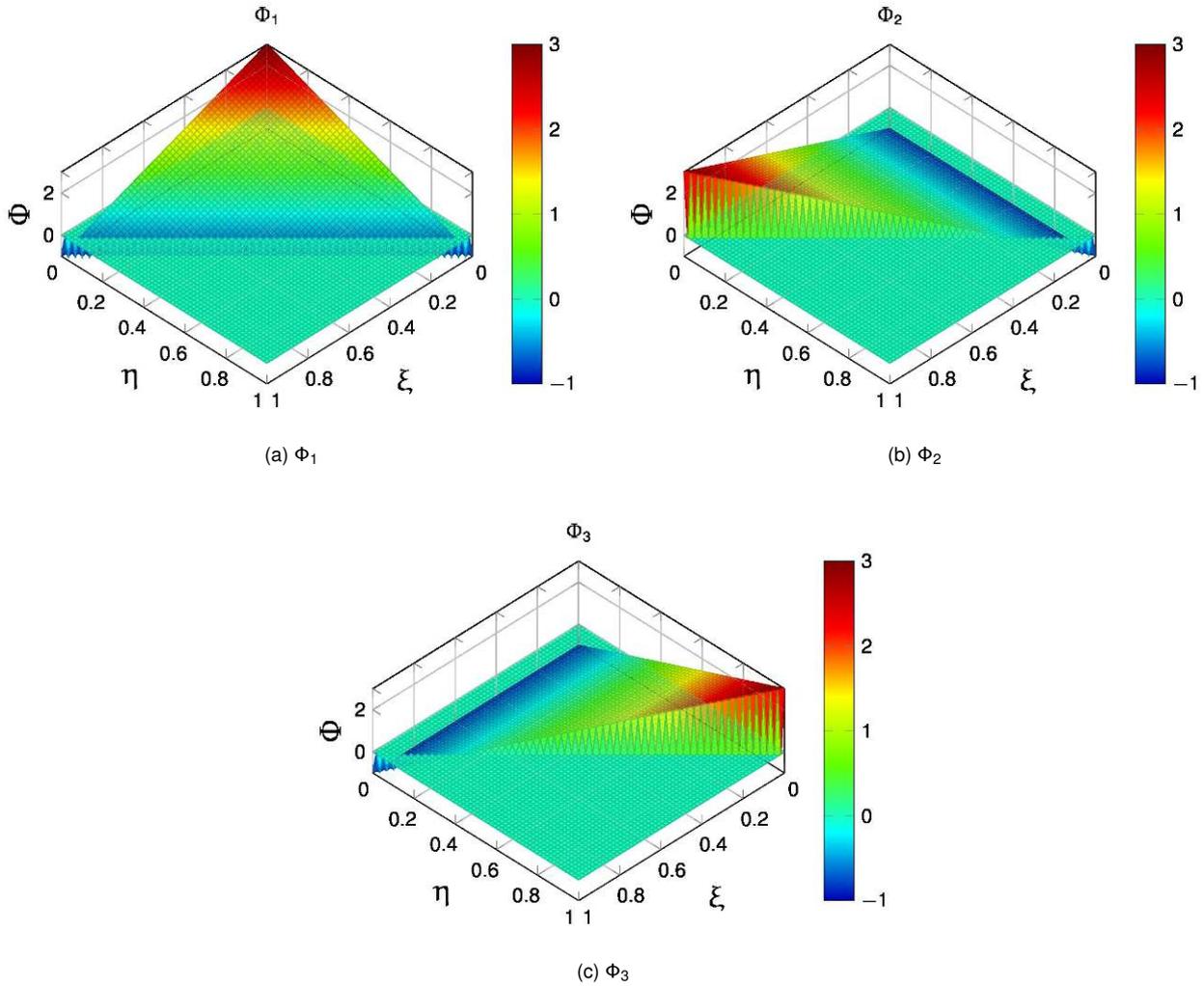


Figure 4.16: Dual shape functions for the 3D linear triangle

Numerical integration of the mortar coupling terms is exclusively performed on the slave side $\Gamma_{c,h}$ of the interface. In (4.28), nodal blocks of the two mortar integral matrices commonly denoted as \mathbf{D}_e and \mathbf{M}_e can be identified. This leads to the following definitions (4.29).

$$(4.29) \quad \begin{aligned} \mathbf{D}[j, k] &= D_{jk} \mathbf{I}_{ndim} = \int_{\Gamma_{c,h}^1} \Phi_j N_k^1 d\Gamma_{co}^i \mathbf{I}_{ndim}, \quad j = 1, \dots, m^1, \quad k = 1, \dots, n^1 = \sum_{g=1}^{n_{gp}} w_g \phi_{gj} N_{gk}^1 \mathcal{J}_g^1 \\ \mathbf{M}[j, l] &= M_{jl} \mathbf{I}_{ndim} = \int_{\Gamma_{c,h}^1} \Phi_j (N_l^2 \circ \chi_h) d\Gamma_{co}^i \mathbf{I}_{ndim}, \quad j = 1, \dots, m^1, \quad k = 1, \dots, n^2 = \sum_{g=1}^{n_{gp}} w_g \phi_{gj} N_{gk}^2 \mathcal{J}_g^1 \end{aligned}$$

With these matrices we can express the functional (4.28) in the following way (4.30) for the standard *Lagrange* multiplier. This definition can be translated into the **ALM** formulation, but here we present the reasoning for the standard *Lagrange* multiplier only, so we can deduce the algebraic calculation of the discrete nodal weighted gap. In here, \mathbf{x}_n corresponds with the normal component of the coordinates of all the nodes, meanwhile $\mathbf{x}_{n,S}$ and $\mathbf{x}_{n,M}$

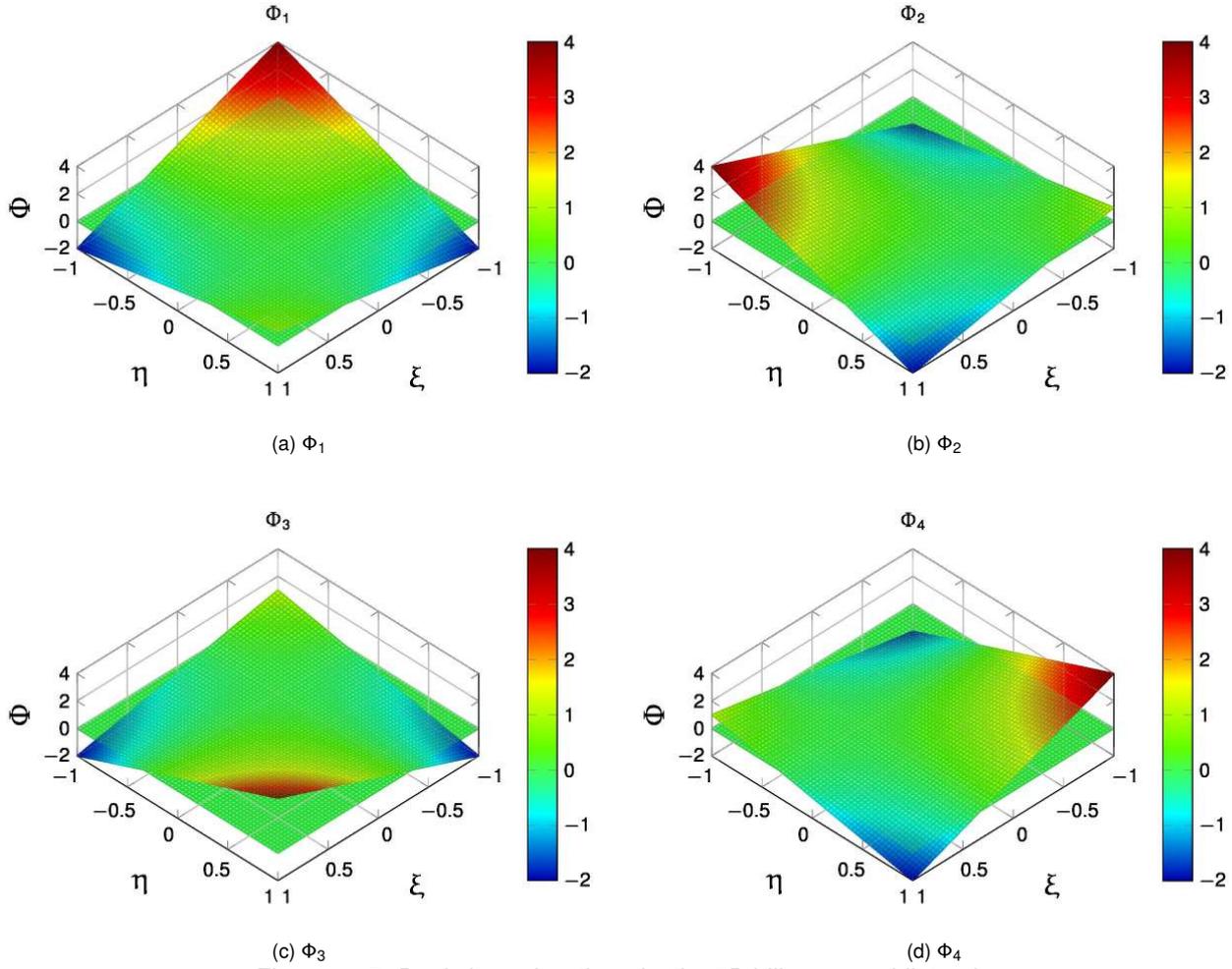


Figure 4.17: Dual shape functions for the 3D bilinear quadrilateral

correspond with the slave and master domain nodes respectively.

$$(4.30) \quad -\delta\mathcal{L}_{co,h} = \delta\mathbf{x}_{nS}^T \mathbf{D}^T \lambda_n - \delta\mathbf{x}_{nM}^T \mathbf{M}^T \lambda_n = \delta \begin{bmatrix} \mathbf{x}_{nN} & \mathbf{x}_{nM} & \mathbf{x}_{nS} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^T \\ \mathbf{D}^T \end{bmatrix} = \delta\mathbf{x}_n \underbrace{\begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^T \\ \mathbf{D}^T \end{bmatrix}}_{\mathbf{B}_{co}^T} \lambda_n = \delta\mathbf{x}_n^T \mathbf{f}_{co}(\lambda_n)$$

Herein, the discrete mortar unilateral contact operator \mathbf{B}_{co} and the resulting discrete vector of unilateral contact forces $\mathbf{f}_{co}(\lambda_n) = \mathbf{B}_{co} \lambda_n$ acting on the slave and the master side of the interface are introduced.

To finalise the discretisation of the considered unilateral contact problem, a closer look needs to be taken at the weak constraint contribution $\delta\mathcal{L}_{\lambda,h}$ in (4.6c). Due to the saddle point characteristics and resulting symmetry of the mixed variational formulation in (4.6b) and (4.6c), all discrete components of $\delta\mathcal{L}_{\lambda,h}$ have already been introduced and the final formulation is given as (4.31), with $\mathbf{g}_n(\mathbf{x}) = \mathbf{B}_{mt} \mathbf{x} \cdot \mathbf{n}$ representing the discrete unilateral contact constraint at the coupling interface.

$$(4.31) \quad -\delta\mathcal{L}_{\lambda,h} = \delta\lambda_n^T \mathbf{D} \mathbf{x}_{nS} - \delta\lambda_n^T \mathbf{M} \mathbf{x}_{nM} = \delta\lambda_n^T \mathbf{B}_{mt} \mathbf{x} \cdot \mathbf{n} = \delta\lambda_n^T \mathbf{g}_n(\mathbf{x})$$

Where, this discrete form of \mathbf{g}_n can be renamed as nodal weighted gap for each node (\tilde{g}_n).

4.3.3.4.2 Derivatives :

To obtain a fully quadratic convergence in the computation of the contact problem we should compute the derivatives of the *Mortar* operators, in consequence the derivatives of the *Mortar* operators are defined in its corresponding section at the end of this chapter, see 4.6. [Derivatives for contact mechanics linearization](#).

4.3.3.4.3 Algebraic form of the problem :

We will consider two cases for the matrix representation of the problem. The first one, corresponding to the scalar **LM**, presented in the corresponding section 4.3.3.2.1. [Scalar Lagrange multiplier](#). This one cannot be condensed, and therefore requires the direct resolution of the problem with all the **DOF**.

The second case corresponds with the vector representation of the **LM**, from the formulation presented in 4.3.3.2.2. [Vector Lagrange multiplier](#). This resulting system can be condensed due to the properties of the **D**, which is diagonal due to the consideration of the dual *Lagrange* multipliers, presented in 4.3.3.4.1. [Dual Lagrange multipliers](#).

4.3.3.4.3.1 Scalar LMM :

The following is the matrix form of the scalar form of the contact pressure, see 4.3.3.2.1.1. **LMM**. Once computed the mortar operators, the resulting system for unilateral contact corresponds with (4.32a). In this representation⁷, and in the following ones, the subindex \mathcal{N} represents all the **DOF** but the ones related with the contact problem, \mathcal{M} will represent the master **DOF** and finally \mathcal{S} will identify the slave **DOF**. At the same time, the slave **DOF** can be divided between the active (\mathcal{A}) and inactive (\mathcal{I}) **DOF**. In here the residual for the **LM**, \mathbf{r}_{λ_A} and \mathbf{r}_{λ_I} , corresponds in an algebraic way with (4.32b). We call it algebraic way to the expression of the residual in considering the mortar operators (**M**, **D**). The corresponding **LHS** is defined by the derivatives of the mortar operators from 4.3.3.4.2.2. [Derivatives](#).

$$(4.32a) \quad \begin{bmatrix} \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{N}\mathcal{M}} & \mathbf{K}_{\mathcal{N}\mathcal{S}_A} & \mathbf{K}_{\mathcal{N}\mathcal{S}_I} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{M}\mathcal{M}} & \mathbf{K}_{\mathcal{M}\mathcal{S}_A} & \mathbf{K}_{\mathcal{M}\mathcal{S}_I} & -(\mathbf{n} \cdot \mathbf{M}_A)^T & -(\mathbf{n} \cdot \mathbf{M}_I)^T \\ \mathbf{K}_{\mathcal{S}_A\mathcal{N}} & \mathbf{K}_{\mathcal{S}_A\mathcal{M}} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_I} & (\mathbf{n} \cdot \mathbf{D}_{AA})^T & (\mathbf{n} \cdot \mathbf{D}_{AI})^T \\ \mathbf{K}_{\mathcal{S}_I\mathcal{N}} & \mathbf{K}_{\mathcal{S}_I\mathcal{M}} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_I} & (\mathbf{n} \cdot \mathbf{D}_{IA})^T & (\mathbf{n} \cdot \mathbf{D}_{II})^T \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{\mathcal{M}}} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{\mathcal{S}_A}} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{\mathcal{S}_I}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{\mathcal{N}} \\ \Delta \mathbf{u}_{\mathcal{M}} \\ \Delta \mathbf{u}_{\mathcal{S}_A} \\ \Delta \mathbf{u}_{\mathcal{S}_I} \\ \Delta \lambda_A \\ \Delta \lambda_I \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathcal{N}} \\ \mathbf{r}_{\mathcal{M}} \\ \mathbf{r}_{\mathcal{S}_A} \\ \mathbf{r}_{\mathcal{S}_I} \\ \mathbf{r}_{\lambda_A} \\ \mathbf{r}_{\lambda_I} \end{bmatrix}$$

$$(4.32b) \quad \begin{cases} \mathbf{r}_{\lambda_A} = -\mathbf{n} \cdot (\mathbf{D}\mathbf{x}_1 - \mathbf{M}\mathbf{x}_2) \\ \mathbf{r}_{\lambda_I} = \lambda_n \end{cases}$$

4.3.3.4.3.2 Components LMM :

Here, the algebraic system upcoming from components representation of the **LMM** in the frictionless contact formulation is presented in (4.33a). The *Lagrangian* has been previously shown in 4.3.3.2.2.1. **LMM**. The main difference is the full consideration of the mortar operators **M** and **D**.

$$(4.33a) \quad \begin{bmatrix} \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{N}\mathcal{M}} & \mathbf{K}_{\mathcal{N}\mathcal{S}_A} & \mathbf{K}_{\mathcal{N}\mathcal{S}_I} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{M}\mathcal{M}} & \mathbf{K}_{\mathcal{M}\mathcal{S}_A} & \mathbf{K}_{\mathcal{M}\mathcal{S}_I} & -\mathbf{M}_A^T & -\mathbf{M}_I^T \\ \mathbf{K}_{\mathcal{S}_A\mathcal{N}} & \mathbf{K}_{\mathcal{S}_A\mathcal{M}} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_I} & \mathbf{D}_{AA}^T & \mathbf{D}_{AI}^T \\ \mathbf{K}_{\mathcal{S}_I\mathcal{N}} & \mathbf{K}_{\mathcal{S}_I\mathcal{M}} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_I} & \mathbf{D}_{IA}^T & \mathbf{D}_{II}^T \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{\mathcal{M}}} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{\mathcal{S}_A}} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{\mathcal{S}_I}} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \lambda_A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{\mathcal{N}} \\ \Delta \mathbf{u}_{\mathcal{M}} \\ \Delta \mathbf{u}_{\mathcal{S}_A} \\ \Delta \mathbf{u}_{\mathcal{S}_I} \\ \Delta \lambda_A \\ \Delta \lambda_I \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathcal{N}} \\ \mathbf{r}_{\mathcal{M}} \\ \mathbf{r}_{\mathcal{S}_A} \\ \mathbf{r}_{\mathcal{S}_I} \\ \mathbf{r}_{\lambda_A} \\ \mathbf{r}_{\lambda_I} \end{bmatrix}$$

In the same way that in the previous definition for the scalar **LM**, but considering the penalisation of the tangent components, we define the **LM** residuals \mathbf{r}_{λ_A} and \mathbf{r}_{λ_I} in the Equation (4.33b). In here, and in the following, we define

⁷We will consider a very close notation with the one presented in *Popo*[PhDPop12].

the vectorial **LM** from 4.3.3.2.2.2.ALM. The **LHS** can be seen in Equation (4.36a), and the **LM RHS** in (4.36b).

$$(4.36a) \quad \begin{bmatrix} \mathbf{K}_{NN} & \mathbf{K}_{NM} & \mathbf{K}_{NS_A} & \mathbf{K}_{NS_I} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{NN} & \mathbf{K}_{MM} & \mathbf{K}_{MS_A} & \mathbf{K}_{MS_I} & -k\mathbf{M}_{A}^T & -k\mathbf{M}_{I}^T \\ \mathbf{K}_{S_A N} & \mathbf{K}_{S_A M} & \mathbf{K}_{S_A S_A} & \mathbf{K}_{S_A S_I} & k\mathbf{D}_{AA}^T & k\mathbf{D}_{AI}^T \\ \mathbf{K}_{S_I N} & \mathbf{K}_{S_I M} & \mathbf{K}_{S_I S_A} & \mathbf{K}_{S_I S_I} & k\mathbf{D}_{IA}^T & k\mathbf{D}_{II}^T \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_M} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{S_A}} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \mathbf{u}_{S_I}} & \frac{\partial \mathbf{r}_{\lambda_A}}{\partial \lambda_A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{k^2}{\varepsilon} \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_N \\ \Delta \mathbf{u}_M \\ \Delta \mathbf{u}_{S_A} \\ \Delta \mathbf{u}_{S_I} \\ \Delta \lambda_A \\ \Delta \lambda_I \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_N \\ \mathbf{r}_M \\ \mathbf{r}_{S_A} \\ \mathbf{r}_{S_I} \\ \mathbf{r}_{\lambda_A} \\ \mathbf{r}_{\lambda_I} \end{bmatrix}$$

$$(4.36b) \quad \begin{cases} \mathbf{r}_{\lambda_A} = k\mathbf{n} \cdot (-\mathbf{n} \cdot (\mathbf{D}\mathbf{x}_1 - \mathbf{M}\mathbf{x}_2)) - \frac{k^2}{\varepsilon} \boldsymbol{\tau} \cdot \boldsymbol{\lambda} \\ \mathbf{r}_{\lambda_I} = \frac{k^2}{\varepsilon} \boldsymbol{\lambda} \end{cases}$$

4.3.3.4.4 Static condensation of the system in considering of the DLMM :

As mentioned before, it is possible to statically condensate the system of equations, solving a system in pure displacements, instead of a system with displacements and **LM**. We already stated that this is possible to the diagonal property of the **D** when considering a **DLMM** formulation. The following presents the construction considered for this system of equations that can be applied in any **LHS** where the **LM** considered is decomposed in *Cartesian* components, both frictionless and frictional formulation. We describe the global **LHS** as shown in Equation (4.37), where the matrix is decomposed in 36×36 .

$$(4.37) \quad \begin{bmatrix} \mathbf{K}_{NN} & \mathbf{K}_{NM} & \mathbf{K}_{NS_A} & \mathbf{K}_{NS_I} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{NN} & \mathbf{K}_{MM} & \mathbf{K}_{MS_A} & \mathbf{K}_{MS_I} & \mathbf{K}_{MLM_A} & \mathbf{K}_{MLM_I} \\ \mathbf{K}_{S_A N} & \mathbf{K}_{S_A M} & \mathbf{K}_{S_A S_A} & \mathbf{K}_{S_A S_I} & \mathbf{K}_{S_A LM_A} & \mathbf{K}_{S_A LM_I} \\ \mathbf{K}_{S_I N} & \mathbf{K}_{S_I M} & \mathbf{K}_{S_I S_A} & \mathbf{K}_{S_I S_I} & \mathbf{K}_{S_I LM_A} & \mathbf{K}_{S_I LM_I} \\ \mathbf{0} & \mathbf{K}_{LM_A M} & \mathbf{K}_{LM_A S_A} & \mathbf{K}_{LM_A S_I} & \mathbf{K}_{LM_A LM_A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{LM_I LM_I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_N \\ \Delta \mathbf{u}_M \\ \Delta \mathbf{u}_{S_A} \\ \Delta \mathbf{u}_{S_I} \\ \Delta \lambda_A \\ \Delta \lambda_I \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_N \\ \mathbf{r}_M \\ \mathbf{r}_{S_A} \\ \mathbf{r}_{S_I} \\ \mathbf{r}_{\lambda_A} \\ \mathbf{r}_{\lambda_I} \end{bmatrix}$$

The last system presented can be simplified if we consider that $\mathbf{K}_{S_I LM_A} = \mathbf{0}$, $\mathbf{K}_{S_A LM_I} = \mathbf{0}$ and $\Delta \lambda_I = \mathbf{0}$. Additionally the $\mathbf{K}_{LM_I LM_I}$ is diagonal, so its resolution is trivial, or can be applied by simply imposing $\lambda_I = \mathbf{0}$. With this we statically condensate (4.37) into (4.38). The resulting system is in pure displacement, instead of mixed.

$$(4.38) \quad \begin{bmatrix} \mathbf{K}_{NN} & \mathbf{K}_{NM} & \mathbf{K}_{NS_A} & \mathbf{K}_{NS_I} \\ \mathbf{K}_{NN} + \mathbf{PK}_{S_A N} & \mathbf{K}_{MM} + \mathbf{PK}_{S_A M} & \mathbf{K}_{MS_A} + \mathbf{PK}_{S_A S_A} & \mathbf{K}_{MS_I} + \mathbf{PK}_{S_A S_I} \\ \mathbf{K}_{S_I N} & \mathbf{K}_{S_I M} & \mathbf{K}_{S_I S_A} & \mathbf{K}_{S_I S_I} \\ \mathbf{CK}_{S_A N} & \mathbf{K}_{LM_A M} + \mathbf{CK}_{S_A M} & \mathbf{K}_{LM_A S_A} + \mathbf{CK}_{S_A S_A} & \mathbf{K}_{LM_A S_I} + \mathbf{CK}_{S_A S_I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_N \\ \Delta \mathbf{u}_M \\ \Delta \mathbf{u}_{S_A} \\ \Delta \mathbf{u}_{S_I} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_N \\ \mathbf{r}_M + \mathbf{Pr}_{S_A} \\ \mathbf{r}_{LM_A} + \mathbf{Cr}_{S_A} \\ \mathbf{r}_{S_I} \end{bmatrix}$$

In this last equation, operators **P** and **C** can be computed as follows in (4.39).

$$(4.39a) \quad \mathbf{P} = (\mathbf{K}_{S_A LM_A}^{-1} \mathbf{K}_{M_A LM_A})^T$$

$$(4.39b) \quad \mathbf{C} = (\mathbf{K}_{S_A LM_A}^{-1} \mathbf{K}_{LM_A LM_A})^T$$

Finally, we can extract and deduce the expression necessary in order to compute the active **LM** in a standalone manner as (4.40). This will be evaluated after computing the displacement **DOF**, from which it depends. It is important to note that $\mathbf{K}_{S_A \mathcal{L} \mathcal{M}_A}$ is a diagonal matrix, as is the result of the global assemble of the mortar operators **D**. Equation (4.40a) can be reformulated as Equation (4.40b).

$$(4.40a) \quad \mathbf{K}_{S_A \mathcal{N}} \cdot \mathbf{r}_{\mathcal{N}} + \mathbf{K}_{S_A \mathcal{M}} \cdot \Delta \mathbf{u}_{\mathcal{M}} + \mathbf{K}_{S_A S_A} \cdot \Delta \mathbf{u}_{S_A} + \mathbf{K}_{S_A S_I} \cdot \Delta \mathbf{u}_{S_I} + \mathbf{K}_{S_A \mathcal{L} \mathcal{M}_A} \cdot \Delta \lambda_A = \mathbf{r}_{S_A}$$

$$(4.40b) \quad \Delta \lambda_A = \mathbf{K}_{S_A \mathcal{L} \mathcal{M}_A}^{-1} (\mathbf{r}_{S_A} - \mathbf{K}_{S_A \mathcal{N}} \cdot \mathbf{r}_{\mathcal{N}} - \mathbf{K}_{S_A \mathcal{M}} \cdot \Delta \mathbf{u}_{\mathcal{M}} - \mathbf{K}_{S_A S_I} \cdot \Delta \mathbf{u}_{S_I} - \mathbf{K}_{S_A S_A} \cdot \Delta \mathbf{u}_{S_A})$$

4.3.3.5 Work-flow. Solution algorithm

The following section introduces the algorithm to be considered in order to solve the frictionless contact problem. It is relevant to highlight the fact that this algorithm is related with the semi-smooth strategy of the next section, 4.3.3.6. Active set strategy (Semi-smooth Newton Raphson), and the previous one, 4.3.3.4.3. Algebraic form of the problem. The algorithm is presented in Algorithm 2, here we will explain the differences between each one of the approaches presented (standard **LM**, penalty, etc.), the approaches are similar but with slight differences that are relevant to highlight.

The most significant difference across formulations is the *threshold* to be considered in the active set computation. For **LM** the *threshold* for each node to be checked is λ_{nn+1}^{i+1} and for the **ALM** is $k\lambda_{nn+1}^{i+1} - \epsilon_n \tilde{g}_n$. λ_{nn+1} may correspond with the **LM** for scalar **LM** or for the normal components ($\lambda_n = \mathbf{n} \cdot \boldsymbol{\lambda}$) in case of vector **LM**. In case of penalty formulation the *threshold* would be $-\epsilon_n \tilde{g}_n$. Additionally, it is important to separate the residual corresponding to the displacement solution and the **LM** solution, in case that the problem considers **LM**.

Algorithm 2 Algorithm for the frictionless contact problem

- 1: **procedure** ALGORITHM FOR THE FRICTIONLESS CONTACT PROBLEM
- 2: $t = 0$ and $i = 0$
- 3: Initialise the solution for $\mathbf{u}^0 = \mathbf{0}$
- 4: In case of solving **LM** solution, Initialise the **LM** solution $\boldsymbol{\lambda}^0 = \mathbf{0}$ or $\lambda_n^0 = \mathbf{0}$
- 5: Initialise the active set \mathcal{A}_1^0 and \mathcal{I}_1^0 such that $\mathcal{A}_1^0 \cup \mathcal{I}_1^0 = \mathcal{S}$ and $\mathcal{A}_1^0 \cap \mathcal{I}_1^0 = \emptyset$
- 6: **while** $t < t_{end}$ **do**
- 7: $t = t + \Delta t$ and $i = i + 1$
- 8: Initialise the increment of solution for $\Delta \mathbf{u}_1^i = \mathbf{0}$
- 9: In case of solving **LM** solution, Initialise the **LM** increment of solution $\Delta \boldsymbol{\lambda}_1^i = \mathbf{0}$ or $\Delta \lambda_{n1}^i = \mathbf{0}$
- 10: Search for potential contact pairs, and if required update the pairs and the active set, respecting step 5
- 11: We define the problem as not converged $conv = false$
- 12: **while** $conv = false$ **do**
- 13: Find the solution corresponding to the system presented in 4.3.3.4.3. Algebraic form of the problem
- 14: Update the solution, so $\mathbf{u}_{n+1}^i = \mathbf{u}_n^i + \Delta \mathbf{u}_{n+1}^i$
- 15: In case of **LM** solution $\lambda_{n+1}^i = \lambda_n^i + \Delta \lambda_{n+1}^i$ or $\lambda_{nn+1}^i = \lambda_{nn}^i + \Delta \lambda_{nn+1}^i$
- 16: Update the active set as in (4.41). The threshold for each one of the cases is represented in (4.42).

$$(4.41) \quad \begin{aligned} \mathcal{I}_{n+1}^{i+1} &:= \{j \in \mathcal{S} | threshold_{n+1}^{i+1} \geq 0\} \\ \mathcal{A}_{n+1}^{i+1} &:= \{j \in \mathcal{S} | threshold_{n+1}^{i+1} < 0\} \end{aligned}$$

$$(4.42) \quad threshold_{LM} = \lambda_n \text{ or } \mathbf{n} \cdot \boldsymbol{\lambda}, \quad threshold_{Penalty} = \epsilon \tilde{g}_n, \quad threshold_{ALM} = k\lambda_n + \epsilon \tilde{g}_n \text{ or } \mathbf{k}\mathbf{n} \cdot \boldsymbol{\lambda} + \epsilon \tilde{g}_n$$

- 17: Compute the corresponding residual to check (4.43).

$$(4.43) \quad \|\mathbf{r}_u\| < tolerance_u, \quad \|\mathbf{r}_\lambda\| < tolerance_\lambda$$

- 18: The solution is converged if $\mathcal{A}_{n+1}^{i+1} = \mathcal{A}_{n+1}^i$, $\mathcal{I}_{n+1}^{i+1} = \mathcal{I}_{n+1}^i$ and the residuals from (4.43) are converged.
-

The former algorithm, particularly the check of convergence can be appreciated in Figure 4.18, where the current contact convergence check present in **Kratos** is shown. In here the different types of residuals are check, as well as

the active set check.

CONVERGENCE CHECK										STEP: 1	TIME: 5.0000e-02	DELTA TIME: 5.0000e-02
ITER	DP RATIO	EXP. RAT	ABS	EXP. ABS	LM RATIO	EXP. RAT	ABS	EXP. ABS	CONVERGENCE	ACTIVE SET CONV		
1	1.000E+00	1.000E-04	6.838E+12	1.000E-09	1.000E+00	1.000E-04	9.167E-01	1.000E-09	Not achieved	Not achieved		
2	5.242E-02	1.000E-04	3.585E+11	1.000E-09	3.873E-22	1.000E-04	3.560E-22	1.000E-09	Not achieved	Not achieved		
3	3.830E-04	1.000E-04	2.619E+09	1.000E-09	3.700E+07	1.000E-04	3.399E+07	1.000E-09	Not achieved	Not achieved		
4	1.672E-07	1.000E-04	1.143E+06	1.000E-09	2.137E+06	1.000E-04	1.959E+06	1.000E-09	Not achieved	Achieved		
5	2.736E-12	1.000E-04	1.871E+01	1.000E-09	8.818E+01	1.000E-04	8.883E+01	1.000E-09	Not achieved	Achieved		
6	7.992E-18	1.000E-04	5.465E-05	1.000E-09	8.964E-05	1.000E-04	8.217E-05	1.000E-09	Achieved	Achieved		

Figure 4.18: Example of convergence check in the frictionless contact

4.3.3.6 Active set strategy (Semi-smooth Newton Raphson)

As previously stated, the fully discretised *unilateral* contact problem introduces one significant complexity in the problem resolution. This issue is the contact specific inequality constraints, which introduces two different sets of discrete active and inactive constraints, which are unknown *a priori*. This previous dilemma does not appear for the mesh tying case, see [A.3.Mesh tying](#), where the subsets are known *a priori*. From a mathematical point of view, this introduces an additional source of non-linearity apart from the already existing geometric and material non-linearity. This issue can be solved with the consideration of an appropriate strategy, a common approach is the **Primal-Dual Active Set Strategies (PDASS)**, which is already presented in detail in the optimisation appendix ([D.Constrained optimisation problems](#)).

The principle behind any active set strategy for *unilateral* contact is to iterate looking for the correct subset of master-slave nodes in contact until there is not variation in the respective subsets in the given time step. In the other hand, the contact non-linearity cannot be simply solved in application of the **PDASS**. This is because finding the correct active set \mathcal{A} cannot be resolved by a standard **NR** type approach.

Can be affirmed that in each one of the non-converged subsets we can apply an standard **NR** type algorithm, in the same manner it is applied to the other types of non-linearity. These non-converged subsets are obtained from rearranging the **KKT** conditions. In [Popp\[PhDPop12\]](#) the discrete **KKT** conditions are reformulated within a so-called **NCP**, this **NCP** is equivalent to the *Lagrangian* (\mathcal{L}) contribution for the **LM** in the **ALM**, therefore in case of consider an **ALM** formulation it is *de facto* considered in the *Lagrangian* (\mathcal{L}) definition. Equation (4.44) shows the expression which defines the frictionless **NCP**. This corresponds with the augmented normal contact pressure $\tilde{\lambda}_n$ presented before (4.13), and the criteria will consist on activate/inactivate the corresponding node if the augmented contact pressure $\tilde{\lambda}_n$ is in compression or traction correspondingly. The graphical representation can be seen in Figure 4.19, where the equivalence with the **KKT** conditions is indicated in red colour, and this is drifted accordingly to the penalty ε contribution.

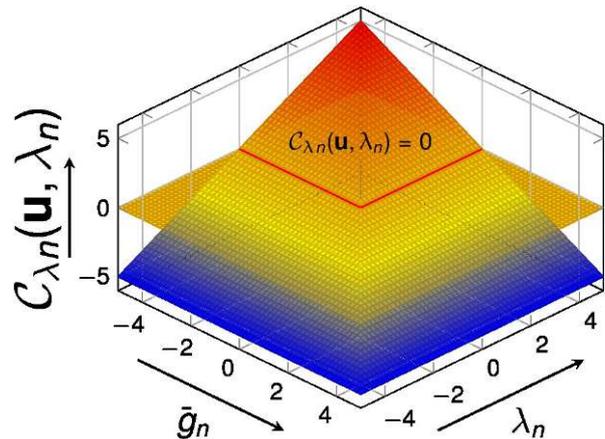


Figure 4.19: Nodal **NCP** function, or *Lagrangian* (\mathcal{L}) contribution for the **LM** in the **ALM**

$$(4.44) \quad C_{\lambda_n} = k\lambda_n - \max(0, k\lambda_n + \varepsilon\tilde{g}_n)$$

Hence, the **PDASS** considered accommodates derivative information on the subsets, allowing the resolution in considering **NR** algorithm also for the contact non-linearity. With the previous consideration, the contact problem can be solved together with the other non-linearity, therefore we can consider one single iterative scheme. For the penalty formulation, just in considering the algorithm presented in [4.3.3.5.Work-flow. Solution algorithm](#) the **PDASS** is taken into account with a standard **NR** strategy.

4.3.4 Frictional contact

In the same manner to the frictionless case before introduce the weak formulation we need to formulate the problem, we require the definition of the strong formulation. The **ALM** formulation presented is a combination of the following references [ArtCC15; PhDGit12; PhDDoc; ArtDPS; ArtYLM].

4.3.4.1 Strong formulation

The part relative to the solutions spaces is exactly the same than the one presented for the frictionless case (4.4) as well as the balance of the linear momentum (4.2). Particularly the developments and definitions presented for the frictionless contact for the **LM** by components, see 4.3.3.2.2. *Vector Lagrange multiplier*, will be applied here. The **KKT** (4.3.3.1) condition is still into play for the frictional problem.

4.3.4.1.1 Tangential contact condition - *Coulomb's law* :

Friction is a complex physical phenomenon. The science that studies the friction, the tribology (see **Tribology**), has concluded the many origins of the physics of the frictional phenomenons. This combines the interactions of elastic and plastic deformations at the contact interface, interaction with wear particles, microfractures, excitation of electrons, etcetera. There exist different friction models as we have already introduced in the state-of-the-art of this chapter, see 4.2.5. *Frictional models*.

In continuum mechanics, the most common description is the phenomenological, or macroscopic, law of *Coulomb* (Figure 4.21), which with *Tresca*⁸ are the most extended and yet simple models of friction. In our models we can identity all these models with \mathcal{F} . In general we will take into consideration the *Coulomb's law*, as it is the most extended and commonly used.

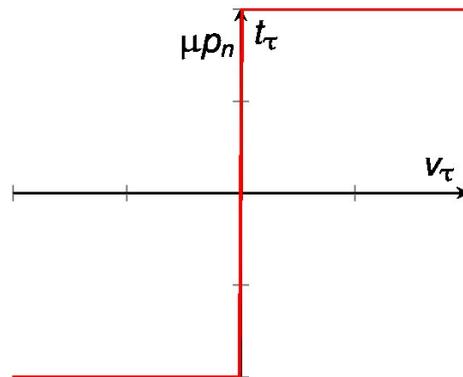


Figure 4.20: *Coulomb's* schematic depiction of frictional contact conditions in tangential direction

The *Coulomb's law* can be defined as (4.45).

$$(4.45a) \quad \phi_{co} := \|\mathbf{t}_{co}^T\| - \mu \|p_n\| \leq 0$$

$$(4.45b) \quad \mathbf{v}_{\tau,rel}(\mathbf{X}^1, t) + \beta \mathbf{t}_{co}^T = \mathbf{0}$$

$$(4.45c) \quad \beta \geq 0$$

$$(4.45d) \quad \phi_{co} \beta = 0$$

Being μ the friction coefficient, and β the velocity-traction ratio. Equation (4.45a) requires the magnitude of the tangential stress vector to not exceed the product of the coefficient of friction and the normal contact pressure. When the tangential stress is less than the *Coulomb* limit ($\phi_{co} < 0$), the continuity equation (4.45d) forces the β to be zero

⁸Which is simpler because does not depend on contact normal pressure, it is just a threshold value.

and, accordingly, the tangential relative velocity to be zero, this is called stick state. When the tangential stress is at the *Coulomb* limit ($\phi_{co} = 0$), β may be greater than zero in (4.45d) and therefore the tangential stress is forced to oppose the relative tangential velocity in (4.45b), this is called the slip state.

The Figure 4.21 presents the graphical representation of *Coulombs* frictional conditions for 3D contact problems. The sub-Figure 4.21a introduces relation between the norm of the tangential velocity and the components of the tangential stress vector. In here the admissible points are situated either in the blue circle ($\beta = 0, \|\mathbf{t}_{co}^\tau\| \leq \mu \|\rho_n\|$), which corresponds with the stick state; or on the surface of the semi-infinite cylinder ($\beta \geq 0, \|\mathbf{t}_{co}^\tau\| \leq \mu \|\rho_n\|$) marked with red colour, associated with the slip state. On the other hand, the sub-Figure 4.21b shows relation between the contact pressure and the components of the contact tangential stress vector. We can see in blue the interior of the *Coulombs* cone ($\|\mathbf{t}_{co}^\tau\| \leq \mu \|\rho_n\|$), which represents the stick state. Contrarily the surface of the cone in red ($\|\mathbf{t}_{co}^\tau\| = \mu \|\rho_n\|$) exemplify the slip state.

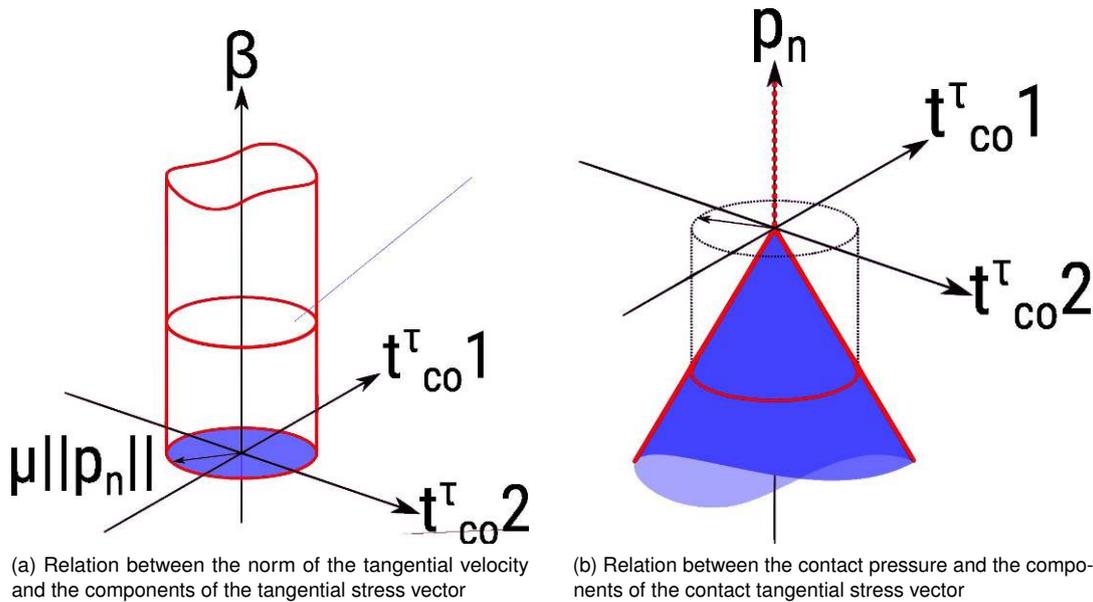


Figure 4.21: Graphical representation of *Coulombs* frictional conditions for 3D contact problem. Inspired [PhDYas11]

Additionally, the Table 4.3 shows the analogy mentioned in 4.2.5. *Frictional models* between the friction and the plasticity phenomenons. The relationship here corresponds with the one presented by *Yastrebov*[PhDYas11], a more detailed analysis can be found in *Antoni*[ArtAnt17].

Friction	Plasticity
Stick state	Elastic deformation
Slip state	Plastic flow
<i>Coulombs's</i> cone $\partial C(\rho_n)$	Yield surface
Maximal frictional stress $\ \mathbf{t}_{co}^\tau\ = \mu \ \rho_n\ $	Yield strength

Table 4.3: Analogies between friction and plasticity

Finally, the definition of the tangent direction is needed. We use the complementary direction to the normal, so calling the normal as \mathbf{n} we can define the tangent as (4.46a). With this we can define for example, the tangent *Lagrange* multiplier as (4.46b).

$$(4.46a) \quad \boldsymbol{\tau} = \mathbf{I} - \mathbf{n} \otimes \mathbf{n}$$

$$(4.46b) \quad \lambda_\tau = \lambda - \mathbf{n} \lambda_n$$

4.3.4.2 Weak formulation

4.3.4.2.1 LMM :

The weak formulation of these equations can be obtained again by multiplying (4.2) with the test function $\mathbf{w}^i \in \mathcal{V}^i$ and integrating in each domain i as (4.47).

$$(4.47) \quad \mathcal{L}^i(\mathbf{u}^i) = \int_{\Omega^i} [\nabla \cdot \boldsymbol{\sigma}^i + \mathbf{b}^i] \cdot \mathbf{w}^i d\Omega^i + \int_{\Gamma_\sigma^i} [\mathbf{t}^i - \boldsymbol{\sigma}^i \cdot \mathbf{n}^i] \cdot \mathbf{w}^i d\Gamma_\sigma^i + \int_{\Gamma_{co}^i} [\mathbf{t}_{co}^i - \boldsymbol{\sigma}^i \cdot \mathbf{n}^i] \cdot \mathbf{w}^i d\Gamma_{co}^i = 0$$

From (4.47) the part corresponding to the frictional contact would be shown in (4.48).

$$(4.48) \quad \mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = \int_{\Gamma_{co}^1} [\mathbf{t}_{co}^1 - \boldsymbol{\sigma}^1 \cdot \mathbf{n}^1] d\Gamma_{co}^1$$

If we consider the test function \mathbf{w}^i as virtual displacements $\delta \mathbf{u}$, and with the application of the *Gauss* divergence theorem it is possible to determine the equivalent expression for the virtual work (4.49).

$$(4.49) \quad \delta \mathcal{L}^i(\mathbf{u}^i, \delta \mathbf{u}^i) = \int_{\Omega^i} \boldsymbol{\sigma}^i : \frac{\partial \delta \mathbf{u}^i}{\partial x_j} d\Omega^i - \int_{\Omega^i} \mathbf{b}^i \cdot \delta \mathbf{u}^i d\Omega^i - \int_{\Sigma_\sigma^i} \mathbf{t}^i \cdot \delta \mathbf{u}^i d\Gamma_\sigma^i - \int_{\Gamma_{co}^i} \mathbf{t}_{co}^i \cdot \delta \mathbf{u}^i d\Gamma_{co}^i = 0 \forall \delta \mathbf{u}^i \in \mathcal{V}^i$$

Equation (4.49) can be regrouped on contributions as in (4.7), obtaining (4.52).

$$(4.50a) \quad -\delta \mathcal{L}_{kin}(\mathbf{u}) = \sum_{i=1}^2 \left[\int_{\Omega^i} \rho^i \ddot{\mathbf{u}}^i \cdot \delta \mathbf{u}^i d\Omega^i \right]$$

$$(4.50b) \quad -\delta \mathcal{L}_{int,ext}(\mathbf{u}) = \sum_{i=1}^2 \left[\int_{\Omega^i} \left(\boldsymbol{\sigma}^i : \frac{\partial \delta \mathbf{u}^i}{\partial x_j} - \mathbf{b} \cdot \delta \mathbf{u}^i \right) d\Omega^i - \int_{\Gamma_\sigma^i} \mathbf{t}^i \cdot \delta \mathbf{u}^i d\Gamma_\sigma^i \right]$$

$$(4.50c) \quad -\delta \mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = \int_{\Gamma_{co}^1} \mathbf{t}_{co}^1 \cdot \delta \mathbf{t}_c^1 d\Gamma_{co}^1$$

Where \mathbf{t}_{co} represents the traction on the interface, the balance between the contact interface implies (4.51)

$$(4.51) \quad \mathbf{t}_{co}^1 d\gamma^1 = -\mathbf{t}_{co}^2 d\gamma^2$$

As in the frictionless case, we will take the slave surface as reference, this allows to rewrite the virtual work integral for the contact as (4.52a).

$$(4.52a) \quad \delta \mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = - \int_{\Gamma_{co}^1} \mathbf{t}_{co}^1 \cdot \delta g d\Gamma_{co}^1$$

Where g represents the general gap (4.52b) (instead of just normal gap).

$$(4.52b) \quad g = (\mathbf{u}^1 - \chi \cdot \mathbf{u}^2)$$

The *Lagrange* multipliers are introduced as additional unknowns on the slave contact surface and are identified as the negative contact traction as (4.52c).

$$(4.52c) \quad \boldsymbol{\lambda} = -\mathbf{t}_{co}^1$$

4.3.4.2.2 Contact constraints :

The Equation (4.52a) is reformulated. Using the *Lagrange* multiplier in (4.52c) we can decompose into normal and tangential components λ_n and λ_τ (4.53).

$$(4.53a) \quad \lambda_n \in \mathbb{R}_0^+ : g(\delta\lambda_n - \lambda_n) \geq 0 \forall \delta\lambda_n \in \mathbb{R}_0^+$$

$$(4.53b) \quad \lambda_\tau \in \mathcal{B}(\mu\lambda_n) : \mathbf{v}_{\tau,rel} \cdot (\delta\lambda_\tau - \lambda_\tau) \leq 0 \forall \delta\lambda_\tau \in \mathcal{B}(\mu\lambda_n)$$

Where $\mathcal{B}(\mu\lambda_n)$ is a $(n1)$ -dimensional sphere with centre 0 and radius $\mu \cdot \lambda_n$ and λ_τ is a *trial* force in the tangential plane. It is the so-called principle of maximal dissipation representing *Coulomb's* law of friction. Then (4.52a) can be rewritten and the weak form of both normal and tangential contact conditions is obtained as (4.54).

$$(4.54a) \quad \lambda \in \mathcal{M}(\lambda) : \int_{\gamma_{co}^1} g(\delta\lambda_n - \lambda_n) d\lambda \geq 0,$$

$$(4.54b) \quad \int_{\gamma_{co}^1} \mathbf{v}_{\tau,rel} (\delta\lambda_\tau - \lambda_\tau) d\lambda \leq 0 \forall \delta\lambda \in \mathcal{M}(\lambda)$$

Where $\mathcal{M}(\lambda)$ is the admissible solution space for the *Lagrange* multiplier and the test space for the trial forces $\delta\lambda$. It is a convex subset $\mathcal{M}(\lambda) \subset \mathcal{M}$ which accommodates the restrictions of λ and $\delta\lambda$ in (4.53). The subspace can be rewritten as (4.55).

$$(4.55) \quad \mathcal{M}(\lambda) := \{ \delta\lambda \in \mathcal{M} : \langle \delta\lambda, \boldsymbol{\eta} \rangle \leq \langle \mu\lambda_n, \|\boldsymbol{\eta}_\tau\| \rangle, \boldsymbol{\eta} \in \mathcal{L}^1 \text{ with } \eta_n \leq 0 \}$$

Here $\langle \cdot, \cdot \rangle$ is the duality pairing of the spaces \mathcal{M} and \mathcal{V}^1 on γ_{co}^1 given by (4.56).

$$(4.56) \quad \langle \delta\lambda, \boldsymbol{\eta} \rangle := \int_{\gamma_{co}^1} \delta\lambda \boldsymbol{\eta} d\gamma$$

4.3.4.2.3 Penalty :

Before define the **ALM** frictional formulation, we introduce the penalty formulation for the frictional problem. The **ALM** formulation will result from the combination of this with the previous **LMM**. The *Lagrangian* (\mathcal{L}) for the problem is defined on (4.57a), separating the normal (4.57b) and tangent contributions (4.57c). In these expressions, $\mathbf{t}_{co}^n = \varepsilon_n \mathbf{g}_n$ defines the normal contact traction and $\mathbf{t}_{co}^\tau = \varepsilon_\tau \mathbf{v}_{\tau,rel}$ corresponds to the tangent contact traction.

$$(4.57a) \quad \mathcal{L}_{co}(\mathbf{u}) = \int_{\Gamma_c^1} l_n + l_\tau d\Gamma_{co}^i$$

$$(4.57b) \quad l_n(\mathbf{g}_n) = \begin{cases} -\frac{\varepsilon_n}{2} \mathbf{g}_n^2, & \mathbf{t}_{co}^n \leq 0, \text{ (Contact zone)} \\ 0, & \mathbf{t}_{co}^n > 0, \text{ (Gap zone)} \end{cases}$$

$$(4.57c) \quad l_\tau(\mathbf{v}_{\tau,rel}) = \begin{cases} \begin{cases} -\frac{\varepsilon_\tau}{2} \mathbf{v}_{\tau,rel} \cdot \mathbf{v}_{\tau,rel}, & \|\mathbf{t}_{co}^\tau\| \leq -\mu \mathbf{t}_{co}^n, \text{ stick} \\ -\frac{\mu}{\varepsilon_\tau} \mathbf{t}_{co}^n \frac{\mathbf{v}_{\tau,rel}}{\|\mathbf{v}_{\tau,rel}\|}, & \|\mathbf{t}_{co}^\tau\| > -\mu \mathbf{t}_{co}^n, \text{ slip} \end{cases}, & \mathbf{t}_{co}^n \leq 0, \text{ (Contact zone)} \\ 0, & \mathbf{t}_{co}^n > 0, \text{ (Gap zone)} \end{cases}$$

With the *Lagrangian* (\mathcal{L}) we can define the derivative in (4.58).

$$(4.58) \quad \delta\mathcal{L}_{co}(\mathbf{u}) = \int_{\Gamma_c^1} \begin{cases} \mathbf{t}_{co}^n \cdot \delta\mathbf{g}_n + \mathbf{t}_{co}^\tau \cdot \delta\mathbf{v}_{\tau,rel} & \text{if } \|\mathbf{t}_{co}^\tau\| \leq -\mu \mathbf{t}_{co}^n \text{ (Contact stick zone)} \\ \mathbf{t}_{co}^n \cdot \delta\mathbf{g}_n - \mu \mathbf{t}_{co}^n \frac{\mathbf{v}_{\tau,rel}}{\|\mathbf{v}_{\tau,rel}\|} \cdot \delta\mathbf{v}_{\tau,rel} & \text{if } \|\mathbf{t}_{co}^\tau\| > -\mu \mathbf{t}_{co}^n \text{ (Contact slip zone)} \\ 0 & \text{if } \mathbf{t}_{co}^n > 0 \text{ (Gap zone)} \end{cases} d\Gamma_{co}^i$$

4.3.4.2.4 ALM :

In the very same way, we have defined an **Augmented Lagrangian** for the frictionless method we can define the same for the frictional case. The **ALM** method to solve contact problems with friction was proposed by *Alart and Curnier*[ArtAC91], additionally to these references we have considered the work of *Cardona*[ArtCC15] and *Yastrebov*[PhDYas11]. The solution obtained for the **ALM** results in a combination of the former **LMM** (4.3.4.2.1.LMM) and penalty (4.3.4.2.3.Penalty) solution.

Focusing in the functional relative to the contact ($\mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = \mathcal{L}_{\nu co} + \mathcal{L}_{\mathcal{M}}$), we can rewrite (4.48) as (4.59).

$$(4.59) \quad \mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = \int_{\Gamma_c^1} I_n + I_\tau d\Gamma_{co}^i$$

Being I_n and I_τ (4.60) the corresponding parts of the *Augmented Lagrangian* formulation for the normal and tangent contributions respectively.

$$(4.60a) \quad I_n(\mathbf{g}_n, \lambda_n) = \begin{cases} \bar{\lambda}_n \mathbf{g}_n - \frac{\varepsilon_n}{2} \mathbf{g}_n^2, & \bar{\lambda}_n \leq 0, \text{ (Contact zone)} \\ -\frac{k^2}{2\varepsilon_n} \lambda_n^2, & \bar{\lambda}_n > 0, \text{ (Gap zone)} \end{cases}$$

With $\bar{\lambda}_n = k\lambda_n + \varepsilon_n \mathbf{g}_n$ being the augmented *Lagrange* multiplier for the normal direction.

$$(4.60b) \quad I_\tau(\mathbf{v}_{\tau,rel}, \boldsymbol{\lambda}_\tau) = \begin{cases} \bar{\boldsymbol{\lambda}}_\tau \cdot \mathbf{v}_{\tau,rel} - \frac{\varepsilon_\tau}{2} \mathbf{v}_{\tau,rel} \cdot \mathbf{v}_{\tau,rel} & , \|\bar{\boldsymbol{\lambda}}_\tau\| \leq -\mu \bar{\lambda}_n, \text{ stick} \\ -\frac{1}{2\varepsilon_\tau} (k^2 \boldsymbol{\lambda}_\tau \cdot \boldsymbol{\lambda}_\tau + 2\mu \bar{\lambda}_n \|\boldsymbol{\lambda}_\tau\| + \mu^2 \bar{\lambda}_n^2) & , \|\bar{\boldsymbol{\lambda}}_\tau\| > -\mu \bar{\lambda}_n, \text{ slip} \\ -\frac{k^2}{2\varepsilon_\tau} \boldsymbol{\lambda}_\tau \cdot \boldsymbol{\lambda}_\tau & , \bar{\lambda}_n > 0, \text{ (Gap zone)} \end{cases}, \bar{\lambda}_n \leq 0, \text{ (Contact zone)}$$

With $\bar{\boldsymbol{\lambda}}_\tau = k\boldsymbol{\lambda}_\tau + \varepsilon_\tau \mathbf{v}_{\tau,rel}$ being the augmented *Lagrange* multiplier for the tangent direction.

Where ε_n and ε_τ are a positive penalty parameter, both for normal and tangent direction, k is a positive scale factor. Using the $\langle \cdot \rangle$ is the *Macaulay* bracket operator, from (4.10), we can express everything as (4.61).

$$(4.61a) \quad I_n(\mathbf{g}_n, \lambda_n) = \frac{1}{\varepsilon_n} (k^2 \lambda_n^2 - \langle \bar{\lambda}_n \rangle^2)$$

$$(4.61b) \quad I_\tau(\mathbf{v}_{\tau,rel}, \boldsymbol{\lambda}_\tau) = \frac{1}{\varepsilon_\tau} (k^2 \boldsymbol{\lambda}_\tau \cdot \boldsymbol{\lambda}_\tau - \|\bar{\boldsymbol{\lambda}}_\tau\|^2 - \langle \|\bar{\boldsymbol{\lambda}}_\tau\| - \mu\bar{\lambda}_n \rangle^2)$$

This functional is \mathcal{C}^1 differentiable *saddle-point*, as shown in Figure 4.22. The solution is obtained as the set of values that render this functional stationary. The *locus* which corresponds to the solution in the normal direction is presented in the optimisation appendix, D.4.3.Applicability on contact problems, in the Figure D.4.

As in the frictionless case, the solution does not depend on the value of parameters ε and k . Nevertheless, the convergence rate does depend on their value. Finally, we can derive (4.59) to obtain the variational form from (4.62).

$$(4.62) \quad \delta \mathcal{L}_{co}(\mathbf{u}, \boldsymbol{\lambda}) = \int_{\Gamma_c^1} \begin{cases} \bar{\lambda}_n \cdot \delta \mathbf{g}_n + k \mathbf{g}_n \delta \lambda_n + \bar{\boldsymbol{\lambda}}_\tau \cdot \delta \mathbf{v}_{\tau,rel} + \mathbf{v}_{\tau,rel} \cdot \delta \bar{\boldsymbol{\lambda}}_\tau & \text{if } \|\bar{\boldsymbol{\lambda}}_\tau\| \leq -\mu \bar{\lambda}_n \text{ (Contact stick zone)} \\ \bar{\lambda}_n \cdot \delta \mathbf{g}_n + k \mathbf{g}_n \delta \lambda_n - \mu \bar{\lambda}_n \frac{\bar{\boldsymbol{\lambda}}_\tau}{\|\bar{\boldsymbol{\lambda}}_\tau\|} \delta \mathbf{v}_{\tau,rel} - \frac{k \boldsymbol{\lambda}_\tau + \mu \bar{\lambda}_n \frac{\bar{\boldsymbol{\lambda}}_\tau}{\|\bar{\boldsymbol{\lambda}}_\tau\|}}{\varepsilon_\tau} \delta \boldsymbol{\lambda}_\tau & \text{if } \|\bar{\boldsymbol{\lambda}}_\tau\| > -\mu \bar{\lambda}_n \text{ (Contact slip zone)} \\ -\frac{k^2}{\varepsilon_n} \lambda_n \delta \lambda_n - \frac{k^2}{\varepsilon_\tau} \boldsymbol{\lambda}_\tau \delta \boldsymbol{\lambda}_\tau & \text{if } \bar{\lambda}_n > 0 \text{ (Gap zone)} \end{cases} d\Gamma_{co}^i$$

The functional from (4.62) makes that the system obtained varies in function if the nodes are present in the contact (slip or stick) or the gap zone, the system is not a priori known like in the frictionless case but with an additional configuration.

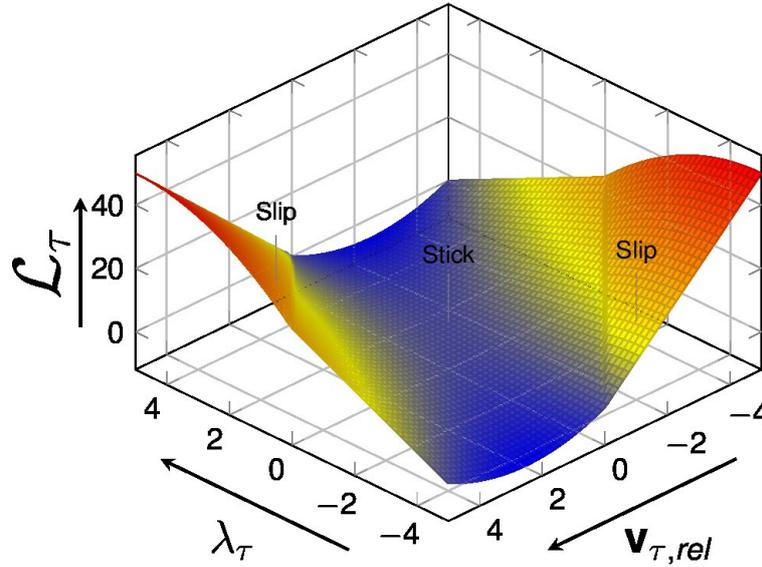


Figure 4.22: Augmented Lagrangian function for the frictional contact problem. Corresponding to (4.61b)

4.3.4.3 Discretisation and numerical integration

Many of the things defined for the frictionless case, like the definition of the dual *Lagrange* multipliers or the definition of the mortar operators remain exactly the same. In this case we will focus in the kinematic definitions for the frictional case using the mortar operators, basically the definition of the discrete contact condition in the tangential direction and the slip in the discrete using the mortar operators.

4.3.4.3.1 Discrete contact condition in tangential direction :

Following the same procedure for the normal direction we can obtain the part equivalent for the tangential direction. The most relevant thing to take into account before any definition is the concept of the relative velocity in the tangential direction $\mathbf{v}_{\tau,rel}$, where we will use for our definition the discrete form of the material velocity field \mathbf{x}^i , which uses the same shape functions for interpolation as the \mathbf{x}^i . We can then define (4.63).

$$(4.63) \quad \int_{\gamma_c^1} \mathbf{v}_{\tau,vel} \cdot (\delta \lambda_{\tau} - \lambda_{\tau}) d\gamma \approx \sum_{j=1}^{n_{slaves}} (\delta \lambda_{\tau} - \lambda_{\tau})^T \tau_j \left[\int_{\gamma_c^1} \Phi_j N_j^1 d\gamma \mathbf{x}_j^1 - \sum_{l=1}^{n_{master}} \int_{\gamma_c^1} \Phi_j (N_l^2 \cdot \xi) d\gamma \mathbf{x}_l^2 \right] \geq 0 \forall \delta \lambda \in \mathcal{M}(\lambda)$$

We can express this equation using the mortar operators, see 4.3.3.4.2, what will give us the following expression (4.64). Where $\tilde{\mathbf{v}}_{\tau j}$ is the weighted relative velocity.

$$(4.64) \quad \int_{\gamma_c^1} \mathbf{v}_{\tau,vel} \cdot (\delta \lambda_{\tau} - \lambda_{\tau}) d\gamma \approx \sum_{j=1}^{n_{slaves}} (\delta \lambda_{\tau} - \lambda_{\tau})^T \tau_j \left[\mathbf{D}_j \mathbf{x}_j^1 - \sum_{l=1}^{n_{master}} \mathbf{M}_l \mathbf{x}_l^2 \right] = \sum_{j=1}^{n_{slaves}} (\delta \lambda_{\tau} - \lambda_{\tau})^T \tilde{\mathbf{v}}_{\tau j} \geq 0$$

4.3.4.3.2 Slip definition :

An important aspect of a proper formulation of frictional laws in the finite sliding context is framed indifference[PhDGit12; ArtYLM] of the rate measures involved. This affects the tangential relative velocity of the contacting bodies ($\mathbf{v}_{\tau,rel}$) in the considered case of frictional contact. This assures that this quantity is unaffected by any rigid

body motion which the two contacting bodies might experience at the instant of the question. Mathematically, this can be tested with formulating the tangential relative velocity $\mathbf{v}_{\tau,rel}$ in an alternative reference frame. Then in the current (mortar projected) instance, we must ensure frame indifference.

Working in the time continuous case first, one may readily show that the tangential component of the mortar projected tangential velocity is not frame indifferent (4.65a). Frame indifference is assessed by viewing the motion from another reference frame, denoted in the following by superscripts $\mathbf{c}(t)$, which can be related to the original spatial frame via (4.65b). Where $\mathbf{c}(t)$ is the relative rigid body translation between the original spatial frame and observer $\mathbf{c}(t)$, while a relative rotation is produced by the proper orthogonal tensor, (4.65c). The frame indifferent relative tangential velocity should satisfy. However, by considering the effect of the transformation (4.65b) on (4.65a), it is readily seen that (4.65d).

$$(4.65a) \quad \tilde{\mathbf{v}}_{\tau}^{nonobj} = \tau_j \left[\mathbf{D}_j \dot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \mathbf{M}_l \dot{\mathbf{x}}_l^2 \right]$$

$$(4.65b) \quad \dot{\mathbf{x}}_j^{(1*)} = \mathbf{c}(t) + \mathbf{Q}(t) \dot{\mathbf{x}}_j^1$$

$$(4.65c) \quad \tilde{\mathbf{v}}_{\tau}^* = \mathbf{Q}(t) \tilde{\mathbf{v}}_{\tau}$$

$$(4.65d) \quad \tilde{\mathbf{v}}_{\tau}^{nonobj*} = \mathbf{Q}(t) \tilde{\mathbf{v}}_{\tau}^{nonobj} - \dot{\mathbf{Q}}(t) \left[\mathbf{D}_j \dot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \mathbf{M}_l \dot{\mathbf{x}}_l^2 \right] \cdot \tau_j$$

Because the term $\left[\mathbf{D}_j \dot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \mathbf{M}_l \dot{\mathbf{x}}_l^2 \right] \neq \mathbf{0}$ in general $\tilde{\mathbf{v}}_{\tau}^{nonobj*}$ does not satisfy the equation (4.65c), and thus some modifications are required to this relative velocity measure to assure material frame indifference. It is possible to restore the objectivity with the inclusion of the rate of a mortar projected distance between the two bodies, denoted as $\dot{\mathbf{g}}$. Then in consequence (4.66) is obtained.

$$(4.66) \quad \tilde{\mathbf{v}}_{\tau} = \tau_j \left[\mathbf{D}_j \dot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \mathbf{M}_l \dot{\mathbf{x}}_l^2 - \dot{\mathbf{g}} \right]$$

We obtain an expression which retains the interpretation of the tangential relative velocity in the case where perfect sliding occurs (*i.e.* when $\dot{\mathbf{g}} = 0$), but which contains the modification necessary to make the velocity measure objectives under all conditions of contact. This is readily seen by using direct calculation to exactly reexpress (4.66) as (4.67b), considering (4.67a).

$$(4.67a) \quad \dot{\mathbf{g}} = \frac{d}{dt} \left[\mathbf{D}_j \dot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \mathbf{M}_l \dot{\mathbf{x}}_l^2 \right] = \left[\dot{\mathbf{D}}_j \dot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \dot{\mathbf{M}}_l \dot{\mathbf{x}}_l^2 \right] + \left[\mathbf{D}_j \ddot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \mathbf{M}_l \ddot{\mathbf{x}}_l^2 \right]$$

$$(4.67b) \quad \tilde{\mathbf{v}}_{\tau} = \tau_j \left[\dot{\mathbf{D}}_j \dot{\mathbf{x}}_j^1 - \sum_{l=1}^{n_{master}} \dot{\mathbf{M}}_l \dot{\mathbf{x}}_l^2 \right]$$

The time derivatives of the mortar operators can be defined using any desired scheme, for example using the *backward Euler* (4.68) scheme as time discretisation.

$$(4.68a) \quad \frac{d(\cdot)}{dt} \approx \frac{(\cdot)^{t+\Delta t} - (\cdot)^t}{\Delta t}$$

$$(4.68b) \quad \frac{d\mathbf{D}}{dt} \approx \frac{\mathbf{D}_j^{t+\Delta t} - \mathbf{D}_j^t}{\Delta t}, \quad \frac{d\mathbf{M}}{dt} \approx \frac{\mathbf{M}_l^{t+\Delta t} - \mathbf{M}_l^t}{\Delta t}$$

With this we can define tangential relative velocity $\tilde{\mathbf{v}}_\tau$ as (4.69a), that multiplies by Δt gives us the nodal slip increment $\tilde{\mathbf{u}}_\tau$ (4.69b).

$$(4.69a) \quad \tilde{\mathbf{v}}_\tau = \tau_j \left[\frac{\mathbf{D}_j^{t+\Delta t} - \mathbf{D}_j^t}{\Delta t} \mathbf{x}_j^1 - \sum_{l=1}^{n_{master}} \frac{\mathbf{M}_l^{t+\Delta t} - \mathbf{M}_l^t}{\Delta t} \dot{\mathbf{x}}_j^2 \right]$$

$$(4.69b) \quad \tilde{\mathbf{u}}_\tau = \tau_j \left[(\mathbf{D}_j^{t+\Delta t} - \mathbf{D}_j^t) \mathbf{x}_j^1 - \sum_{l=1}^{n_{master}} (\mathbf{M}_l^{t+\Delta t} - \mathbf{M}_l^t) \dot{\mathbf{x}}_j^2 \right]$$

4.3.4.3.3 Algebraic form of the problem :

The resulting systems of equations presented in this section will consider always a vectorial **LM** in the cases for the **LMM** and **ALM**, as opposed to the solutions shown in 4.3.3.4.3. Algebraic form of the problem. Because of this reason, we will be able to consider always the static condensation presented in 4.3.3.4.4. Static condensation of the system in considering of the **DLMM** for the **LM** approaches. It is also relevant to spotlight that in the following sections the symbols representing the slip and stick states in addition to the states already presented in 4.3.3.4.3.1. Scalar **LMM**. These states are represented respectively as *sl* and *st* subindexes.

4.3.4.3.3.1 LMM :

Equation (4.70a) presents the system of equations relative to the standard **LMM** for the frictional problem. We can see that, we have an additional block to block presented for the frictionless solution in 4.3.3.4.3.2. Components **LMM**. This is due to the fact that now the active **DOF** relative to the **LM** can be split into two different groups. The group relative to the slip state (*sl*) and the stick state (*st*). The residual of the contact solution is therefore also split into these two subgroups, as see in Equation (4.70b).

$$(4.70a) \quad \begin{bmatrix} \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{N}\mathcal{M}} & \mathbf{K}_{\mathcal{N}\mathcal{S}_A} & \mathbf{K}_{\mathcal{N}\mathcal{S}_I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{M}\mathcal{M}} & \mathbf{K}_{\mathcal{M}\mathcal{S}_A} & \mathbf{K}_{\mathcal{M}\mathcal{S}_I} & -\mathbf{M}_{\mathcal{A}sl}^T & -\mathbf{M}_{\mathcal{A}st}^T & -\mathbf{M}_{\mathcal{I}}^T \\ \mathbf{K}_{\mathcal{S}_A\mathcal{N}} & \mathbf{K}_{\mathcal{S}_A\mathcal{M}} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_I} & \mathbf{D}_{\mathcal{A}sl}^T & \mathbf{D}_{\mathcal{A}st}^T & \mathbf{D}_{\mathcal{AI}}^T \\ \mathbf{K}_{\mathcal{S}_I\mathcal{N}} & \mathbf{K}_{\mathcal{S}_I\mathcal{M}} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_I} & \mathbf{D}_{\mathcal{I}sl}^T & \mathbf{D}_{\mathcal{I}st}^T & \mathbf{D}_{\mathcal{II}}^T \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_{A}sl}}{\partial \mathbf{u}_{\mathcal{M}}} & \frac{\partial \mathbf{r}_{\lambda_{A}sl}}{\partial \mathbf{u}_{\mathcal{S}_A}} & \frac{\partial \mathbf{r}_{\lambda_{A}sl}}{\partial \mathbf{u}_{\mathcal{S}_I}} & \frac{\partial \mathbf{r}_{\lambda_{A}sl}}{\partial \lambda_{A}sl}} & \frac{\partial \mathbf{r}_{\lambda_{A}sl}}{\partial \lambda_{A}st}} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_{A}st}}{\partial \mathbf{u}_{\mathcal{M}}} & \frac{\partial \mathbf{r}_{\lambda_{A}st}}{\partial \mathbf{u}_{\mathcal{S}_A}} & \frac{\partial \mathbf{r}_{\lambda_{A}st}}{\partial \mathbf{u}_{\mathcal{S}_I}} & \frac{\partial \mathbf{r}_{\lambda_{A}st}}{\partial \lambda_{A}sl}} & \frac{\partial \mathbf{r}_{\lambda_{A}st}}{\partial \lambda_{A}st}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{\mathcal{N}} \\ \Delta \mathbf{u}_{\mathcal{M}} \\ \Delta \mathbf{u}_{\mathcal{S}_A} \\ \Delta \mathbf{u}_{\mathcal{S}_I} \\ \Delta \lambda_{A}sl \\ \Delta \lambda_{A}st \\ \Delta \lambda_{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathcal{N}} \\ \mathbf{r}_{\mathcal{M}} \\ \mathbf{r}_{\mathcal{S}_A} \\ \mathbf{r}_{\mathcal{S}_I} \\ \mathbf{r}_{\lambda_{A}sl} \\ \mathbf{r}_{\lambda_{A}st} \\ \mathbf{r}_{\lambda_{\mathcal{I}}} \end{bmatrix}$$

Equation (4.70b) shows the different residuals associated to the **LM**, and presented in Equation (4.70a). The respective derived terms in the **LHS**, as in the frictionless case, require the mortar operators derivatives defined in 4.3.3.4.2.2. Derivatives.

$$(4.70b) \quad \begin{cases} \mathbf{r}_{\lambda_{A}sl} = \mathbf{n} \cdot (-\mathbf{n} \cdot (\mathbf{D}\mathbf{x}_1 - \mathbf{M}\mathbf{x}_2)) - (\boldsymbol{\tau} \cdot \boldsymbol{\lambda} - \mathcal{F}) \\ \mathbf{r}_{\lambda_{A}st} = \mathbf{n} \cdot (-\mathbf{n} \cdot (\mathbf{D}\mathbf{x}_1 - \mathbf{M}\mathbf{x}_2)) + \tilde{\mathbf{u}}_\tau \\ \mathbf{r}_{\lambda_{\mathcal{I}}} = \lambda \end{cases}$$

$$(4.70c) \quad \mathcal{F} = -\mu \lambda_n \boldsymbol{\tau}$$

The \mathcal{F} is the generic representation of the frictional threshold, we have already mentioned this in the section dedicated to the *Coulomb's* frictional law (4.3.4.1.1. Tangential contact condition - *Coulomb's law*). We will consider this frictional law, as our theoretical developments have departed from this hypothesis. In any case, the formulation can be adapted to consider a different frictional criterion changing \mathcal{F} from (4.70b), as well as in the corresponding active set computation (see 4.3.4.4. Work-flow. Solution algorithm). The corresponding expression for \mathcal{F} for the *Coulomb's* frictional law is presented in (4.70c), and it will be considered in the following for the standard **LM**. Additionally, $\tilde{\mathbf{u}}_\tau$, corresponds with the expression from Equation (4.69b).

4.3.4.3.2 Penalty :

For the frictional solution, as we must split solution between the slip and the stick state and we lack of **LM** in our system, we separate the contact **DOF** between slip and stick in order to separate the contact contributions added to the displacement **DOF**. Because of this, the system in Equation (4.71a) does not look as an square **LHS**, 6×3 instead of 6×6 , but it is actually a square system of equations.

$$(4.71a) \quad \begin{bmatrix} \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{N}\mathcal{M}} & \mathbf{K}_{\mathcal{N}\mathcal{S}} \\ \mathbf{K}_{\mathcal{M}\mathcal{N}}^{sl} & \mathbf{K}_{\mathcal{M}\mathcal{M}}^{sl} - \varepsilon_n \left(\mathbf{n} \cdot \mathbf{M}^{slT} + \frac{\partial(\mathbf{n} \cdot \mathbf{M}^{slT})}{\partial \mathbf{u}_{\mathcal{M}}} \mathbf{x}_{\mathcal{M}}^{sl} \right) + \frac{\partial \mathcal{F}_{\mathcal{M}}}{\partial \mathbf{u}_{\mathcal{M}}} & \mathbf{K}_{\mathcal{M}\mathcal{S}}^{sl} - \varepsilon_n \frac{\partial(\mathbf{n} \cdot \mathbf{M}^{slT})}{\partial \mathbf{u}_{\mathcal{S}_A}} \mathbf{x}_{\mathcal{M}}^{sl} + \frac{\partial \mathcal{F}_{\mathcal{M}}}{\partial \mathbf{u}_{\mathcal{S}_A}} \\ \mathbf{K}_{\mathcal{S}_A\mathcal{N}}^{sl} & \mathbf{K}_{\mathcal{S}_A\mathcal{M}}^{sl} + \varepsilon_n \frac{\partial(\mathbf{n} \cdot \mathbf{D}_A^{slT})}{\partial \mathbf{u}_{\mathcal{M}}} \mathbf{x}_{\mathcal{S}_A}^{sl} + \frac{\partial \mathcal{F}_{\mathcal{S}}}{\partial \mathbf{u}_{\mathcal{M}}} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_A}^{sl} + \varepsilon_n \left(\mathbf{n} \cdot \mathbf{D}_A^{slT} + \frac{\partial(\mathbf{n} \cdot \mathbf{D}_A^{slT})}{\partial \mathbf{u}_{\mathcal{S}_A}} \mathbf{x}_{\mathcal{S}_A}^{sl} \right) + \frac{\partial \mathcal{F}_{\mathcal{S}}}{\partial \mathbf{u}_{\mathcal{S}_A}} \\ \mathbf{K}_{\mathcal{M}\mathcal{N}}^{st} & \mathbf{K}_{\mathcal{M}\mathcal{M}}^{st} - \left(\varepsilon_n \mathbf{n} \cdot \mathbf{M}^{stT} + \frac{\partial(\mathbf{n} \cdot \mathbf{M}^{stT})}{\partial \mathbf{u}_{\mathcal{M}}} \mathbf{x}_{\mathcal{M}}^{st} \right) - \varepsilon_t \frac{\partial \ddot{\mathbf{u}}_{\tau}}{\partial \mathbf{u}_{\mathcal{M}}} & \mathbf{K}_{\mathcal{M}\mathcal{S}}^{st} - \varepsilon_n \frac{\partial(\mathbf{n} \cdot \mathbf{M}^{stT})}{\partial \mathbf{u}_{\mathcal{S}_A}} \mathbf{x}_{\mathcal{M}}^{st} - \varepsilon_t \frac{\partial \ddot{\mathbf{u}}_{\tau}}{\partial \mathbf{u}_{\mathcal{S}_A}} \\ \mathbf{K}_{\mathcal{S}_A\mathcal{N}}^{st} & \mathbf{K}_{\mathcal{S}_A\mathcal{M}}^{st} + \varepsilon_n \frac{\partial(\mathbf{n} \cdot \mathbf{D}_A^{stT})}{\partial \mathbf{u}_{\mathcal{M}}} \mathbf{x}_{\mathcal{S}_A}^{st} + \varepsilon_t \frac{\partial \ddot{\mathbf{u}}_{\tau}}{\partial \mathbf{u}_{\mathcal{M}}} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_A}^{st} + \varepsilon_n \left(\mathbf{n} \cdot \mathbf{D}_A^{stT} + \frac{\partial(\mathbf{n} \cdot \mathbf{D}_A^{stT})}{\partial \mathbf{u}_{\mathcal{S}_A}} \mathbf{x}_{\mathcal{S}_A}^{st} \right) + \varepsilon_t \frac{\partial \ddot{\mathbf{u}}_{\tau}}{\partial \mathbf{u}_{\mathcal{S}_A}} \\ \mathbf{K}_{\mathcal{S}_I\mathcal{N}} & \mathbf{K}_{\mathcal{S}_I\mathcal{M}} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_I} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{\mathcal{N}} \\ \Delta \mathbf{u}_{\mathcal{M}}^{sl} \\ \Delta \mathbf{u}_{\mathcal{S}_A}^{sl} \\ \Delta \mathbf{u}_{\mathcal{M}}^{st} \\ \Delta \mathbf{u}_{\mathcal{S}_A}^{st} \\ \Delta \mathbf{u}_{\mathcal{S}_I} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathcal{N}} \\ \mathbf{r}_{\mathcal{M}}^{sl} - \varepsilon_n \mathbf{n} \cdot \mathbf{M}^{sl} \mathbf{x}_{\mathcal{M}}^{sl} + \mathcal{F}_{\mathcal{M}} \\ \mathbf{r}_{\mathcal{S}_A}^{sl} + \varepsilon_n \mathbf{n} \cdot \mathbf{D}_A^{sl} \mathbf{x}_{\mathcal{S}_A}^{sl} + \mathcal{F}_{\mathcal{S}} \\ \mathbf{r}_{\mathcal{M}}^{st} - \varepsilon_n \mathbf{n} \cdot \mathbf{M}^{st} \mathbf{x}_{\mathcal{M}}^{st} - \varepsilon_t \ddot{\mathbf{u}}_{\tau} \\ \mathbf{r}_{\mathcal{S}_A}^{st} + \varepsilon_n \mathbf{n} \cdot \mathbf{D}_A^{st} \mathbf{x}_{\mathcal{S}_A}^{st} + \varepsilon_t \ddot{\mathbf{u}}_{\tau} \\ \mathbf{r}_{\mathcal{S}_I} \end{bmatrix}$$

The frictional threshold (\mathcal{F}) for the penalty formulation, considering the *Coulomb's* frictional law, corresponds with $\mathcal{F} = -\mu \varepsilon_n \bar{g}_n \tau$. It consists in take the penalty approach for the normal contact pressure ($\varepsilon_n \bar{g}_n$), and multiply by the friction coefficient (μ) in the tangent direction (τ). In this case, as we lack the **LM** solution in order to determine the tangent direction τ we must consider the direction of the slip increment ($\ddot{\mathbf{u}}_{\tau}$), defined as $\tau = \frac{\ddot{\mathbf{u}}_{\tau}}{\|\ddot{\mathbf{u}}_{\tau}\|}$.

4.3.4.3.3 ALM :

As we have already pointed in 4.3.4.2.4.ALM we have to consider two different penalties, one in the normal direction (ε_n) and the other one for the tangent direction (ε_{τ}). This increases a little bit the complexity of the expressions in comparison with the **LMM** one. In (4.72a) the resulting system of equations is shown.

$$(4.72a) \quad \begin{bmatrix} \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{N}\mathcal{M}} & \mathbf{K}_{\mathcal{N}\mathcal{S}_A} & \mathbf{K}_{\mathcal{N}\mathcal{S}_I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{\mathcal{M}\mathcal{N}} & \mathbf{K}_{\mathcal{M}\mathcal{M}} & \mathbf{K}_{\mathcal{M}\mathcal{S}_A} & \mathbf{K}_{\mathcal{M}\mathcal{S}_I} & -k\mathbf{M}_{\mathcal{A}sl}^T & -k\mathbf{M}_{\mathcal{A}st}^T & -k\mathbf{M}_{\mathcal{I}}^T \\ \mathbf{K}_{\mathcal{S}_A\mathcal{N}} & \mathbf{K}_{\mathcal{S}_A\mathcal{M}} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_A\mathcal{S}_I} & k\mathbf{D}_{\mathcal{A}sl}^T & k\mathbf{D}_{\mathcal{A}st}^T & k\mathbf{D}_{\mathcal{A}I}^T \\ \mathbf{K}_{\mathcal{S}_I\mathcal{N}} & \mathbf{K}_{\mathcal{S}_I\mathcal{M}} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_A} & \mathbf{K}_{\mathcal{S}_I\mathcal{S}_I} & k\mathbf{D}_{\mathcal{I}sl}^T & k\mathbf{D}_{\mathcal{I}st}^T & k\mathbf{D}_{\mathcal{I}I}^T \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}sl}}}{\partial \mathbf{u}_{\mathcal{M}}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}sl}}}{\partial \mathbf{u}_{\mathcal{S}_A}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}sl}}}{\partial \mathbf{u}_{\mathcal{S}_I}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}sl}}}{\partial \lambda_{\mathcal{A}sl}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}st}}}{\partial \lambda_{\mathcal{A}st}} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}st}}}{\partial \mathbf{u}_{\mathcal{M}}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}st}}}{\partial \mathbf{u}_{\mathcal{S}_A}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}st}}}{\partial \mathbf{u}_{\mathcal{S}_I}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}st}}}{\partial \lambda_{\mathcal{A}st}} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{A}st}}}{\partial \lambda_{\mathcal{A}st}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda_{\mathcal{I}}}}{\lambda_{\mathcal{I}}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{\mathcal{N}} \\ \Delta \mathbf{u}_{\mathcal{M}} \\ \Delta \mathbf{u}_{\mathcal{S}_A} \\ \Delta \mathbf{u}_{\mathcal{S}_I} \\ \Delta \lambda_{\mathcal{A}sl} \\ \Delta \lambda_{\mathcal{A}st} \\ \Delta \lambda_{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathcal{N}} \\ \mathbf{r}_{\mathcal{M}} \\ \mathbf{r}_{\mathcal{S}_A} \\ \mathbf{r}_{\mathcal{S}_I} \\ \mathbf{r}_{\lambda_{\mathcal{A}sl}} \\ \mathbf{r}_{\lambda_{\mathcal{A}st}} \\ \mathbf{r}_{\lambda_{\mathcal{I}}} \end{bmatrix}$$

As in the **LMM** case, we separate the **LM** residual in three parts, for inactive, active slip and active stick **DOF**. The resulting residuals in (4.72b). In the case of the **ALM**, the \mathcal{F} for the *Coulomb's* frictional law is computed considering the augmented **LM** ($\bar{\lambda}_n$) instead of the standard one (λ_n), this can be expressed as $\mathcal{F} = -\mu \bar{\lambda}_n \tau$.

$$(4.72b) \quad \begin{cases} \mathbf{r}_{\lambda_{\mathcal{A}sl}} = k\mathbf{n} \cdot (-\mathbf{n} \cdot (\mathbf{D}\mathbf{x}_1 - \mathbf{M}\mathbf{x}_2)) - \frac{k^2}{\varepsilon_n} (\boldsymbol{\tau} \cdot \boldsymbol{\lambda} - \frac{\mathcal{F}}{k}) \\ \mathbf{r}_{\lambda_{\mathcal{A}st}} = k\mathbf{n} \cdot (-\mathbf{n} \cdot (\mathbf{D}\mathbf{x}_1 - \mathbf{M}\mathbf{x}_2)) + k\ddot{\mathbf{u}}_{\tau} \\ \mathbf{r}_{\lambda_{\mathcal{I}}} = \frac{k^2}{\varepsilon_n} \mathbf{n} \cdot \boldsymbol{\lambda} + \frac{k^2}{\varepsilon_{\tau}} \boldsymbol{\tau} \cdot \boldsymbol{\lambda} \end{cases}$$

4.3.4.4 Work-flow. Solution algorithm

The main changes relative to the solution loop presented in 4.3.3.5. Work-flow. Solution algorithm for the frictionless solution is the need to compute a more complex active set, which includes the slip and stick state, as well as the computation of the corresponding residuals for these states. The last stated, means that the residual corresponding to the **LM** is divided into three different components, the relative to the normal direction, the relative to the tangent direction associated to the slip state, and finally the one obtained from the stick state. It is relevant to separate the residuals from the slip/stick states, as the magnitude orders from these components change greatly and it may be difficult to achieve a convergence if mixing them. We remark the need to do this search at each time step in order to adapt to the evolution of the geometry.

Algorithm 3 Algorithm for the frictional contact problem

- 1: **procedure** ALGORITHM FOR THE FRICTIONAL CONTACT PROBLEM
- 2: $t = 0$ and $i = 0$
- 3: Initialise the solution for $\mathbf{u}^0 = \mathbf{0}$
- 4: In case of solving **LM** solution, Initialise the **LM** solution $\boldsymbol{\lambda}^0 = \mathbf{0}$
- 5: Initialise the active set \mathcal{A}_1^0 and \mathcal{I}_1^0 such that $\mathcal{A}_1^0 \cup \mathcal{I}_1^0 = \mathcal{S}$ and $\mathcal{A}_1^0 \cap \mathcal{I}_1^0 = \emptyset$
- 6: Initialise the slip/stick set \mathcal{A}_{sl}^0 and \mathcal{A}_{st}^0 such that $\mathcal{A}_{sl}^0 \cup \mathcal{A}_{st}^0 = \mathcal{A}_1^0$ and $\mathcal{A}_{sl}^0 \cap \mathcal{A}_{st}^0 = \emptyset$
- 7: **while** $t < t_{end}$ **do**
- 8: $t = t + \Delta t$ and $i = i + 1$
- 9: Initialise the increment of solution for $\Delta \mathbf{u}_i^i = \mathbf{0}$
- 10: In case of solving **LM** solution, Initialise the **LM** increment of solution $\Delta \boldsymbol{\lambda}_1^i = \mathbf{0}$
- 11: Search for potential contact pairs, and if required update the pairs and the active set, respecting step 5
- 12: We define the problem as not converged $conv = false$
- 13: **while** $conv = false$ **do**
- 14: Find the solution corresponding to the system presented in 4.3.4.3.3. Algebraic form of the problem
- 15: Update the solution, so $\mathbf{u}_{n+1}^i = \mathbf{u}_n^i + \Delta \mathbf{u}_{n+1}^i$ and in case of **LM** solution $\boldsymbol{\lambda}_{n+1}^i = \boldsymbol{\lambda}_n^i + \Delta \boldsymbol{\lambda}_{n+1}^i$
- 16: Update the active set as in (4.73). The threshold for each one of the cases is represented in (4.74).

$$(4.73) \quad \begin{aligned} \mathcal{I}_{n+1}^{i+1} &:= \{j \in \mathcal{S} \mid threshold_{n+1}^{i+1} \geq 0\} \\ \mathcal{A}_{n+1}^{i+1} &:= \{j \in \mathcal{S} \mid threshold_{n+1}^{i+1} < 0\} \end{aligned}$$

$$(4.74) \quad threshold_{LM} = \lambda_n \text{ or } \mathbf{n} \cdot \boldsymbol{\lambda}, \quad threshold_{Penalty} = \epsilon \bar{g}_n, \quad threshold_{ALM} = k \lambda_n + \epsilon \bar{g}_n \text{ or } k \mathbf{n} \cdot \boldsymbol{\lambda} + \epsilon \bar{g}_n$$

- 17: Update the slip/stick set as in (4.75). To evaluate this we require the frictional threshold \mathcal{F} , considering *Coulomb's law*, for the cases represented in (4.76). We also require the tangent contact stress (\mathbf{t}_{co}^T) (4.77).

$$(4.75) \quad \begin{aligned} \mathcal{A}_{sln+1}^{i+1} &:= \{j \in \mathcal{A} \mid \mathbf{t}_{co}^T \geq \mathcal{F}_{n+1}^{i+1}\} \\ \mathcal{A}_{stn+1}^{i+1} &:= \{j \in \mathcal{A} \mid \mathbf{t}_{co}^T < \mathcal{F}_{n+1}^{i+1}\} \end{aligned}$$

$$(4.76) \quad \mathcal{F}_{LM} = \mu \mathbf{n} \cdot \boldsymbol{\lambda}, \quad \mathcal{F}_{Penalty} = \mu \epsilon_n \bar{g}_n, \quad \mathcal{F}_{ALM} = \mu (k \mathbf{n} \cdot \boldsymbol{\lambda} + \epsilon_n \bar{g}_n)$$

$$(4.77) \quad \mathbf{t}_{LMco}^T = \|\boldsymbol{\tau} \cdot \boldsymbol{\lambda}\|, \quad \mathbf{t}_{Penaltyco}^T = \|\epsilon_\tau \bar{\mathbf{u}}_\tau\|, \quad \mathbf{t}_{ALMco}^T = \|k \boldsymbol{\tau} \cdot \boldsymbol{\lambda} + \epsilon_\tau \bar{\mathbf{u}}_\tau\|$$

- 18: Compute the corresponding residual to check (4.78).

$$(4.78) \quad \|\mathbf{r}_u\| < tolerance_u, \quad \|\mathbf{r}_{\lambda_n}\| < tolerance_{\lambda_n}, \quad \|\mathbf{r}_{\lambda_\tau^{sl}}\| < tolerance_{\lambda_\tau^{sl}}, \quad \|\mathbf{r}_{\lambda_\tau^{st}}\| < tolerance_{\lambda_\tau^{st}}$$

- 19: Solution converged if $\mathcal{A}_{sln+1}^{i+1} = \mathcal{A}_{sln+1}^i$, $\mathcal{A}_{stn+1}^{i+1} = \mathcal{A}_{stn+1}^i$, $\mathcal{I}_{n+1}^{i+1} = \mathcal{I}_{n+1}^i$ and residuals (4.78) are converged.
-

As well as we did in the frictionless case, we present on Figure 4.23 the output in *Kratos* representing the **NL** convergence check of a time step. We can appreciate the four different residuals which are check, in addition to the

two different active set checked.

CONVERGENCE CHECK														STEP: 1		TIME: 6.0000e-02		DELTA TIME: 6.0000e-02	
ITER	NP	EXP	ABS	EXP	ABS	EXP	ABS	EXP	ABS	EXP	ABS	EXP	ABS	CONVERGENCE	ACTIVE SET	CONV	SLIP/STICK	CONV	
1	1.000E+00	1.000E-04	6.000E-12	1.000E-00	1.000E+00	1.000E-04	1.252E+08	1.000E-00	1.000E-00	1.000E-04	1.666E-03	1.000E-00	0.000E+00	1.000E-04	0.000E+00	1.000E-00	Not achieved	Not achieved	Achieved
2	6.702E-02	1.000E-04	2.000E-11	1.000E-00	1.000E-01	1.000E-04	1.254E+07	1.000E-00	1.177E-17	1.000E-04	1.001E-14	1.000E-00	0.000E+00	1.000E-04	0.000E+00	1.000E-00	Not achieved	Not achieved	Achieved
3	4.171E-06	1.000E-04	2.871E-09	1.000E-00	1.170E+07	1.000E-04	1.416E+07	1.000E-00	2.720E+04	1.000E-04	6.331E+07	1.000E-00	0.000E+00	1.000E-04	0.000E+00	1.000E-00	Not achieved	Not achieved	Achieved
4	5.809E-05	1.000E-04	3.022E-08	1.000E-00	5.100E+05	1.000E-04	6.393E+06	1.000E-00	6.801E-02	1.000E-04	1.100E-06	1.000E-00	1.000E-04	2.012E+06	1.000E-00	1.000E-00	Not achieved	Achieved	Not achieved
5	0.2027E-08	1.000E-04	6.970E-07	1.000E-00	1.4070E+04	1.000E-04	2.4020E+04	1.000E-00	1.2264E-07	1.000E-04	2.0300E+07	1.000E-00	3.2720E-07	1.000E-04	8.2400E+04	1.000E-00	Not achieved	Achieved	Not achieved
6	7.011E-11	1.000E-04	4.794E-02	1.000E-00	4.359E+02	1.000E-04	6.606E+02	1.000E-00	7.031E-75	1.000E-04	1.172E-71	1.000E-00	9.904E-04	1.000E-04	2.232E+03	1.000E-00	Not achieved	Achieved	Achieved
7	7.021E-10	1.000E-04	6.340E-08	1.000E-00	3.029E-02	1.000E-04	4.416E-02	1.000E-00	5.767E-10	1.000E-04	9.611E-102	1.000E-00	7.001E-00	1.000E-04	1.002E-01	1.000E-00	Not achieved	Achieved	Achieved
8	2.3021E-08	1.000E-04	1.007E-07	1.000E-00	2.091E-09	1.000E-04	2.002E-08	1.000E-00	6.470E-105	1.000E-04	1.000E-101	1.000E-00	4.000E-12	1.000E-04	1.215E-05	1.000E-00	Achieved	Achieved	Achieved

Figure 4.23: Example of convergence check in the frictional contact

4.3.4.5 Active set strategy (Semi-smooth Newton Raphson)

The reformulation of frictional contact conditions is similar to the frictionless case. As well as in the former case, the **LM** contribution of the *Lagrangian* (\mathcal{L}) in the **ALM** (4.60) replaces the **NCP** presented in *Popp*[PhDPop12] and *Gitterle*[PhDGit12]. This **NCP** takes the form of a two component vector equation and is written as (4.79), the visual representation can be seen in Figure 4.24. This representation shows a distinctive behaviour for the slip/stick states in the similar manner to the one presented in the Figure 4.22 for the frictional *Lagrangian* (\mathcal{L}) contribution. As in the frictionless case, for the frictional problem the **PDASS** is taken into consideration in considering the workflow presented in 4.3.4.4. *Work-flow. Solution algorithm.*

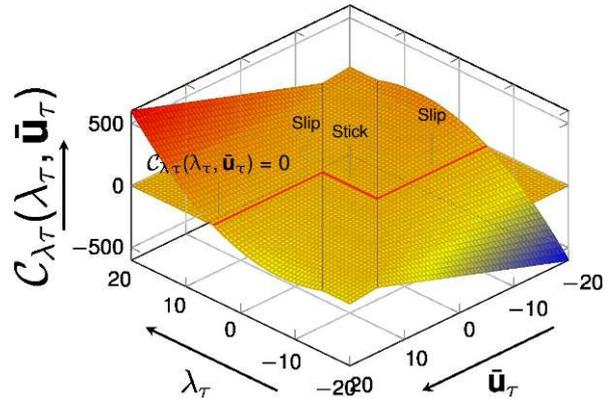


Figure 4.24: **NCP** for the frictional contact problem.

$$(4.79) \quad C_{\tau}(\lambda_{\tau}, \bar{\mathbf{u}}_{\tau}) = \max(\mu \bar{\lambda}_n, \|\bar{\lambda}_{\tau}\|) \lambda_{\tau} - \mu \max(0, \lambda_n) \bar{\lambda}_{\tau}$$

4.4 Contact detection. Search techniques

4.4.1 Introduction

Detection phase it is quite different depending on if we are considering an implicit or an explicit approach[PhDYas11]:

- **Explicit approach:** Here, as the time step is usually very small, we need to detect the solids that actually have penetrated each other, and apply the corresponding forces in order to repulse them. There are different ways to apply these repulsive forces, but the point is that we focus on real penetration, not estimations.
- **Implicit approach:** As the solution of the system depends on the values that have not been completed yet, we need to estimate potential pairings. With this potential pairing, we include the corresponding **DOF**, with the corresponding additional *Lagrange* multiplier, and assemble the **LHS** and **RHS** related to these potential pairings.

As the target of this work is the consideration of an implicit approach, we need to take care of the problematic of the former. The detection phase is crucial, as the additional **DOF** and assembling is not only numerically expensive, but the condition of the system is significantly affected by them.

Additionally, for a problem with n slave elements and m master elements, the cost of a naive brute force searching algorithms (searching all master elements for valid intersections with each slave element) is $\mathcal{O}(nm)$, which is unacceptable for a problem with a large amount of contact surface elements. The ultimate goal of a contact searching algorithm is to find all proximate slave-master element pairs. In conclusion, this phase may be as important as the formulation considered as it influences the whole resolution of the problem.

These methodologies have been particularly developed in recent years, thanks not only for **FE** requirements, but also computer graphics(physically based modelling, animation)[ArtZK12; ArtRKC02], robotics (motion planning, collision avoidance)[ArtTK14], industrial applications (virtual prototyping, assembly tests)[ArtFig10; ArtFF03] and

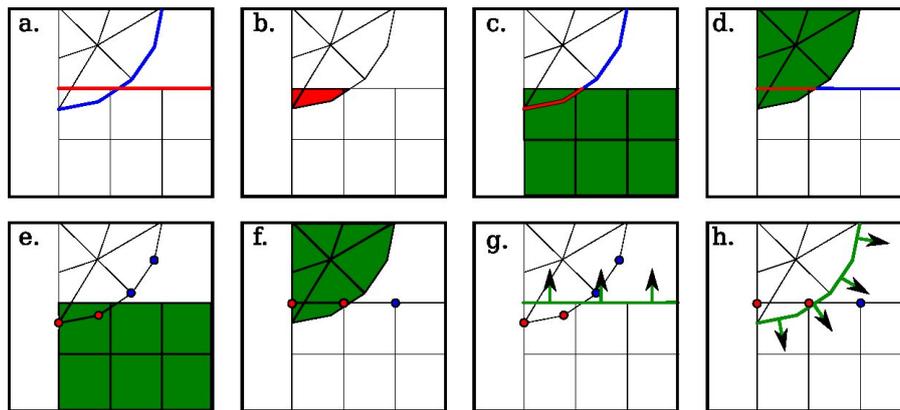


Figure 4.25: Conventional penetration estimations. Symmetric: a segment intersection; b volume intersection. Asymmetric: c,d segment in volume; e,f node in volume; g,h node under surface. Source[PhDYas11]

videogames[BookEri04; ArtCai+14], where the **Collision Detection (CD)** plays an important role in their respective fields.

The penetration is another aspect to take into account when detecting potential contact pairs. There is a whole science in the contact penetration or gap estimation. From the more conventional methods, Figure 4.25, through more advanced or recent methods such as the estimation of penetration by ray-tracing[ArtPR15a]. In this work we present an approach consistent with the *Mortar* formulation considered for the contact computation.

In what follows, a contact searching algorithm with **Bounding Volume Hierarchies (BVH)** can be applied (4.4.3.Tree structures), furthermore, there are different kinds of bounding volumes that can be used to improve this search, see 4.4.2.Bounding volumes. Finally, the penetration estimation is an important step too, and the proposed approach will be presented at 4.4.4.Penetration definition.

4.4.2 Bounding volumes

There are many kinds of **DOP** that can be used to define the bounding volumes, Figure 4.26a illustrates some of them. We address to the literature, particularly to the work of *Dinas*[ArtDB15], which summarises each methodology and makes a comparison between them. There are several different ways to define a bounding volume for a geographical region.

The simplest **DOP** probably will be to consider the nodes inside a radius, a radius search or **BS**. Another very simple solution, define a cuboid whose surfaces are parallel to the $x - y$, $y - z$ and $x - z$ surfaces, is called **AABB**, but these two methods have the drawback of the lack of tightness, and many slave and master surfaces are likely to be tested for proximity even though they are not so close (Figure 4.26b). Another approach is the **OBB**, which is much tighter and therefore giving a better fit, but with an additional computational cost and memory consumption. Finally, it can be considered the **k-DOP** (convex hull) to define the bounding volumes, which are always tighter than **AABB** and may be tighter or looser than **OBB** depending of the geometries of the objects. However the **k-DOP** are much simpler, cheaper and more efficient for the detection of **k-DOP** overlapping, which is quite relevant for our applications since deformation of contact bodies requires frequent updating of these volumes[ArtYL08].

In any case the Figure 4.26a as the decision to be taken results in a compromise solution between numerical cost, complexity and precision. For this reason the **OBB** has been chosen among all the options, as a intermediate solution that will satisfy the requirements. Originally a **BS** approach was chosen, but it was insufficient, as in order to detect all the contact pairs, a large enough radius factor must be taken into account (see Figure 4.27a), returning many *false potential pairs*, which impoverished the convergence of the problem. Additionally, this approach is very dependent on the radius r chosen, as can be seen in Figure 4.27b.

4.4.2.1 OBB implementation

Fortunately, we have plenty of documentation on how to implement this type of object thanks to its wide use in video game graphics engines. Our implementation is based on the one considered in the documentation of the *GeometricTools*[BookEbe99] library and the tutorials found for game development[OnlCho].

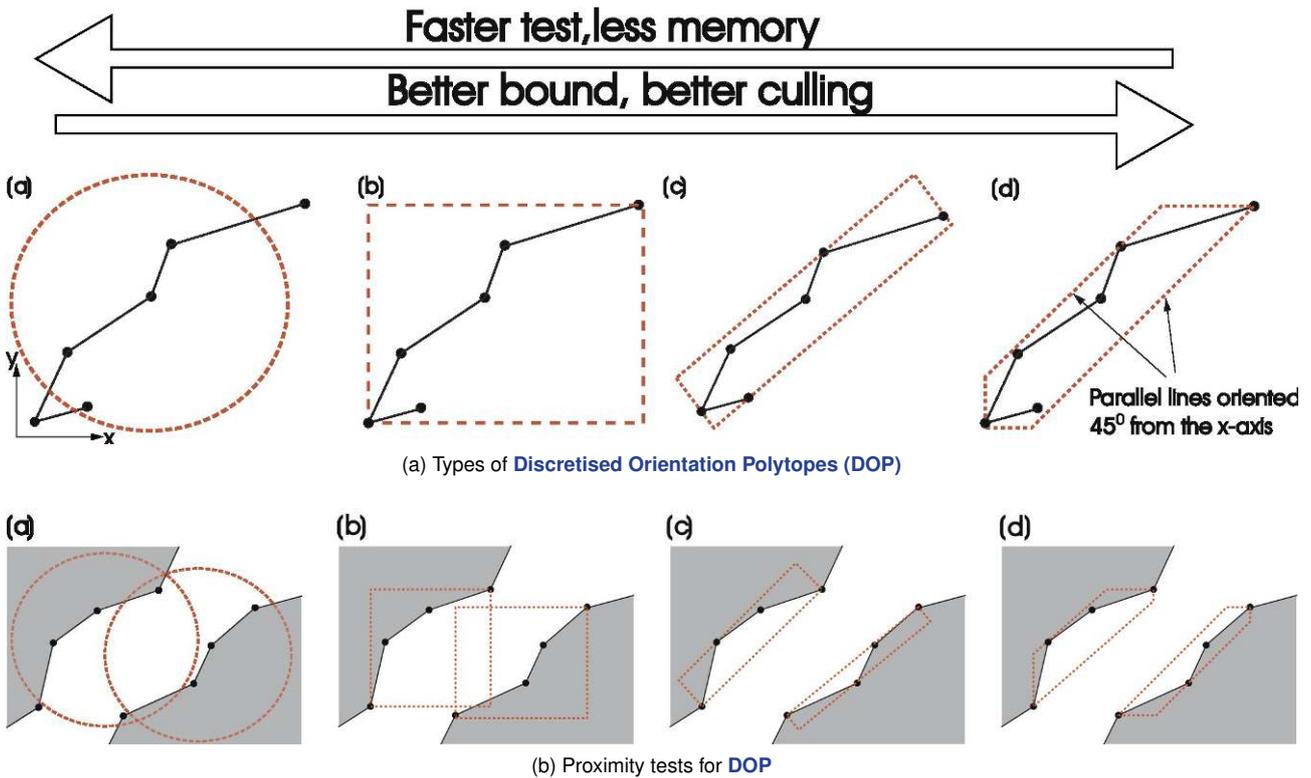


Figure 4.26: Types of bounding volumes. a) The radius search (**Bounding Spheres (BS)**). b) **Axis-Aligned Bounding Box (AABB)**. c) **OBB**. d) **Discrete Orientation Polytopes (k-DOP)** (convex hull). Inspired on [ArtYL08]

In order to define the collision algorithms, we can consider between two **OBB** we need to define the geometry which defines it properly. Considering we are in \mathbb{R}^3 . An **OBB** is defined [BookEbe99] by a centre \mathbf{C} , a set of right-handed orthogonal axes $\mathbf{A}_0, \mathbf{A}_1$ and \mathbf{A}_2 , and a set of real positive extends a_0, a_1 and a_2 . With this we can define the **OBB** as solid box following Equation (4.80a). Additionally Equation (4.80b) represents the eight vertices of the box, where $|\sigma_i| = 1$ for all i .

$$(4.80a) \quad \mathbf{C} + \sum_{i=0}^2 x_i \mathbf{A}_i : |x_i| \leq |a_i| \text{ for all } i$$

$$(4.80b) \quad \mathbf{C} + \sum_{i=0}^2 \sigma_i a_i \mathbf{A}_i$$

Our interest would be to build the **OBB** from a provided geometry in order to automate its construction. We have two options in 3D for surface geometries. The first approach takes the **AABB**, the lowest and highest point, and builds and axes and defines an orthonormal base. The second approach will be based on the normal which defines that surface geometry. Taking this normal, we can define a base, rotate the points to this base and get the point at the maximum distance from the geometry centre. Combining the tangent base in order to define a vector in the direction to the furthest point, then the third axis is taken as the cross product of the normal and this vector. This last method defines a less arbitrary *bounding box* than the first one. The difference can be seen in Figure 4.29, where we build the **OBB** for 4 triangles, 2 in the top and 2 over the first ones. The first one, Figure 4.29a, may look more regular, but just takes an arbitrary direction as all the nodes are in the same plane. The one from Figure 4.29b defines a larger **OBB**, and it is oriented to the furthest point from the centre, which is less arbitrary.

4.4.2.1.1 Collision detection :

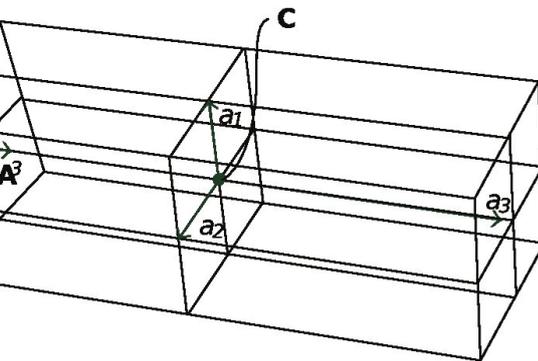
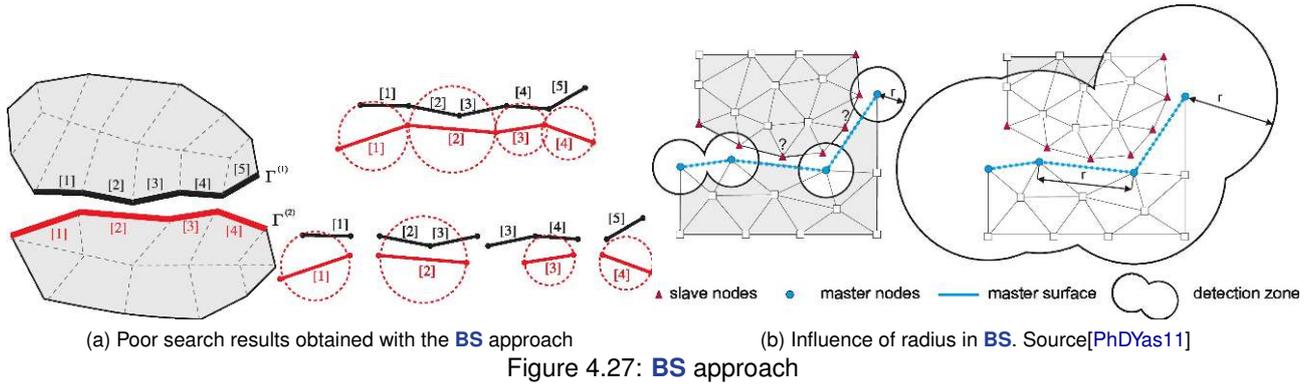


Figure 4.28: **OBB**. Source[ArtRom11]

The simplest collision detection consists in two steps, a first stage where we check that the points of one **OBB** are inside the other, the second stage consists in check if there is an intersection between the faces of the **OBB**. For the first step, we need to define the *inside* operations for a **OBB**. In order to do, the simplest procedure consists in rotating the points so these points became oriented with the *Cartesian* axis, Equation (4.81) and (4.82), and therefore consider the standard **AABB** *inside* algorithm.

For 2D we consider the following signs:

$$(4.81a) \quad \begin{aligned} sign_x^{2D} &= [-1.0 \quad 1.0 \quad 1.0 \quad -1.0] \\ sign_y^{2D} &= [-1.0 \quad -1.0 \quad 1.0 \quad 1.0] \end{aligned}$$

For each point i of the destination **OBB**:

$$(4.81b) \quad \mathbf{x}_i^{aabb} = \mathbf{C}^{dest} + sign_{xi}^{2D} \mathbf{A}_0^{dest} a_0^{dest} + sign_{yi}^{2D} \mathbf{A}_1^{dest} a_1^{dest}$$

We check if is inside the origin **OBB**:

$$(4.81c) \quad \|\mathbf{x}_{xi}^{aabb} - \mathbf{C}_x^{orig}\| \leq a_0^{orig} \quad \text{and} \quad \|\mathbf{x}_{yi}^{aabb} - \mathbf{C}_y^{orig}\| \leq a_1^{orig}$$

For 3D we consider the following signs:

$$(4.82a) \quad \begin{aligned} sign_x^{3D} &= [-1.0 \quad 1.0 \quad 1.0 \quad -1.0 \quad -1.0 \quad 1.0 \quad 1.0 \quad -1.0] \\ sign_y^{3D} &= [-1.0 \quad -1.0 \quad 1.0 \quad 1.0 \quad -1.0 \quad -1.0 \quad 1.0 \quad 1.0] \\ sign_z^{3D} &= [-1.0 \quad -1.0 \quad -1.0 \quad -1.0 \quad 1.0 \quad 1.0 \quad 1.0 \quad 1.0] \end{aligned}$$

For each point i of the destination **OBB**:

$$(4.82b) \quad \mathbf{x}_i^{aabb} = \mathbf{C}^{dest} + sign_{xi}^{3D} \mathbf{A}_0^{dest} a_0^{dest} + sign_{yi}^{3D} \mathbf{A}_1^{dest} a_1^{dest} + sign_{zi}^{3D} \mathbf{A}_2^{dest} a_2^{dest}$$

We check if is inside the origin **OBB**:

$$(4.82c) \quad \|\mathbf{x}_{xi}^{aabb} - \mathbf{C}_x^{orig}\| \leq a_0^{orig} \quad \text{and} \quad \|\mathbf{x}_{yi}^{aabb} - \mathbf{C}_y^{orig}\| \leq a_1^{orig} \quad \text{and} \quad \|\mathbf{x}_{zi}^{aabb} - \mathbf{C}_z^{orig}\| \leq a_2^{orig}$$

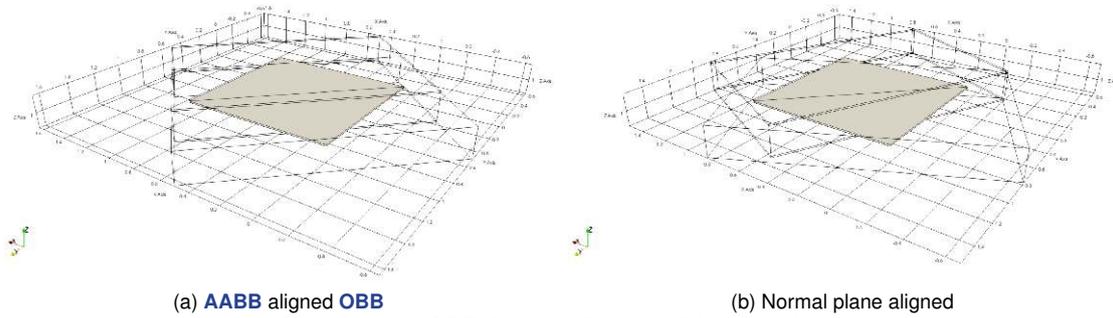


Figure 4.29: **OBB** from surfaces build comparison

For the intersection of the faces, the very same algorithms considered for the intersections of lines in 2D and quadrilaterals in 3D can be considered here. In total, the number of checks necessities in order to check the intersection is **24** in 2D and **52** in **3D**. Of course, not all checks are required. In case of one positive check we already know that there is at least one intersection.

4.4.2.1.2 Collision detection with **SAT** :

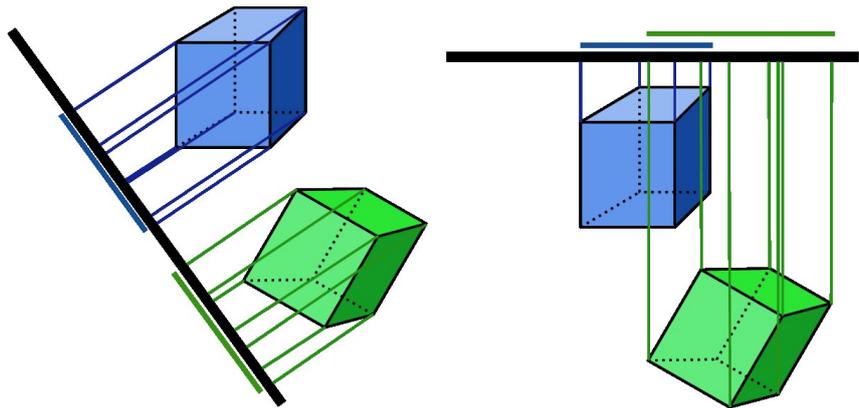


Figure 4.30: Separating axis theorem

This algorithm is more refined than the previous one. In this case instead of direct check each one of the nodes and faces, we apply a very simple geometrical concept. **SAT**[ArtRom11] is based on the separating axis theorem, Figure 4.30, which states that it is sufficient to find one axis that separates *A* and *B* to be sure that they are disjoint and so do not overlap. Therefore, if we are able to draw a line to separate two polygons, then they do not collide (Figure 4.31). This concept can be used not only between **OBB** but also between general polygons or even a polygon and a circle. The definition of the **SAT** consists on two steps, during the first step we define each one of the potential planes to be checked, the definition of each plane is found in Equation (4.83), and secondly we check if there is a separation plane between the two **OBB** with the one presented on (4.84).

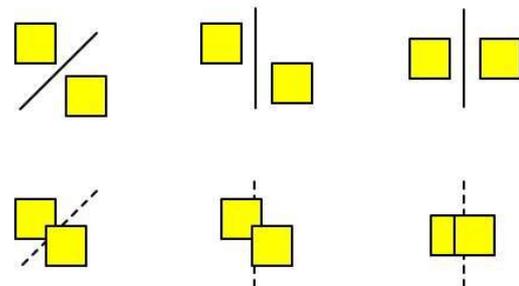


Figure 4.31: **SAT** concept. Source[OnlCho]

The normal vectors to be considered in 2D correspond with the right-handed orthogonal axes \mathbf{A}_0 and \mathbf{A}_1 from

both **OBB**, and the following *cross-products*:

$$(4.83a) \quad \mathbf{n} \begin{cases} \mathbf{A}_0^{dest}, \mathbf{A}_1^{dest}, \mathbf{A}_0^{orig}, \mathbf{A}_1^{orig} \\ \mathbf{A}_0^{dest} \times \mathbf{A}_0^{orig}, \mathbf{A}_0^{dest} \times \mathbf{A}_1^{orig} \\ \mathbf{A}_1^{dest} \times \mathbf{A}_0^{orig}, \mathbf{A}_1^{dest} \times \mathbf{A}_1^{orig} \end{cases}$$

The normal vectors to be considered in 3D correspond with the right-handed orthogonal axes \mathbf{A}_0 , \mathbf{A}_1 and \mathbf{A}_2 from both **OBB**, and the following *cross-products*:

$$(4.83b) \quad \mathbf{n} \begin{cases} \mathbf{A}_0^{dest}, \mathbf{A}_1^{dest}, \mathbf{A}_2^{dest}, \mathbf{A}_0^{orig}, \mathbf{A}_1^{orig}, \mathbf{A}_2^{orig} \\ \mathbf{A}_0^{dest} \times \mathbf{A}_0^{orig}, \mathbf{A}_0^{dest} \times \mathbf{A}_1^{orig}, \mathbf{A}_0^{dest} \times \mathbf{A}_2^{orig} \\ \mathbf{A}_1^{dest} \times \mathbf{A}_0^{orig}, \mathbf{A}_1^{dest} \times \mathbf{A}_1^{orig}, \mathbf{A}_1^{dest} \times \mathbf{A}_2^{orig} \\ \mathbf{A}_2^{dest} \times \mathbf{A}_0^{orig}, \mathbf{A}_2^{dest} \times \mathbf{A}_1^{orig}, \mathbf{A}_2^{dest} \times \mathbf{A}_2^{orig} \end{cases}$$

In 2D the algorithm for checking if there is a separation plane, being \mathbf{n} the normal which defines the plane:

$$(4.84a) \quad \begin{aligned} & \|(\mathbf{C}^{dest} - \mathbf{C}^{orig}) \cdot \mathbf{n}\| > \\ & \|(\mathbf{A}_0^{dest} a_0^{dest}) \cdot \mathbf{n}\| + \|(\mathbf{A}_1^{dest} a_1^{dest}) \cdot \mathbf{n}\| + \\ & \|(\mathbf{A}_0^{orig} a_0^{orig}) \cdot \mathbf{n}\| + \|(\mathbf{A}_1^{orig} a_1^{orig}) \cdot \mathbf{n}\| \end{aligned}$$

In 3D the algorithm for checking if there is a separation plane:

$$(4.84b) \quad \begin{aligned} & \|(\mathbf{C}^{dest} - \mathbf{C}^{orig}) \cdot \mathbf{n}\| > \\ & \|(\mathbf{A}_0^{dest} a_0^{dest}) \cdot \mathbf{n}\| + \|(\mathbf{A}_1^{dest} a_1^{dest}) \cdot \mathbf{n}\| + \|(\mathbf{A}_2^{dest} a_2^{dest}) \cdot \mathbf{n}\| + \\ & \|(\mathbf{A}_0^{orig} a_0^{orig}) \cdot \mathbf{n}\| + \|(\mathbf{A}_1^{orig} a_1^{orig}) \cdot \mathbf{n}\| + \|(\mathbf{A}_2^{orig} a_2^{orig}) \cdot \mathbf{n}\| \end{aligned}$$

The resulting number of checks therefore to be evaluated in this case would be of **8** for the 2D case and **15** in 3D.

4.4.3 Tree structures

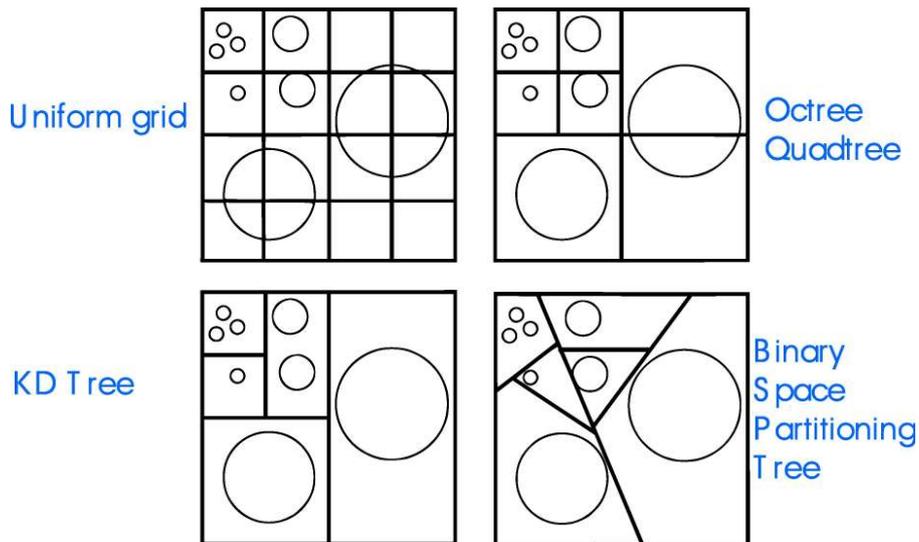


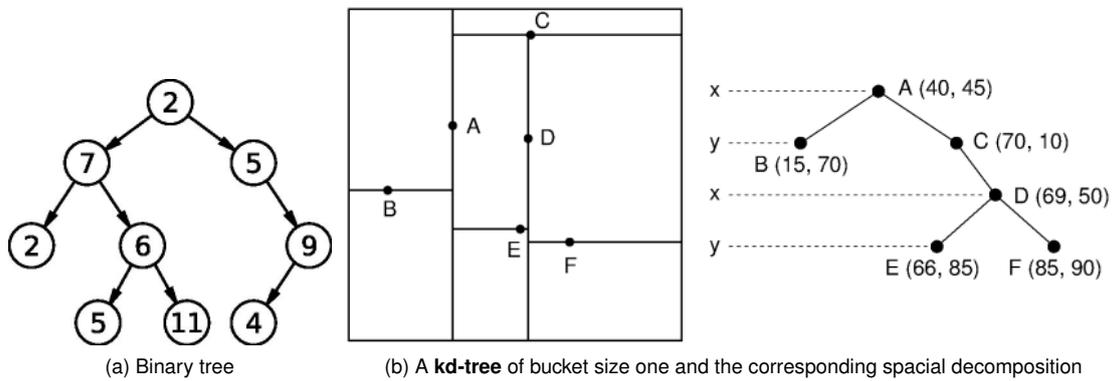
Figure 4.32: Space subdivision approaches

The tree structures are computer data structures that emulate a tree structure with a set of binary nodes[Book-FSS12], see Figure 4.33a. A node that is upper in the hierarchy respect to other nodes which is connected with him is called **parent node**, and node below is denominated **child node**. The node in the top of the tree is called **root node**, while the nodes in the bottom are called **leaf nodes**.

A data structure which stores the data points with respect to their position in space is needed to perform a search for neighbours. In two-dimensional spaces, the standard data structure is a **quadtree**, and the **octree** is analogue in three-dimensional space.

The **kd-tree** data structure is based on a recursive subdivision of space into disjoint hyperrectangular regions called cells (see Figure 4.33b). Each node of the tree is associated with a region called box, and is associated with a set of data points that lie within this box. The root node of the tree is associated with a bounding box that contains all the data points.

In addition to the data points themselves, a **kd-tree** is specified by two additional parameters, the **bucket size** (threshold of the number of data points associated with a node) and a **splitting rule** (which determine how a hyperplane is selected). For more details look up the literature[ArtYL08; BookPhD06; OnIRtr].



(a) Binary tree (b) A **kd-tree** of bucket size one and the corresponding spatial decomposition
Figure 4.33: Tree structures

The *binary structure tree* that will be considered in this work will be therefore the **kd-tree**. This *tree* is already available in the **Kratos** code.

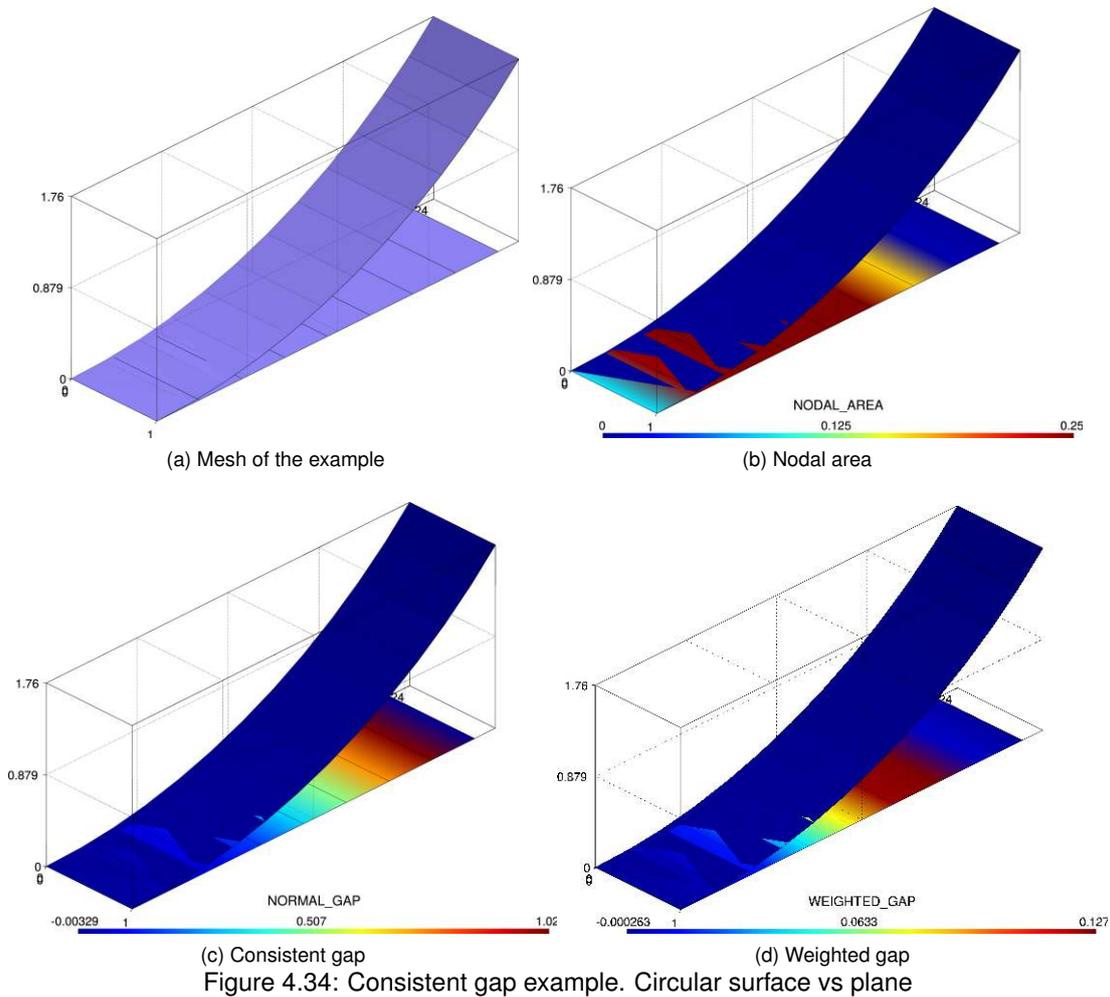
4.4.4 Penetration definition

As we have already mentioned, the main difference in explicit and implicit simulation detection is that in the former case we predict possible contact phenomena, and we define potential contact pairs before penetration happens. This means that slave nodes approaching master surfaces have to be detected at a certain distance, which is usually denoted as **Maximal Detection Distance (MDD)**. The **MDD** is a significative parameter of the detection procedure. This **MDD** should be considered as small as possible in order to avoid the creation of non-necessary contact pairs. In order to do that we should define an automatic procedure which defines this value in function of certain parameters, such as the element size (h). Once this is defined we need to estimate the penetration, so we can compare it with the **MDD**.

For that, we highlight that contact penetration estimation has its own advantages and disadvantages, and usually these techniques are uncoupled of the contact formulation considered. In order to compute a consistent penetration, or gap (g_n), with the weighted gap (\tilde{g}_n) computed during the contact contributions on 4.3.3.4.2. **Mortar operators** we will consider this very same *Mortar* formulation as the base concept.

This is done using the *Mortar* mapper developed in this work. The concept is to map the coordinates of one domain to the other, and once these coordinates have been mapped into a destination domain we can compute the consistent gap *node-by-node* applying the g_n formula, $g_n = -\mathbf{n} \cdot (\mathbf{x}_1 - \chi_h \mathbf{x}_2)$. The obtained gap is equal to the weighted gap divided by the total *Mortar* integration area which involves the node. This value has length units and therefore can be compared with some reference length. This very same concept can be applied in order to compute a consistent slip in case it is of interest (mostly for post-process motives).

We can illustrate this former concept with an example. The Figure 4.34 shows a full example, consistent in a circular surface and a plane surface, Figure 4.34a. As one of the sides is plane, it is easy to estimate the gap as the y coordinate of the circular side. The nodal area corresponds with the integrated region in each node, which is the whole section until the circular section does not intersect with the plane any more, see Figure 4.34b. With this we can



see as if dividing \bar{g}_n (Figure 4.34d) by the normal area, the consistent normal gap (g_n) is obtained (Figure 4.34c). This one coincides with the predicted values (the height to the circular region).

A more complex case is illustrated in the Figure 4.35, where the double curvature considered to test the *Mortar* mapper in *E.Mortar mapper* is considered to compute a consistent gap. The first figure shows the mesh, Figure 4.35a, and the second one in Figure 4.35b. In this last image, we can see like the gap varies according to the curvature and following the "parallel lines" that define this double curvature geometry. Some peaks appear in the nodes that belong to coarse mesh regions.

The details of the implementation of this mapper can be found in the respective appendix, *E.Mortar mapper*. The definition of the consistent gap is detailed in the algorithm 8.

4.4.5 Self-contact detection

4.4.5.1 Introduction

There are mechanical problems for which the determination of the contact pairs is not trivial, and especially complex are the problems where we do not enforce any *master-slave*⁹ (or *mortar* and *non-mortar*) relationship. For two or more independent bodies *a priori* we can assign these relationships, but this is not the case for *self-contact* problems, where there is only one body which contact with itself as the name indicates. This class of contact problems needs a particular contact detection procedure.

⁹We will consider the *master-slave* criteria considered in the formulation, as done by Popp[PhDPop12], where the slave conditions correspond with the one where the integration is performed.

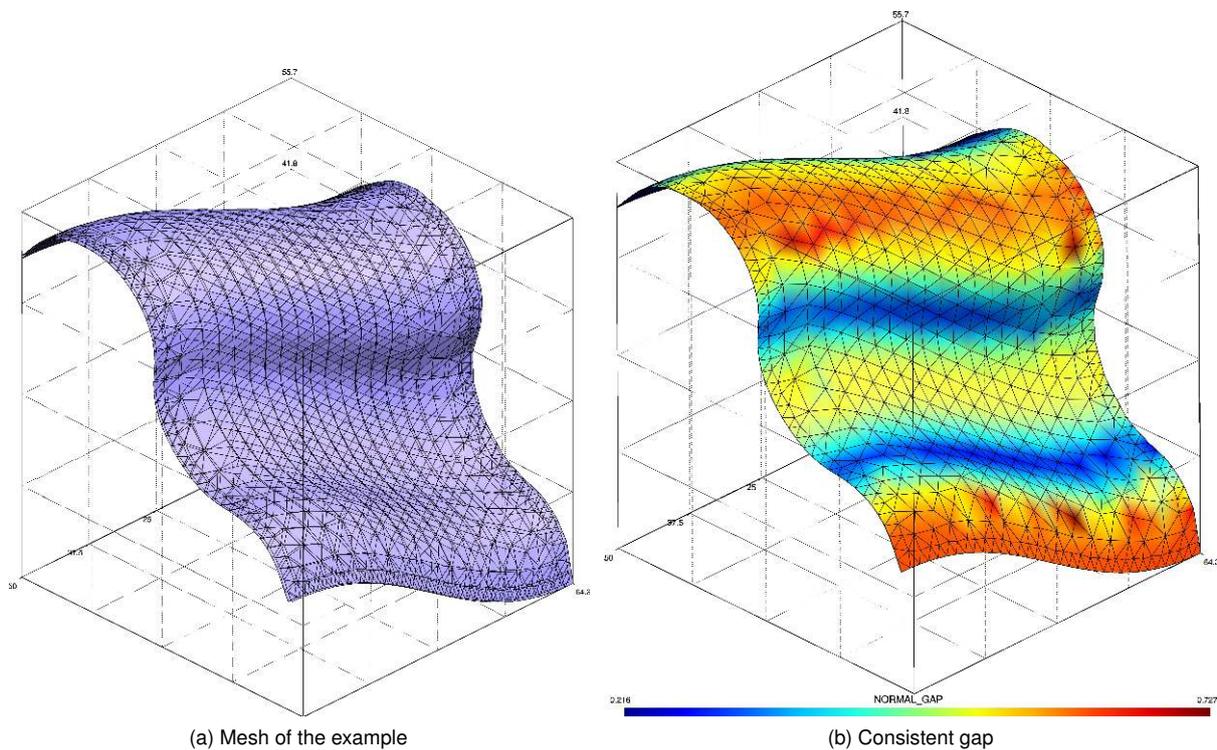


Figure 4.35: Consistent gap example. Double curvature surfaces

There is not much in the literature detailing the treatment of the *self-contact* problem, and the more detailed are probably the contributions of *Yastrebov*[PhDYas11; ArtYCF11; BookYas13], which are designed for the **NTS** integration. There are indeed a lot of publications[ArtSN17; ArtDem+12; ArtCam15] detailing this kind of problematic for fabric or very slender elements, as rods, where the *self-contact* is a common issue. A self-contact is more probable for thin or oblong solids, for which one or two dimensions are much smaller than others than for solids with all dimensions of the same order. In any case, the work presented here is more focused for solids than the slender geometries mentioned before.

4.4.5.2 Algorithm considered

The detection time is higher than for the contact of the same order with known *a priori* master-slave discretisation, because the preliminary stage requires several steps hardly parallelisable. This detection times vary in function of several parameters, like the normal discrimination, the initially assigned pairs, and in general all the steps involved in the presented algorithm.

A relevant, but also a big limitation when automatically assign contact pairs is that in principle a node cannot be master and slave at the same time, and that situation should be avoided. We will deactivate a condition when assigning a master/slave relation to the nodes and that condition holds master and slave nodes at the same time. This for example can be seen in the tubular example from 4.4.5.3.2. *Tubular detection*, where the conditions that share master and slave nodes will be deactivated. This is a big drawback, but in large meshes with small sizes, these conditions appear isolated and does not represent a significative amount respect the total.

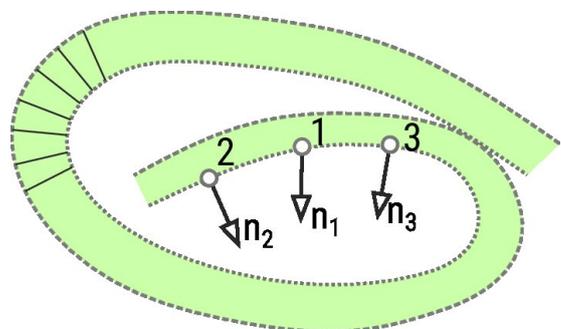


Figure 4.36: Self-contact normals discrimination

Before starting our algorithm for automatically pair conditions that may contact between them, we perform an initial search, where we potentially pair all conditions against all conditions. As this can be very expensive, numerically speaking, not only to perform, but also to filter later with our approach, we should pre-filter as many pairs as it is

possible for us. The method which provides the best results, and the simplest, is probably to check the similarity between the normals of the conditions. If the normals are close, not necessarily equal, therefore we can consider that these are pointing in the same direction and therefore will not deal with contact. This can be appreciated in Figure 4.36, where normals 1 and 3 are quite similar, so we can easily discriminate the pair 1 – 3. On the other hand, the difference between 1 and 2 is more subjective and it will depend on the threshold we consider to discriminate the pair 1 – 2.

Our proposal takes as a main hypothesis that the master/slave conditions are neighbours between them. This assumption is based on the fact that when we define manually certain conditions as the master or slave usually we define a region for each domain. Therefore, our goal is to replicate or *mimic* the behaviour that a human would consider when assigning the domains. Therefore, we will create a list of potential slave conditions and we will fill this list following a neighbourhood criterion. The idea is to create a list¹⁰ of conditions, that we will fill based on the proximity respect to the condition previously added to the list, except if we have already assigned that condition as a master, in that case we will jump to the next candidate.

Algorithm 4 Self-contact detection algorithm

```

1: procedure SELF-CONTACT DETECTION ALGORITHM
2:   We perform an initial search with all conditions againsts all conditions
3:   for all node  $\in$  ContactMeshnodes do
4:     Reset SLAVE flag
5:     Reset MASTER flag
6:   for all condition  $\in$  ContactMeshconditions do
7:     Reset SLAVE flag
8:     Reset MASTER flag
9:     Reset ACTIVE flag
10:  We create a set ordered_conditions that will contain all conditions in order of proximity between them
11:  We add as first condition to this set one chosen arbitrarily
12:  We create a map containing the boundaries of all conditions
13:  for all condition  $\in$  ContactMeshconditions do
14:    for all boundary  $\in$  conditionboundaries do
15:      Insert the connectivity of the boundary and the geometry where it belongs
16:  for all condition  $\in$  ordered_conditions do
17:    for all boundary  $\in$  conditionboundaries do
18:      if A neighbour is found then
19:        Add that neighbour to ordered_conditions
20:      else
21:        Add arbitrary condition not added previously to ordered_conditions
22:  for all condition  $\in$  ordered_conditions do
23:    for all pair  $\in$  conditionpotential – pairs do
24:      if Condition are neighbours then
25:        Skip to the next pair
26:      else
27:        if Check the nodes of the pair are not SLAVE then
28:          Assign MASTER flag to condition and nodes in the pair condition
29:          Clear pair condition database
30:        if At least one pair is MASTER then
31:          Assign SLAVE flag to condition and nodes
32:  for all condition  $\in$  ContactMeshconditions do
33:    if condition shares MASTER/SLAVE nodes then
34:      Unset ACTIVE flag to condition
35:    else
36:      Set ACTIVE flag to condition

```

¹⁰We have considered `std::unordered_set` in particular.

The previous list will be filled at first by an arbitrary condition, and starting from this first condition the rest of the list will be filled. In order to determine the neighbours of each condition, we will create a map¹¹ containing all the boundaries nodes and the original conditions where it belongs. Later, with this list of boundaries we will fill the set of conditions progressively, adding a neighbour condition to the last one added. If no neighbour is found we will add other arbitrary conditions, not added previously.

Once we have filled the set of conditions, we proceed with the potential master/slave assignment. We iterate over this set of ordering conditions, if the condition has been previously marked as masters we skip it, if for we check their potential pairs. If the conditions are neighbours, we skip to the next potential pair. In the opposite case, we must check that no node of the potential pair is already assigned as slaves, if this is the case we can assume this pair is a master surface, and we will clear its potential pairs, simplifying our assigning process. If the condition has any master condition assigned, we can assume is a slave condition. Finally, we will check if the conditions share master/slave nodes, if that is the case we deactivate the condition, if not we can consider is an active one.

All the previously stated procedure is summarised in the Algorithm 4, which additionally includes some additional implementation details, in any case we present it in the most *pseudo-code* way possible.

4.4.5.3 Examples

4.4.5.3.1 Planes detection :

The first example we will show is a very simple case where there is no node shared between the *domains*, and therefore we can easily manually assign the master/slave roles. If the procedure works properly will return the expected result, which means that one plane will be fully master, and the other one will be fully slaves. This simple case is part of the `ContactStructuralMechanicsApplication` test suite and it is located in the following [file](#).

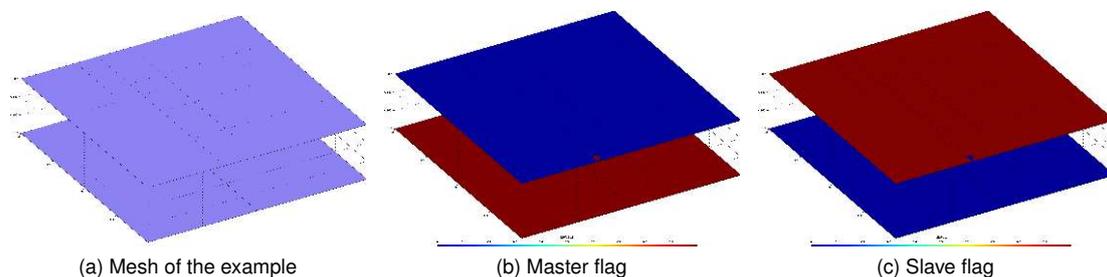


Figure 4.37: Self-contact detection simplest case, two parallel planes

The result can be seen in Figure 4.37, where we can appreciate like the lower plane is fully master (Figure 4.37b), and the upper plane fully slave (Figure 4.37c).

4.4.5.3.2 Tubular detection :

The case consists of a very simplified ring, formed on quadrilaterals. The complexity comes from the fact that there are shared nodes between the potential domains, which means that as we confront a pure *self-contact* problem. In this case we assign as initial potential pairs everything with everything, our procedure will filter this lately. This case can be found as a test in [here](#).

The first example is the case considering only one element of thickness. See Figure 4.38, were due to the initial assignment of everything with everything, the number of master conditions (Figure 4.38b) is larger than the number of the slave ones (Figure 4.38c). This is due to the fact that as we follow the procedure, we set as master all the potential pairs, before jump to the next potential slave condition. Additionally can be seen like those two conditions will be inactive, as they share master and slave nodes simultaneously.

The second example is a little bit more complex, in here we consider two elements of thickness. This increases the complexity, as the neighbourhood is not as simple as in the first case. The results obtained, Figure 4.39, look like an extension of the former case. In that case, nothing else to add to the comments previously stated.

¹¹We do this at the programming level considering `std::unordered_map` in this case.

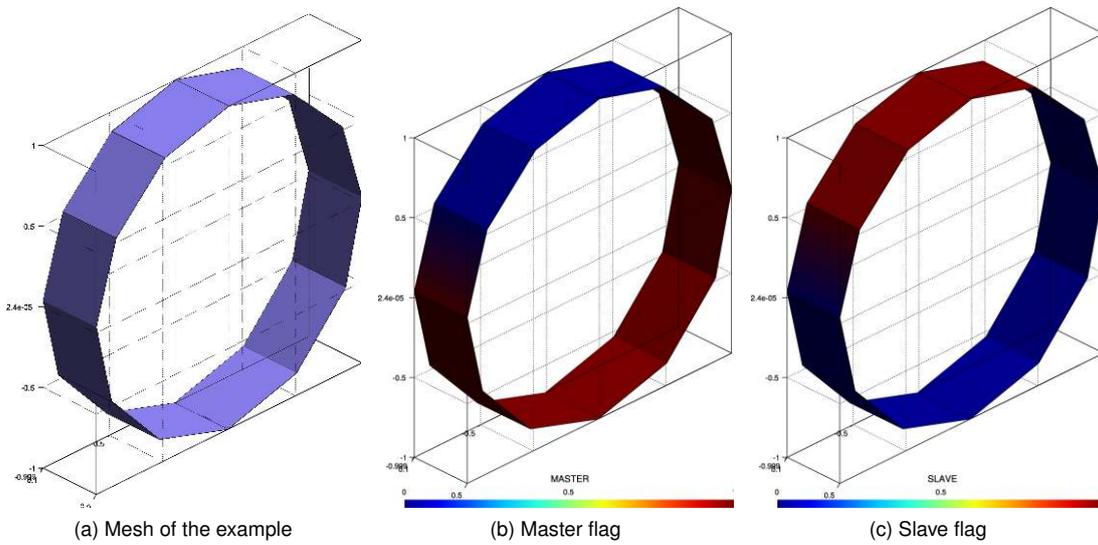


Figure 4.38: Self-contact detection for simple tubular case

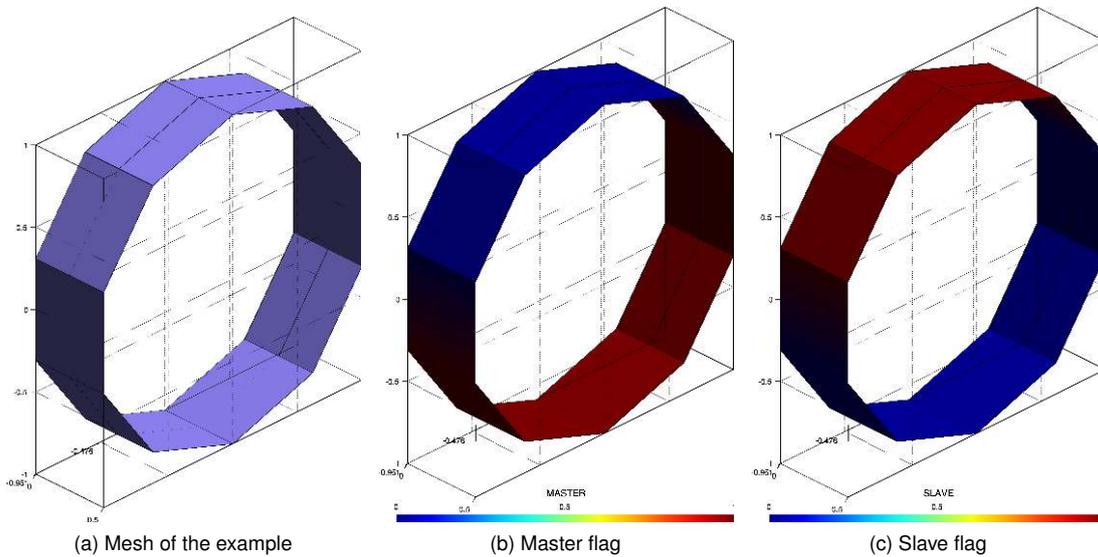
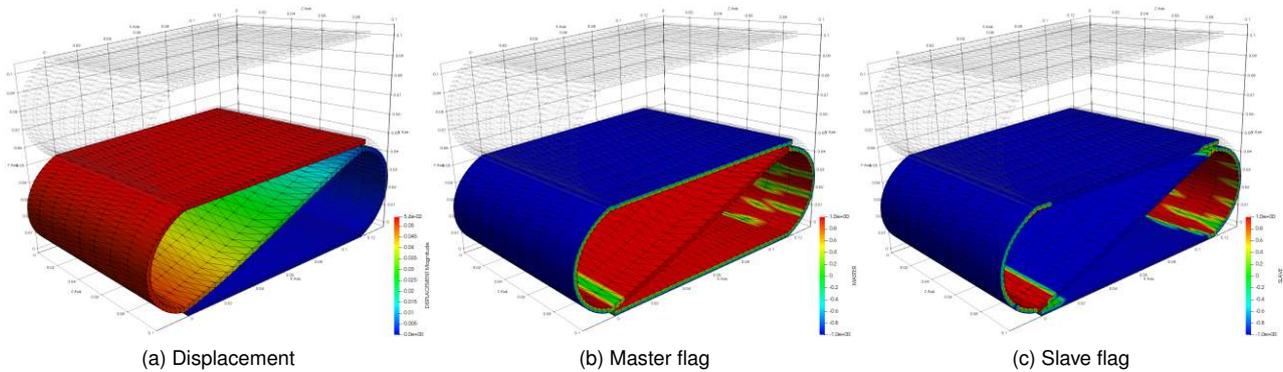
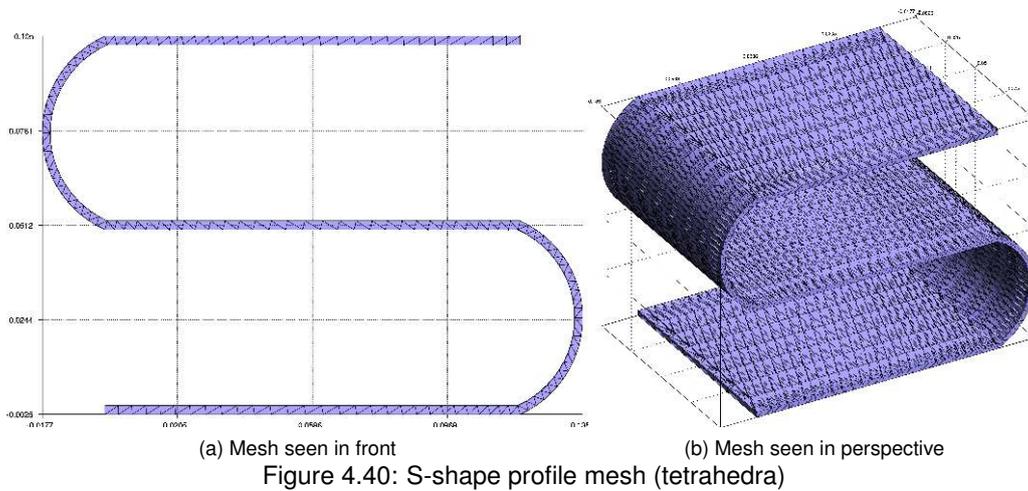


Figure 4.39: Self-contact detection for tubular case

4.4.5.3.3 Contact example (S-shape profile) :

The problem consists in a *S-shape* profile which is enforced to self-contact, where we impose a negative vertical displacement on the top face. The geometry and the mesh of the problem can be seen in Figure 4.40. It is relatively easy, but a bit tedious, to manually assign the master/slave pairs, at least for the initial deformations. Because of this, we can consider it as a good example to test the procedure. In any case the automatic assignment is not trivial as the number of triangular faces is large and the neighbourhood must be properly detected. The example can be found in the *Kratos Examples repository*, in [here](#).

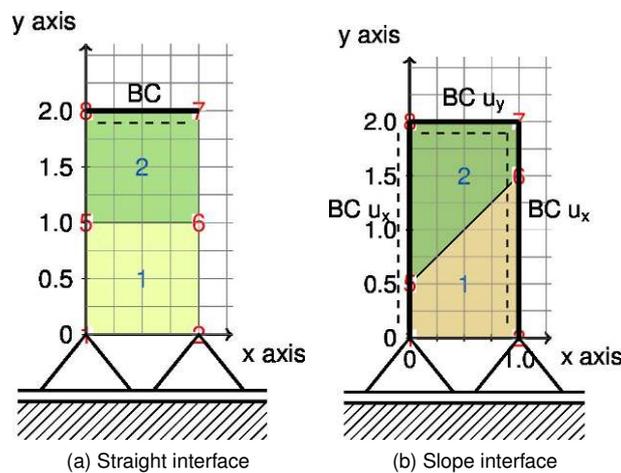
The results can be seen in Figure 4.41. Where on Figure 4.41a the resulting displacement can be seen, with the corresponding expected contact in the corners. Additionally the Figure 4.41b shows the master nodes, which are the vast majority and the Figure 4.41c the slave nodes, where the featured nodes correspond to the corners in contact. In order to better understand the result obtained it is relevant to remember the fact the domain considered as the one where to perform the integration is the slave side, therefore the default domain corresponds with the master one. Because of that, the most predominant domain the master side.



4.5 Numerical examples

The following section presents several examples commonly used in order to check the proper functioning of the **CCM** implementations. The tests presented will go from less complex to more complex, starting from the most basic patch tests to the most common and complex contact benchmarks. The examples are solved considering **ALM**.

4.5.1 Basic patch test



Here the most basic of all possible patch test is checked. These consist in to block sharing a clear interface between the domains. The result expected is a continuous gradient in the displacement in the normal direction of the interface for the frictionless contact, a much more extensive casuistry for frictional case which will be explained in detail below. Here we present two different cases, the first one from Figure 4.42a where the interface between two 1×1 squares is straight. The second case, from Figure 4.42b, where the interface is a slope between two non-regular quadrilaterals.

Body	E	ν	ρ
Die	$2.069 \times 10^{11} Pa$	0.29	1000
Block	$2.069 \times 10^{11} Pa$	0.29	1000

Table 4.4: Parameters considered for simple patch test

4.5.1.1 Frictionless

For both problems a vertical displacement of 0.1 meters will be imposed in the top face of the upper quadrilateral. The properties of the materials correspond with the ones shown in Table 4.6. With this given BC, the solution obtained for the first problem, Figure 4.43a, is the expected one, a continuous gradient of the displacement in the vertical direction. On the other hand, the result of the second one, Figure 4.43b, also corresponds with the expected behaviour, a continuous solution in the vertical direction.

One additional remark to the first case is the fact that we are considering the same *Poisson's ratio* (ν) for both bodies, the deformation of the interface is symmetric in the interface, giving the impression that the displacement is also tied in the tangential direction. But for example if one of the ν is taken as zero, this body will deform just vertically and the other one will experiment *Poisson's* effect.

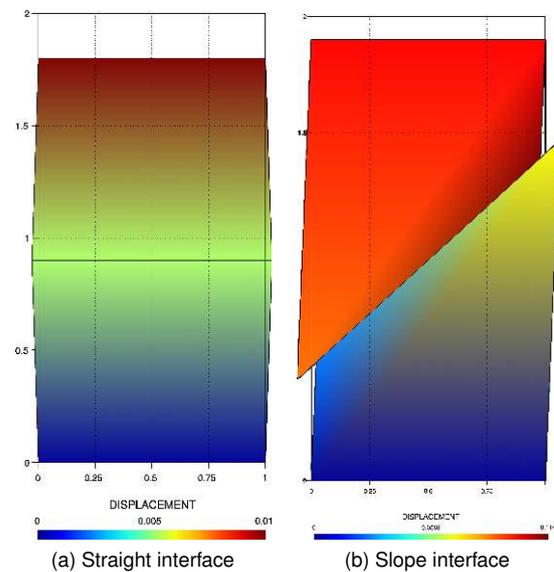


Figure 4.43: Solution for the frictionless simplest patch test

4.5.1.2 Frictional

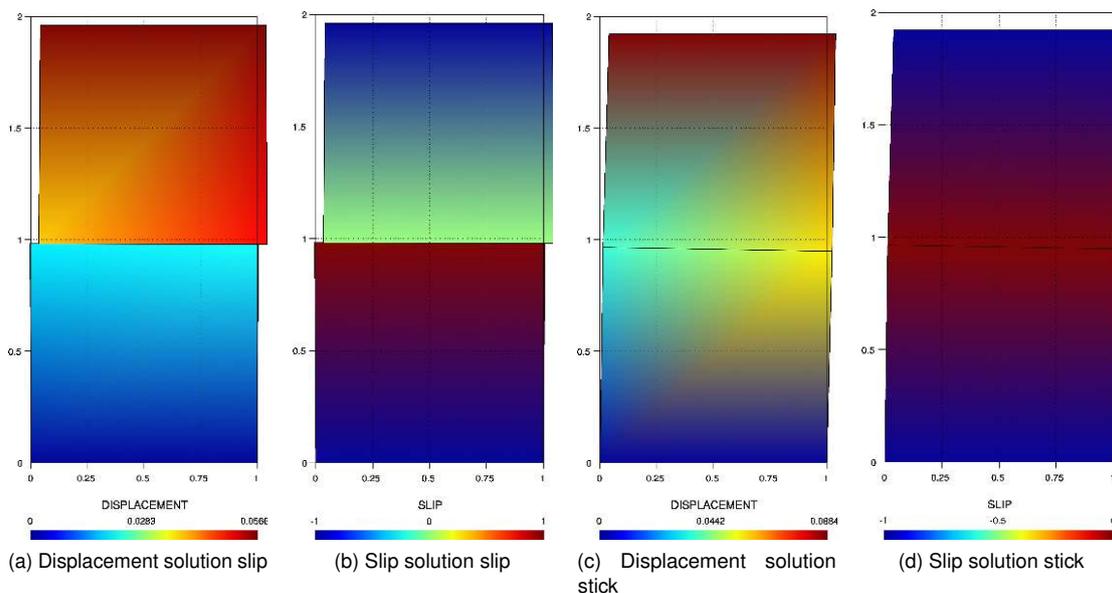


Figure 4.44: Geometry and solution for the frictional simplest patch test

Here for the sake of simplicity only the problem with straight interface, from Figure 4.42a, will be studied. In the study we need to differentiate between two possible states in the frictional case, the slip and the stick state. In here, in contrast with the frictionless case, we set a vertical load in the top face of the upper element, additionally a tangent load in order to trigger the tangent behaviour. This problem thanks to its simplicity allows us to check the correctness of the slip/stick state detection, where just adjusting the friction coefficient μ it is possible to change between one state and the other. The first case corresponds with the slip states, where in Figures 4.44a and 4.44b we can appreciate like a relative drift appears on the interface. On the other hand, on the Figures 4.44c and 4.44d, the solution for the stick state is shown, and can be seen as the two blocks move in solidarity. Notice that Figures 4.44b and 4.44d represent the `SLIP` flag, the value can be only 0 and 1, being 1 slip state, and 0 stick state.

4.5.2 Taylor patch test

The *Taylor[ArtTP]* patch test is slightly more complex problem than the former patch test. In this case the interface is not coincident between the two domains, additionally a distributed load is considered in the upper face of the domains. The load considered is $p = 10Pa$, and the properties of the materials correspond with the ones shown in the Table 4.5. We will present the solution for the 2D and 3D problem.

Body	E	ν	ρ
Die	$3 \times 10^3 Pa$	0.4	1000
Block	$3 \times 10^3 Pa$	0.4	1000

Table 4.5: Parameters considered for *Taylor[ArtTP]* patch test

4.5.2.1 2D

The Figure 4.45a illustrate the setup of the *Taylor* test in 2D. The solution obtained in the other hand corresponds with a continuous gradient of vertical displacement as shown in Figure 4.45b and a continuous vertical stress Figure 4.45c.

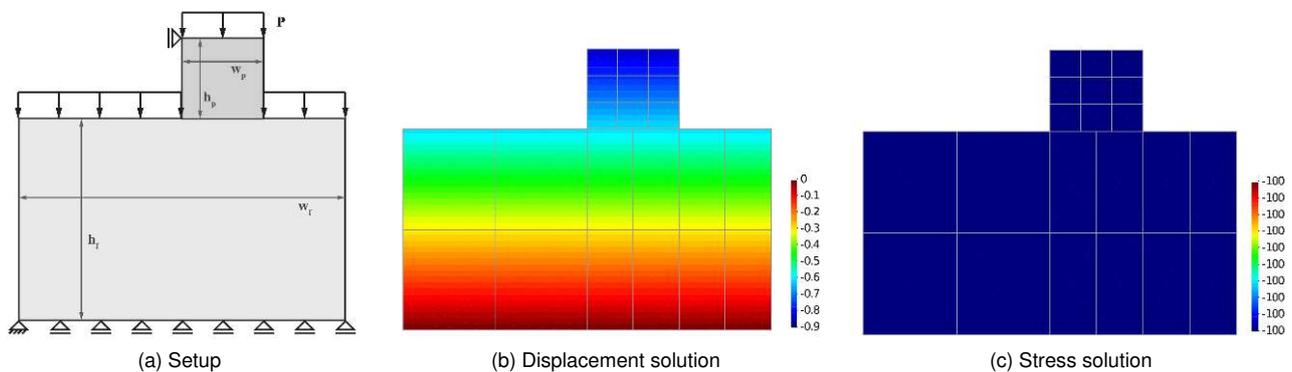


Figure 4.45: Solution for the *Taylor* patch test in 2D

4.5.2.2 3D

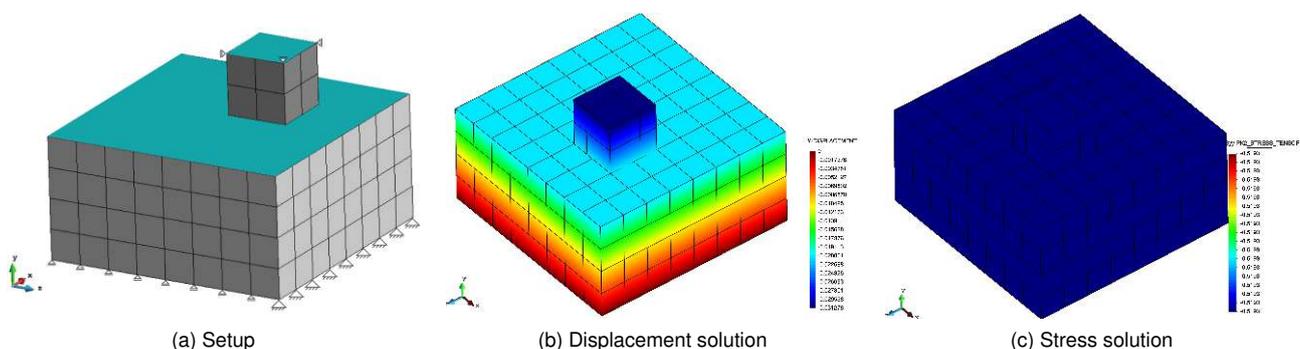


Figure 4.46: Solution for the *Taylor* patch test in 3D

For the 3D case, see Figure 4.46a, the same conclusions than in the 2D case can be applied here. Continuous gradient of displacement (Figure 4.46b) and continuous stress field (Figure 4.46c) in the vertical direction.

4.5.3 Friction base test

The following test, extracted from the work of *Dong*[ArtDon99], is a test which allows to study the effects in the contact frictional behaviour with respect to different values of the friction coefficient μ . The properties of the bodies are the ones shown in the Table 4.6, where μ has been considered with three different values, 0.25, 0.5 and 1. The geometry and mesh considered are the ones from Figure 4.47 and the load $q = 20000\text{KN/m}$. The solution is compared with the one in reference[ArtDon99] in Figure 4.48b, with a very good agreement. The solution from Figure 4.48a shows the detachment that appears in the interface for $\mu = 0.25$.

Body	E	ν	μ
Die	$2.1 \times 10^{11} \text{ Pa}$	0.29	1, 0.5, 0.25
Block	$2.1 \times 10^{11} \text{ Pa}$	0.29	1, 0.5, 0.25

Table 4.6: Parameters considered for simple patch test

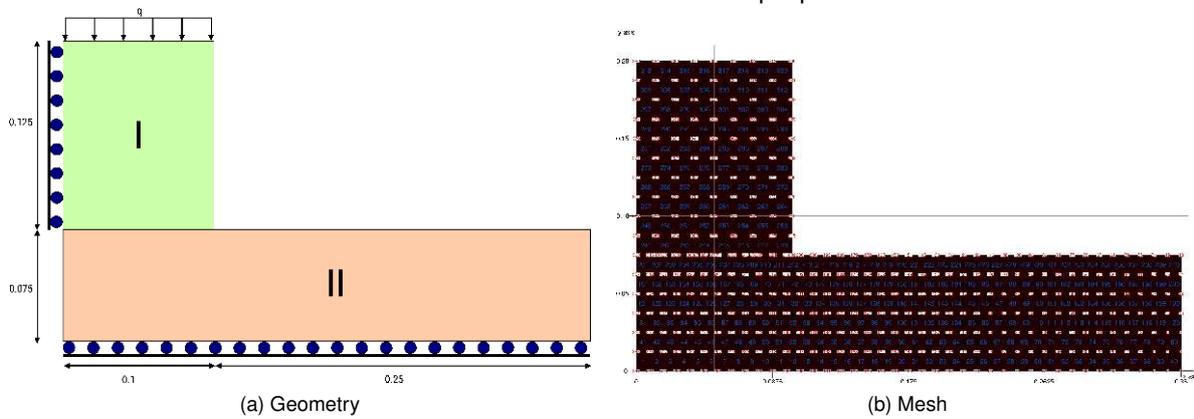


Figure 4.47: Friction problem from *Dong*[ArtDon99]

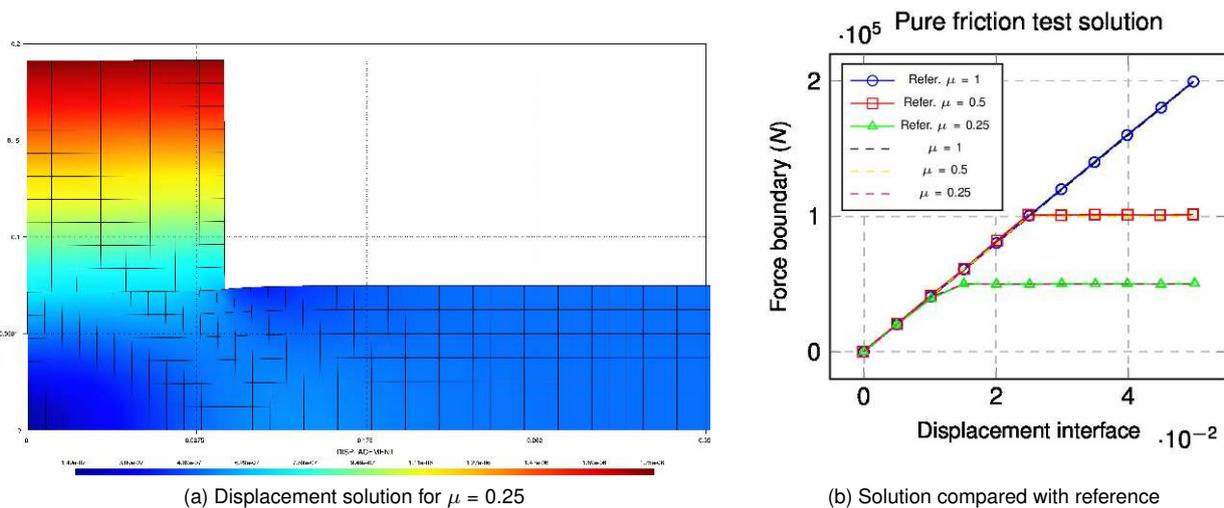


Figure 4.48: Solution for pure friction problem

4.5.4 Hertz problem

We have already highlighted the relevance of this test in the former sections. This is the most commonly used and extended benchmark for contact mechanics, originally published by *Hertz*[ArtHer82] in 1882. This benchmark has the advantage that the original solution is analytical and therefore we can compare with a solution that we have known a

priori for different combinations of material and geometries. Using as main reference the solution of *Zhu*[ArtZhu12] for the analytical solutions of the different *Hertz* problems. We will study basically two cases, the rigid plane-sphere contact and sphere-sphere¹² contact problem, both for 2D and 3D problems. It will be assumed we work under the hypothesis that we work in the domain of infinitesimal deformations.

4.5.4.1 2D

4.5.4.1.1 Plane-sphere :

The plane-sphere configuration from Figure 4.49 corresponds with the problem to solve, which requires in 2D to be solved considering axisymmetric formulation. We specialise the resolution of the problem for a radius equal to 6.1237 meters, and pressure of $P = 5 \times 10^5 Pa$. The properties considered for the two solids are represented on Table 4.7, where the plane can be considered *de facto* rigid. We have considered different mesh sizes, and we have compared the results between then and the respective error.

Body	E	ν
Sphere	$1 \times 10^8 Pa$	0.29
Block	$1 \times 10^{26} Pa$	0.29

Table 4.7: Parameters considered for 2D sphere-plane *Hertz* benchmark

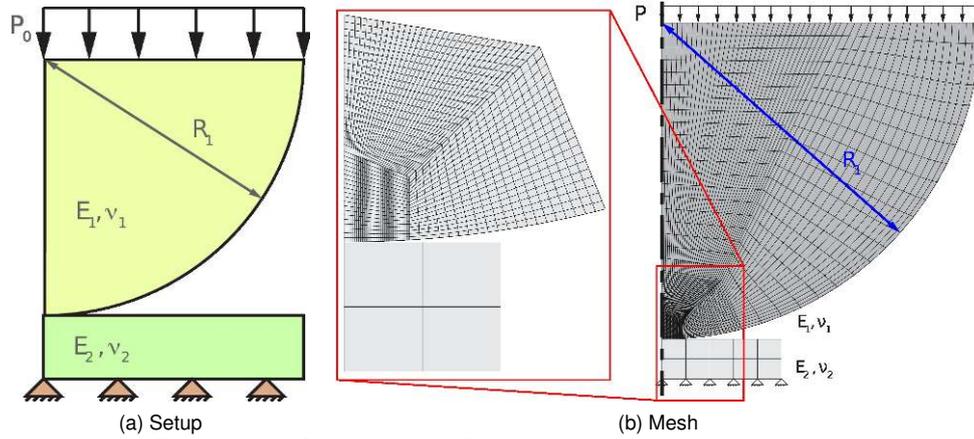


Figure 4.49: Setup for the 2D sphere-plane *Hertz* benchmark

The analytical solution is presented in Equation (4.85), taken from the reference[ArtZhu12]. The Equation (4.85c) gives the analytical solution for the contact pressure in all the domain b , then the Equation (4.85d) provides the vertical displacement of the contact interface.

$$(4.85a) \quad E_{eff} = \frac{1}{\frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}}, \quad r = \sqrt{x^2 + y^2}$$

$$(4.85b) \quad b = \sqrt[3]{\frac{3P\pi R^3(1-\nu^2)}{4E_{eff}}}, \quad p_0 = 3P \frac{R^2}{2b^2}, \quad d_0 = \frac{b^2}{R}, \quad f_0 = \sqrt[3]{\left(E_{eff} \sqrt{Rd_0^3}\right)^4}$$

$$(4.85c) \quad p_n = p_0 \sqrt{1 - \frac{r^2}{b^2}}$$

$$(4.85d) \quad y = -\frac{2}{3} p_0 \pi b^2 \sqrt{\left(1 - \frac{r^2}{b^2}\right)^3}$$

As previously stated, the solution obtained corresponds with the solution for different mesh sizes. Figure 4.50a presents the different solutions for the vertical displacement for different mesh sizes, and Figure 4.50b the same but

¹²In 3D, in 2D a contact cylinder-cylinder will be considered

for the contact pressure. It is notorious that the displacement solution converges even for very coarse meshes and the pressure solution it is more difficult to be converged, particularly in the contact frontier. In fact, it can be noticed that the finer mesh does not provide necessarily the better solution, and more intermediate meshes present better results.

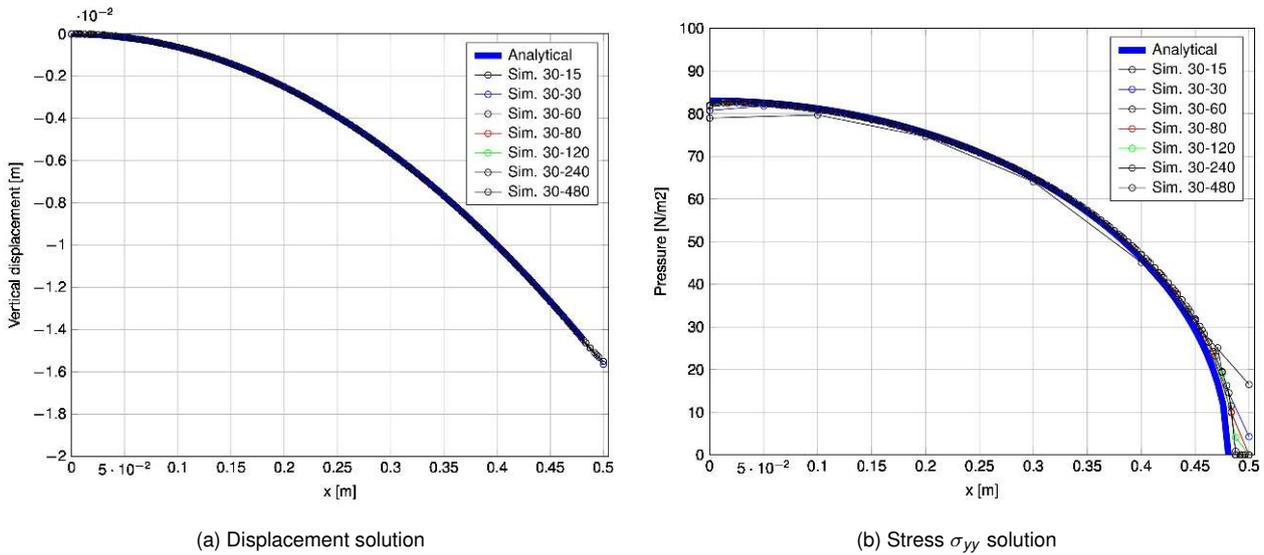


Figure 4.50: Solution compared for different mesh sizes for the 2D *Hertz* plane-sphere contact

This last statement can be seen in an easier manner in the solutions presented on Figure 4.51, where the error for the displacement (Figure 4.51a) and the pressure (Figure 4.51b) are compared for different mesh sizes respect the analytical solutions.

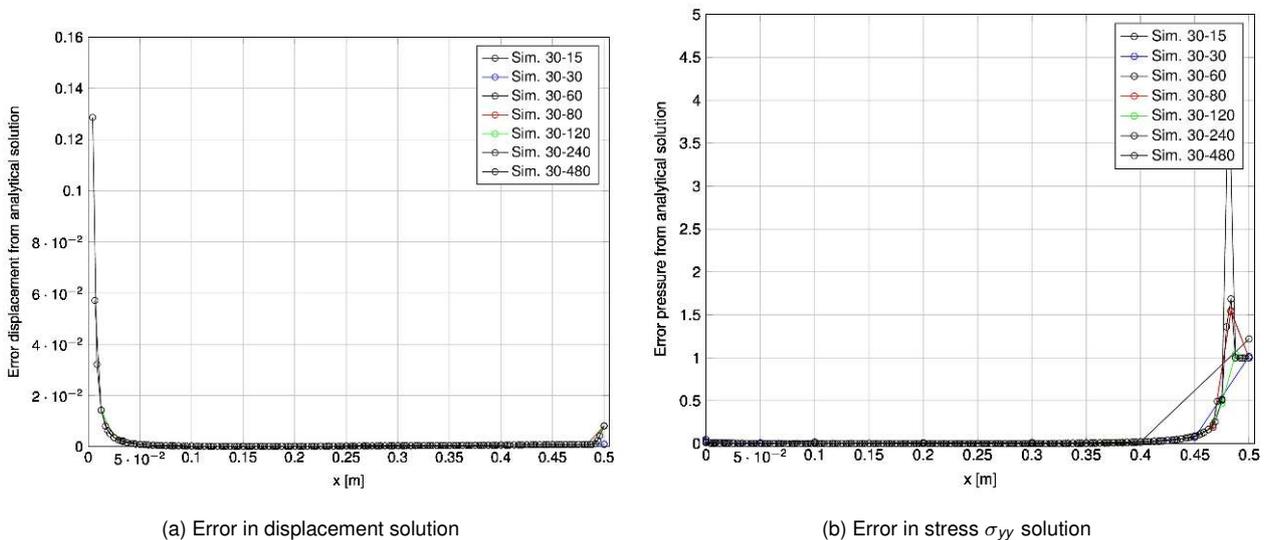


Figure 4.51: Error compared for different mesh sizes for the 2D *Hertz* plane-sphere contact

4.5.4.1.2 Cylinder-cylinder :

The configuration for this problem is shown in Figure 4.52. This consists in two infinite cylinders, not to be confused with spheres, that become into contact. We will consider the parameters from Table 4.8. Additionally the following values have been taken into account as **BC** and geometrical values: $q = 0.05851 Pa$, $p = 0.625 Pa$ and $R = 8m$.

4.5.4.1.2.1 Frictionless case :

Body	E	ν	μ
Upper cylinder	200Pa	0.3	0.2
Lower cylinder	200Pa	0.3	0.2

Table 4.8: Parameters considered for 2D cylinder-cylinder *Hertz* benchmark

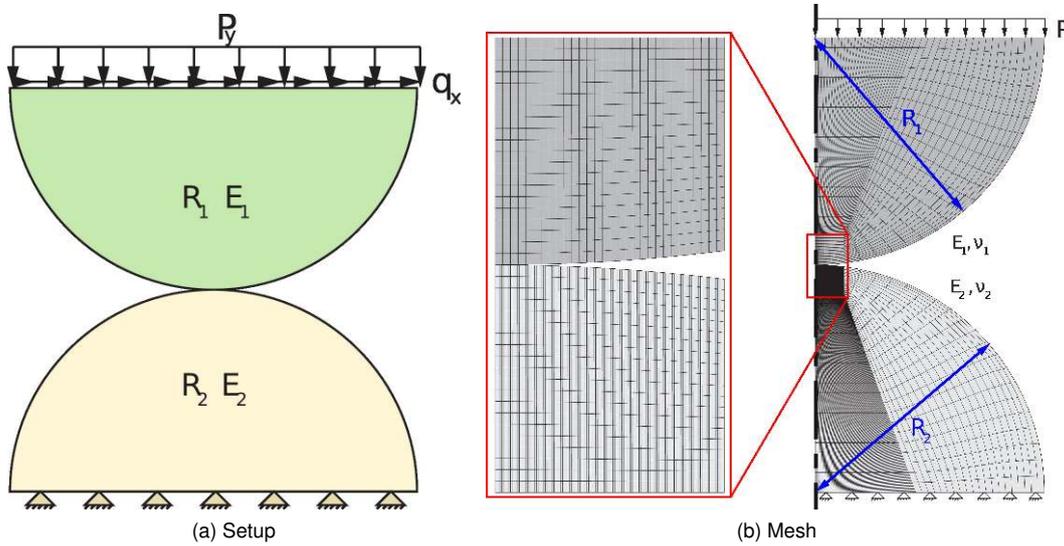
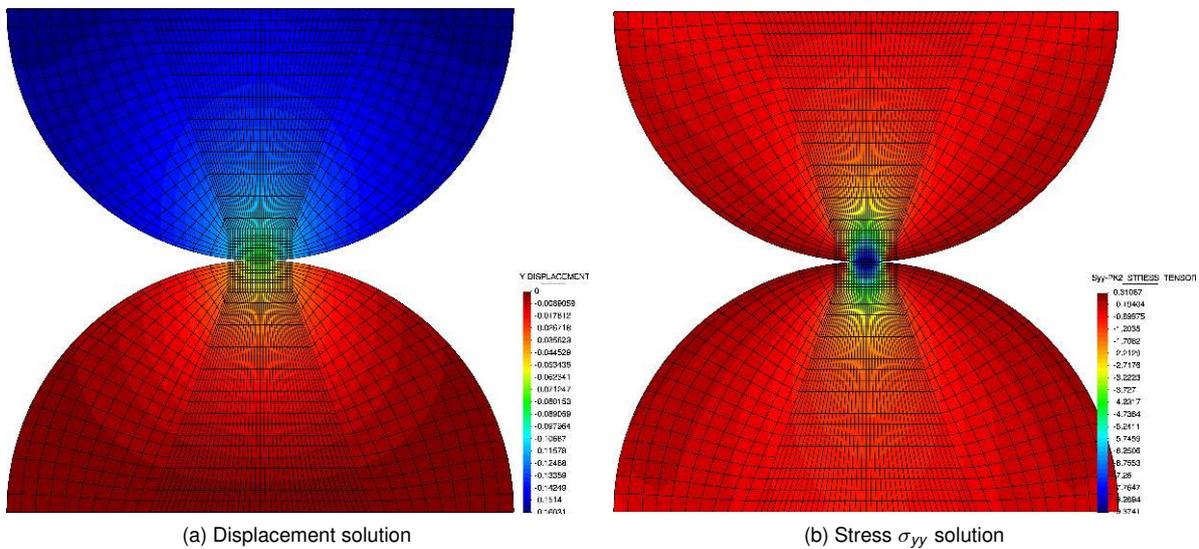


Figure 4.52: Setup for the two cylinders *Hertz* benchmark

First we present the solution for the frictionless part. Considering $\mu = 0$, the normal contact pressure corresponds with (4.86b). This is what is represented in Figure 4.53. The comparison with the analytical solution will be presented all together with the frictional solution.

(4.86a)
$$b = 2\sqrt{\frac{2R^2\rho(1-\nu^2)}{E\pi}}$$

(4.86b)
$$p_n = \frac{4R\rho}{\pi b^2}\sqrt{b^2 - x^2}$$



(a) Displacement solution (b) Stress σ_{yy} solution
Figure 4.53: Solution for the two cylinders frictionless *Hertz* benchmark

4.5.4.1.2.2 Frictional case :

The solution considered has been taken from Wang[ArtWZ13] and Gitterle[PhDGit12], and the resulting is compared with normal (4.86b) and tangential pressure (4.87b) of reference. Figure 4.54 presents and compares the solution with the reference, showing a very good agreement with it.

$$(4.87a) \quad c = b \sqrt{1 - \frac{q}{\mu p}}$$

$$(4.87b) \quad \begin{aligned} p_t &= \mu \frac{4Rp}{\pi b^2} (\sqrt{b^2 - x^2} - \sqrt{c^2 - x^2}) & \text{if } |x| \leq c \\ p_t &= \mu \frac{4Rp}{\pi b^2} \sqrt{b^2 - x^2} & \text{if } c < |x| \leq b \end{aligned}$$

Contact traction for frictional Hertz cylinder

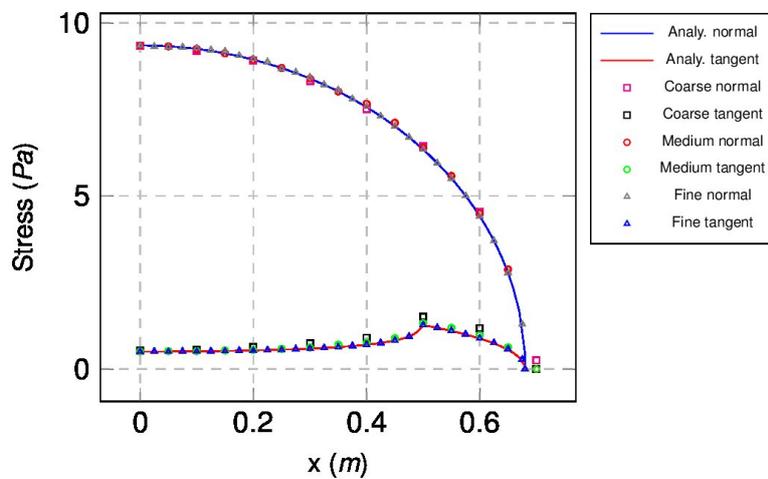


Figure 4.54: Solution for the two cylinders frictional Hertz benchmark

4.5.4.2 3D

4.5.4.2.1 Plane-sphere :

The description and solution for this problem correspond with the previously introduced at 4.5.4.1.1. Plane-sphere, as this was solved considering an axisymmetry. We will consider a sphere of 12.2474 meters of diameter with a load of $p = 5 \times 10^5 Pa$ and the material properties from Table 4.9. We consider two different mesh sizes, Figure 4.55 shows the finer mesh. The calculation is done in just one static step. This problem can be found in the following link.

We compare the resolution of both meshes with the reference solution. Figure 4.56 exposes the displacement and VM stress solution. For the displacement we got a very good agreement with the analytical solution in both cases, but particularly good in the case of the refined mesh, Figure 4.57a, in the case of the contact pressure we got a better approach within the finer case, Figure 4.58a.

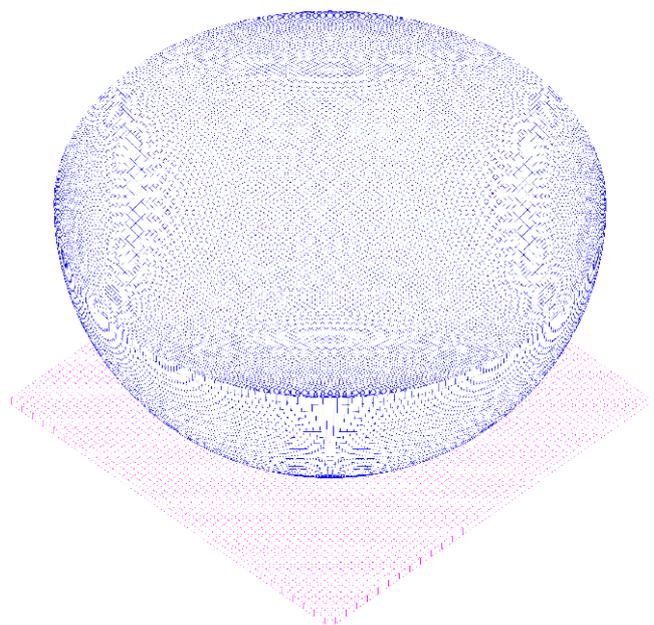
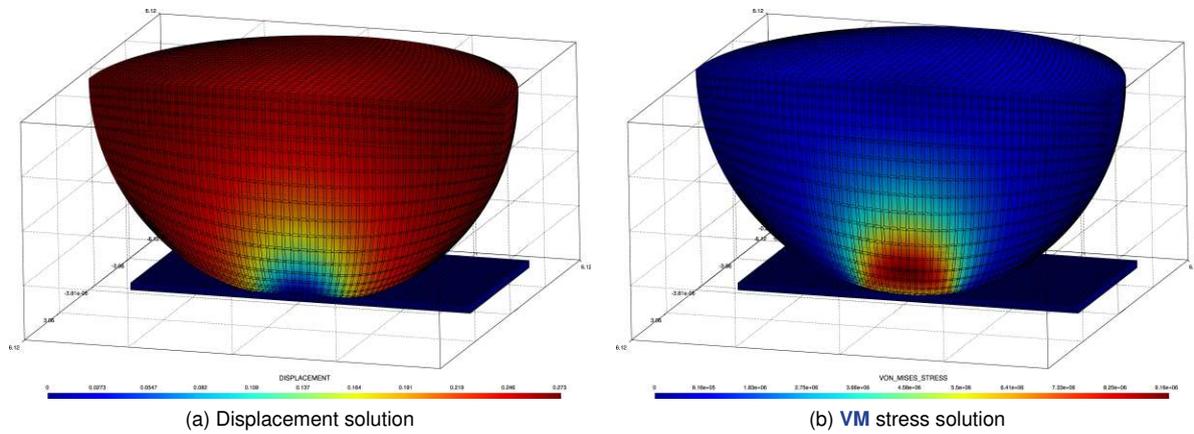


Figure 4.55: Mesh considered for 3D Hertz plane-sphere

Body	E	ν
Sphere	$1 \times 10^8 \text{ Pa}$	0.29
Plane	$2.1 \times 10^{11} \text{ Pa}$	0.29

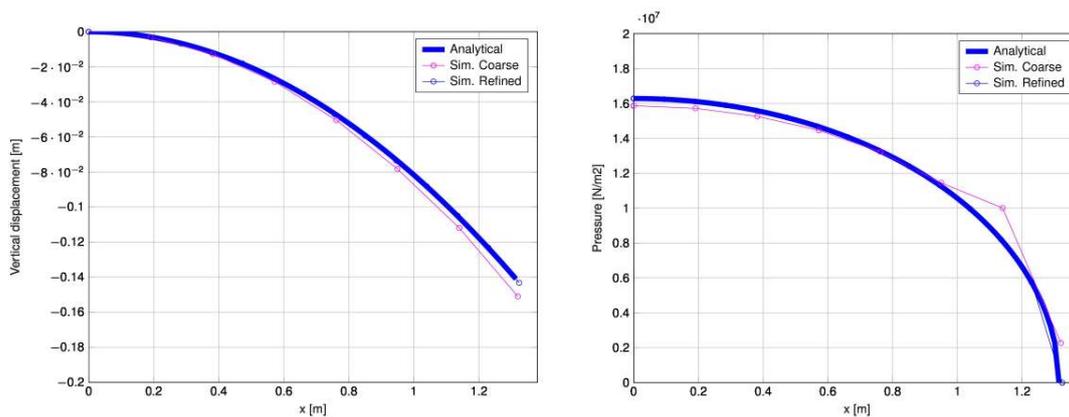
Table 4.9: Parameters considered for 3D *Hertz* plane-sphere problem



(a) Displacement solution

(b) VM stress solution

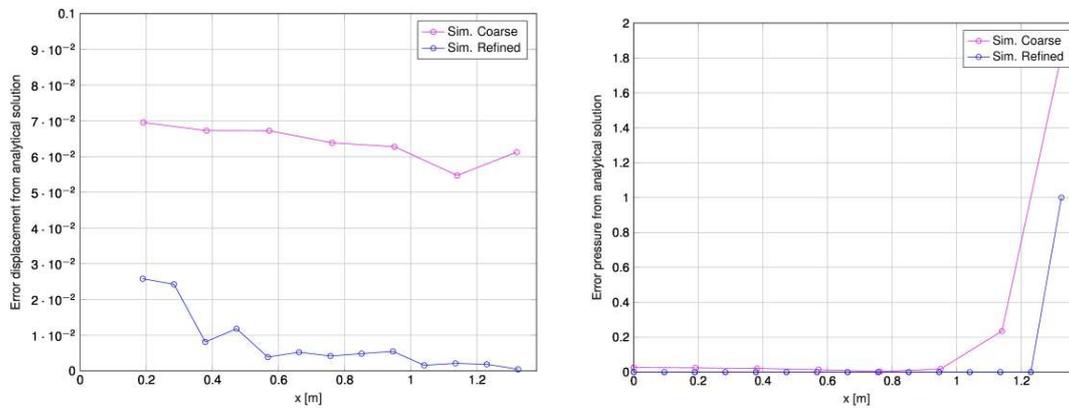
Figure 4.56: Solution for the 3D *Hertz* plane-sphere contact



(a) Displacement solution

(b) Stress σ_{yy} solution

Figure 4.57: Solution compared for different mesh sizes for the 3D *Hertz* plane-sphere contact



(a) Error in displacement solution

(b) Error in stress σ_{yy} solution

Figure 4.58: Error compared for different mesh sizes for the 3D *Hertz* plane-sphere contact

4.5.4.2.2 Sphere-sphere :

This problem, which comes from [here](#) consists in two contacting spheres as seen in the reference solution from Figure 4.59. We consider two hemispheres of 12.2474 meters of diameter, and considering an applied load of $q = 1.0e3Pa$. We can transform that given pressure to the corresponding force of the reference solution with $F = q \frac{\pi D^2}{4}$. The characteristic parameters of the spheres are the ones present in Table 4.10. The calculation is performed in just one static step.

As well as the cooperation with the reference solution, we will compare the radius of the contact area and the maximum contact pressure, corresponding with the analytical solution from (4.88). The obtained solution is compared with the reference one in (4.89), in here the maximal error is around a 1%. The displacement and **VM** stress field from Figure 4.60 shows the obtained solution.

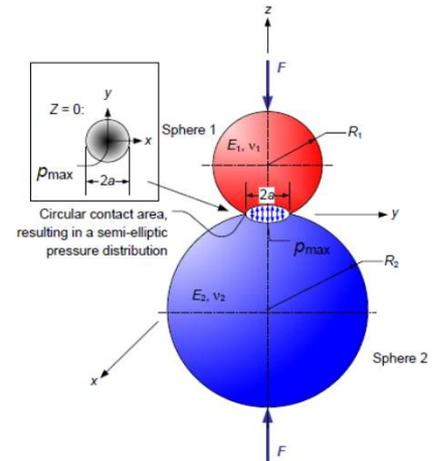


Figure 4.59: Reference solution for two spheres Hertz contact. Source[ArtZhu12]

(4.88a)
$$P_{max} = \frac{3F}{2\pi a^2}$$

(4.88b)
$$a = \sqrt[3]{\frac{3F \left[\frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} \right]}{4 \left(\frac{1}{R_1} + \frac{1}{R_2} \right)}}$$

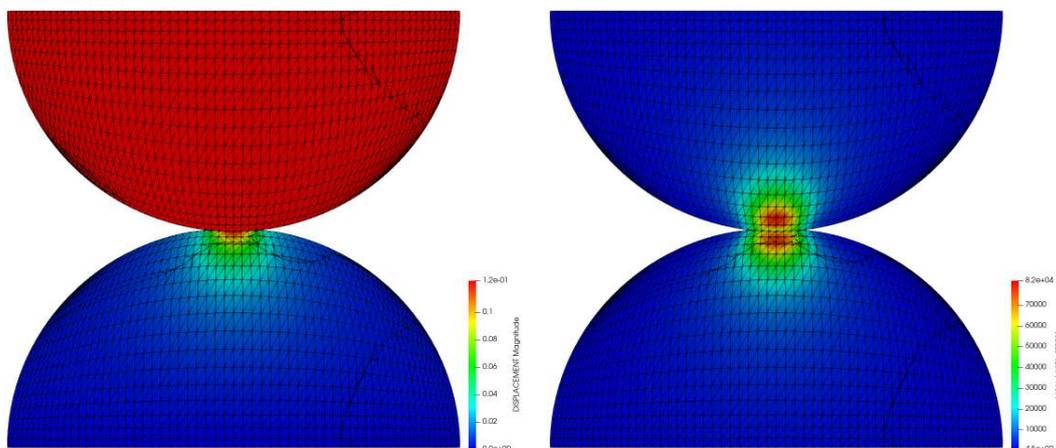
(4.89a)
$$F = 1.0e3 \cdot \pi \cdot \frac{12.2474^2}{4} = 150000/4 \cdot \pi = 117808.787N$$

(4.89b)
$$a = 0.6301m \text{ vs } 0.627m \rightarrow \underline{0.5\% \text{ error}}$$

(4.89c)
$$P_{max} = 1.41641 \times 10^5 Pa \text{ vs } 1.435467 \times 10^5 Pa \rightarrow \underline{1.3\% \text{ error}}$$

Body	E	ν
Upper body	$1 \times 10^8 Pa$	0.29
Lower body	$1 \times 10^6 Pa$	0.29

Table 4.10: Parameters considered for 3D Hertz sphere-sphere problem



(a) Displacement solution (b) **VM** stress solution

Figure 4.60: Solution for two hemispheres Hertz contact

4.5.5 Teeth model

The problem consists in a simplified model of a tooth with different types of layers, as the following: the first model with enamel-composite (Figure 4.61a) and the second one formed of enamel-dentine-composite (Figure 4.61b). The idea underlying this study is to compare the solutions between the two models and determine the benefits of the additional layer of dentine as a reinforcement of the composed structure. The properties of each one of the layers of the model appear on Table 4.11. It can be found [here](#).

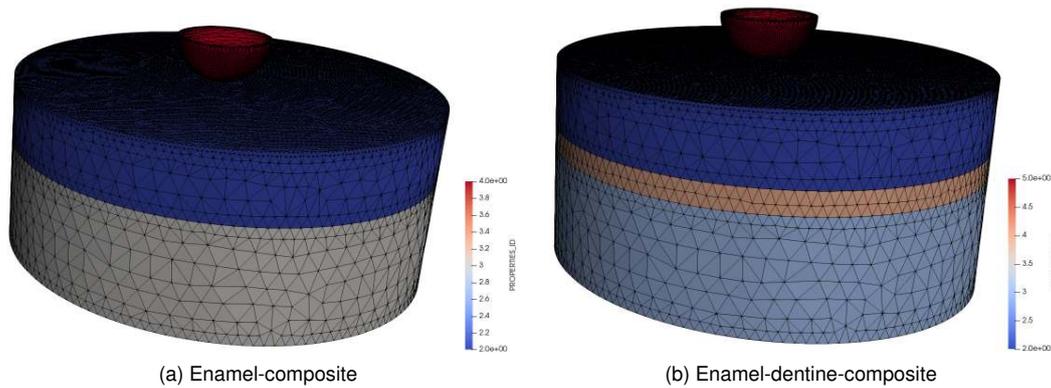


Figure 4.61: Teeth layers model

Body	E	ν
<i>Punch</i>	$2.069 \times 10^{11} Pa$	0.29
<i>Enamel (color 3)</i>	$8 \times 10^{10} Pa$	0.3
<i>Dentine (color 4)</i>	$2 \times 10^{10} Pa$	0.3
<i>Composite (color 2)</i>	$1.03 \times 10^{10} Pa$	0.3

Table 4.11: Parameters considered for teeth model

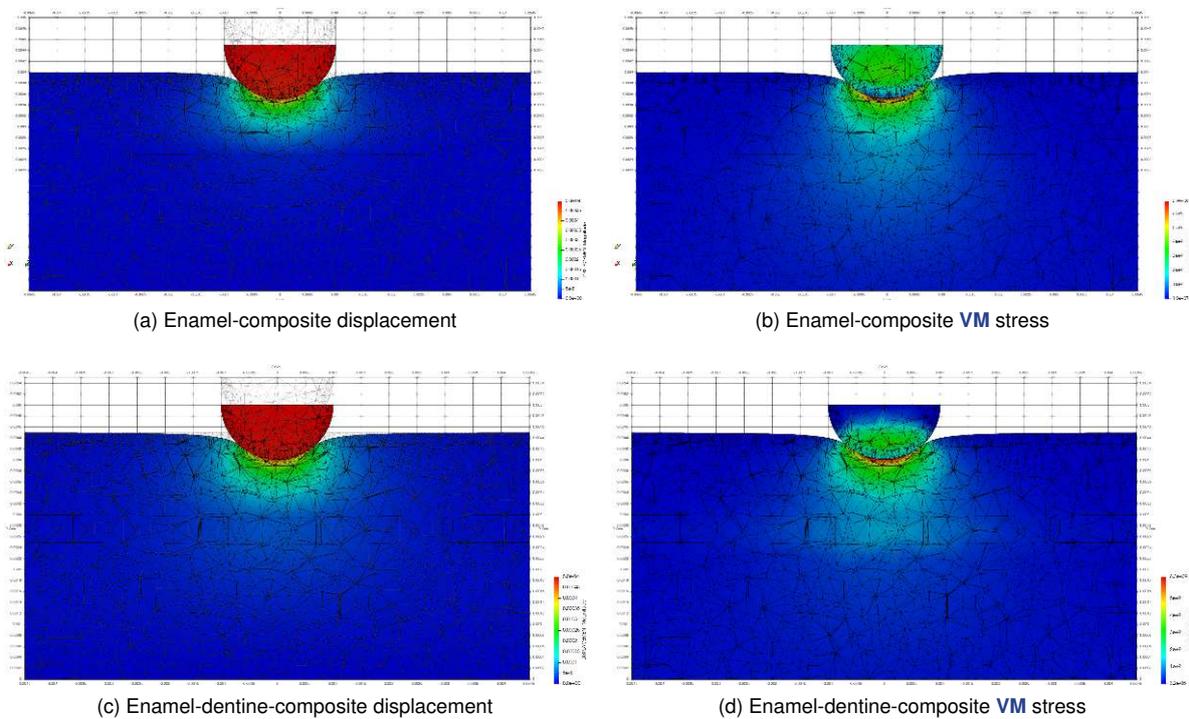


Figure 4.62: Solution for teeth layers model

Figure 4.62 the solutions compared of both alternatives allows us to see the advantages of the additional layer of dentine to the model.

4.5.6 Energy conservation

The following is an example to show the energy conservation in a frictionless contact simulation. In order to do so we will present the geometry shown in Figure 4.63. In this pictures what we see is a ring on $2m$ outer diameter and an inner diameter of $1.8m$. Inside the ring we have a cylinder of $0.4m$. The problem is a fully 3D simulation with $0.1m$ thickness in the Z axis. In this problem the cylinder is only subjected to its own weight. The inner cylinder is softer than the outer ring, which we can consider rigid from a practical point of view, see Table 4.12.

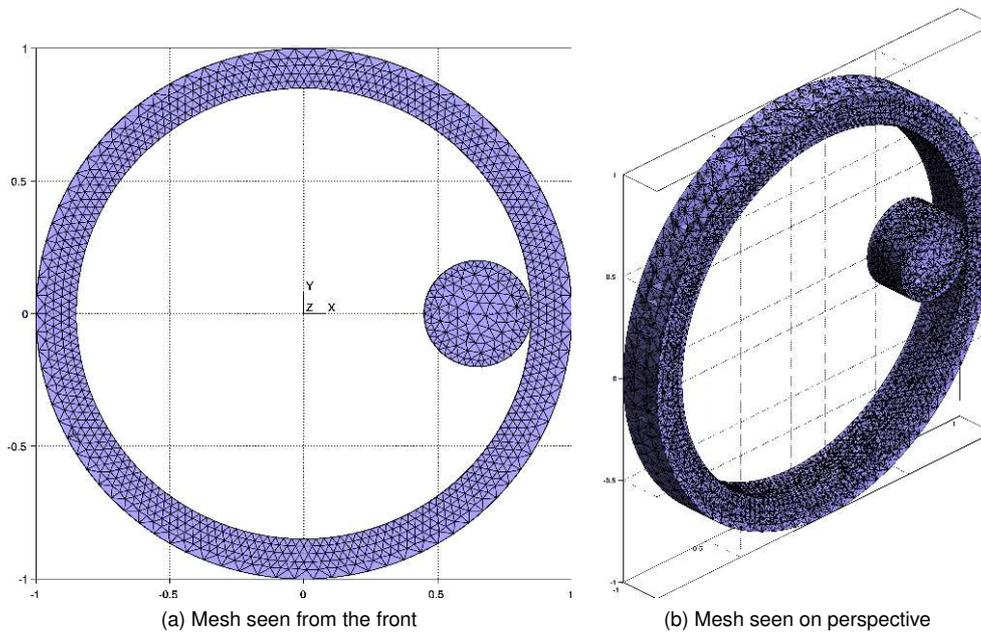


Figure 4.63: Energy conservation test. Cylinder inside ring. The example can be found in [here](#)

Body	E	ν	ρ
Ring	$2.069 \times 10^{11} Pa$	0.29	7850
Cylinder	$2 \times 10^8 Pa$	0.29	1000

Table 4.12: Parameters considered for energy conservation test

With the given information, we can predict the exact movement of the body the total energy and the velocity in any given time. For that we formulate the total energy as the total between the *kinematic* and the *potential* energy from Equation (4.90a). In here h is the height of the cylinder in a given time, m the total mass of the cylinder, g the gravity and v its velocity. If we define as only **DOF** the angle between the cylinder and the centre, being $\theta = 0$ then $h = 0.7m$ and $v = 0m/s$ as initial values. The total mass is as easy to calculate as $m = 0.2^2 \times \pi \times 0.1 \times 1000 = 4\pi kg$. The total energy is then, taking into account the initial **BC** previously mentioned then (4.90b). With this we can easily estimate the maximum velocity as Equation (4.90c).

$$(4.90a) \quad E_{tot} = E_{kin} + E_{pot} = hmg + \frac{1}{2}mv^2$$

$$(4.90b) \quad E_{tot} = 0.7 \times 4\pi \times 9.81 = 86.2632J$$

$$(4.90c) \quad v_{max} = \sqrt{\frac{2E_{tot}}{m}} = 3.7m/s$$

Then taking as **DOF** θ we must express then the velocity in terms of $\dot{\theta}$, considering the radius of the given trajectory as $R = 0.7$ then $v = R\dot{\theta}$. h can be represented with $h = R(1 - \sin(\theta))$. With this we can define the equation which defines analytically our movement as (4.91a). Replacing the operations properly, we obtain the following expression

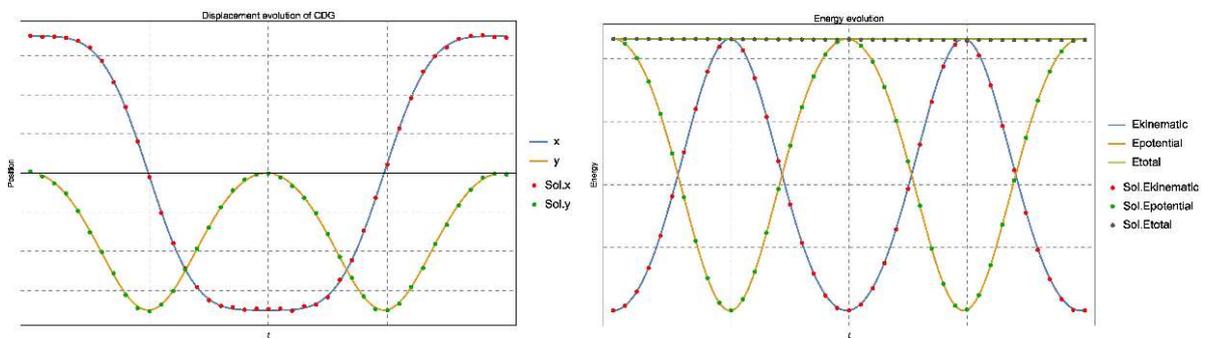
(4.91b). Solving the corresponding **ODE** the expression for θ results on Equation (4.91c).

$$(4.91a) \quad E_{tot} = R(1 - \sin(\theta))mg + \frac{1}{2}m(R\dot{\theta})^2$$

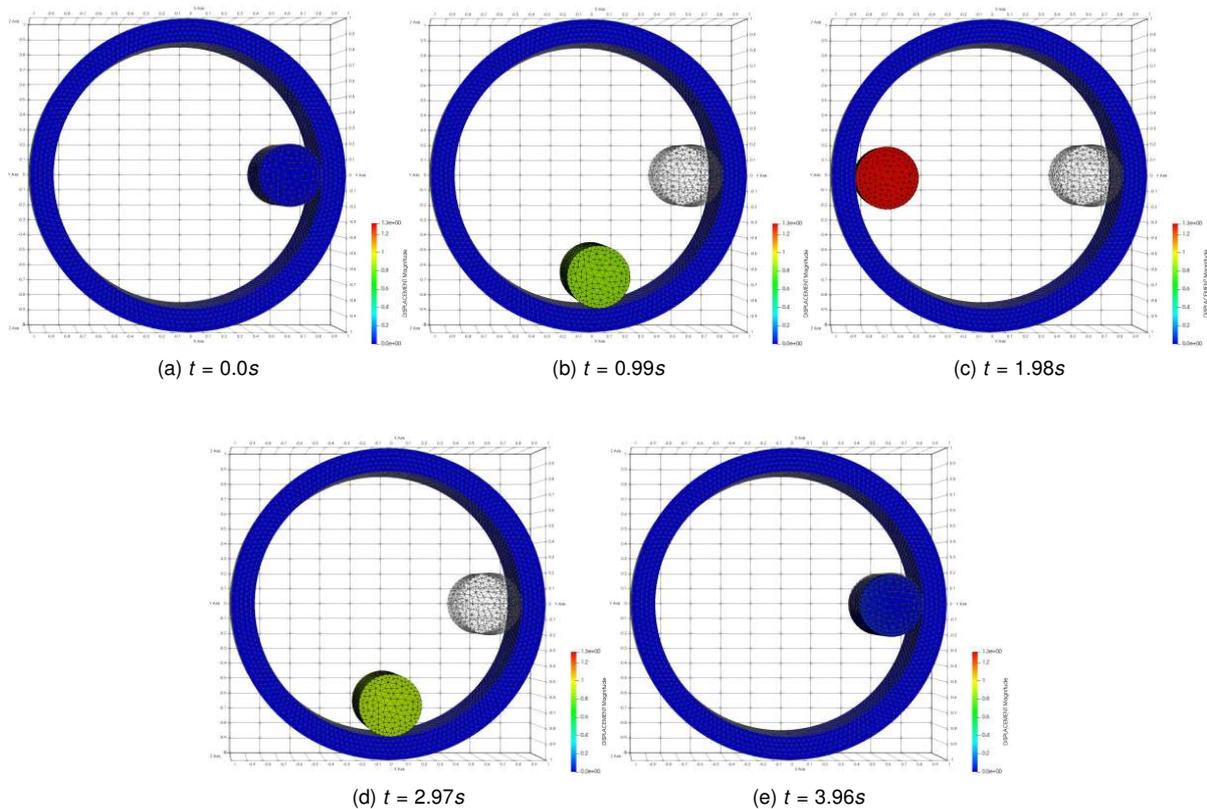
$$(4.91b) \quad \dot{\theta} = \sqrt{\frac{2g}{R} \sin(\theta)}$$

$$(4.91c) \quad \theta = \frac{1}{2} \left(\pi - 4 \operatorname{am} \left(\frac{1}{4} \left(-c_1 \sqrt{\frac{2g}{R}} + t \left(-\sqrt{\frac{2g}{R}} \right) \right) \middle| 2 \right) \right)$$

Where am is *Jacobi* elliptical function amplitude[OnJac]. The constant c_1 can be obtained replacing the **BC** previously stated, giving us as result $c_1 = -0.990539$. On Figure 4.64 we can see the comparison on the solution obtained with the analytical reference. The error on the solution obtained is less than a 1%.



(a) Displacement evolution (b) Energy evolution
Figure 4.64: Evolution of displacement and energy compared with analytical solution



(a) $t = 0.0s$ (b) $t = 0.99s$ (c) $t = 1.98s$
(d) $t = 2.97s$ (e) $t = 3.96s$
Figure 4.65: Solution of the problem at certain times. The time is specified in each subfigure

The Figure 4.65 shows the energy conservation in our simulation, and we can see as at the predicted time the cylinder returns into its original position, then the loop starts over. The obtained maximum velocity coincides with the predicted in (4.91c). Something that is relevant to remark is the fact as expected in a frictionless case, and therefore there is no rotation movement, the contact point between the cylinder and the ring is always the same. Besides, the energy is preserved despite the numerical dissipation due to *Bossak scheme* (see 2.4.4.2. *Bossak algorithm*).

4.5.7 Double arc benchmark

In this test case, we will consider the crushing of a hyperelastic bi-material half-ring on a hyperelastic base. This test makes it possible to take into account large displacements, large deformations and large landslides coupled to contact / non-contact transitions. The reference solutions have been taken from *G. Drouet*[PhDDro15] and *K.Poulios and Y.Renard*[ArtPR15a]. The problem geometry as well as the boundary conditions are sketched below on Figure 4.66, both 2D and 3D. It can be found at the Examples [repository](#).

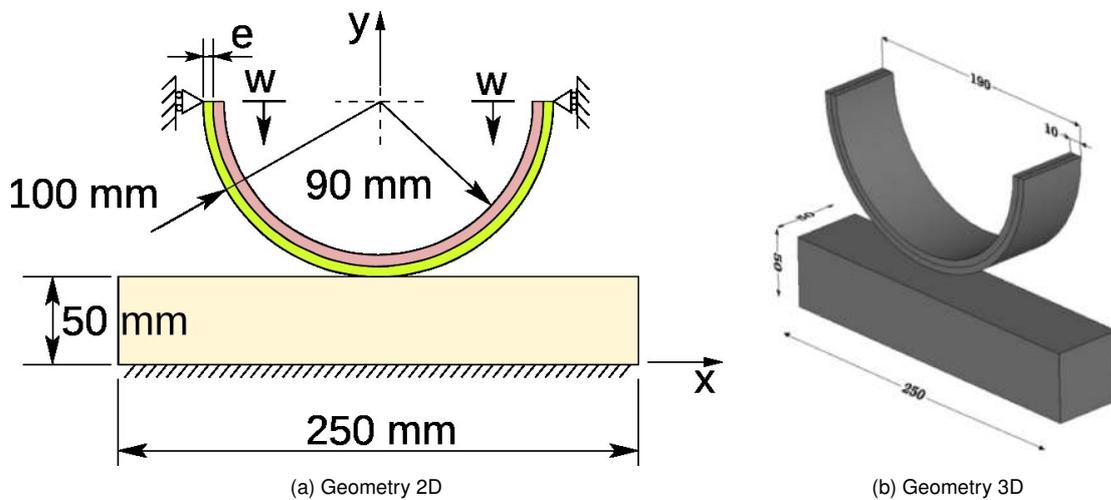


Figure 4.66: Double arc benchmark

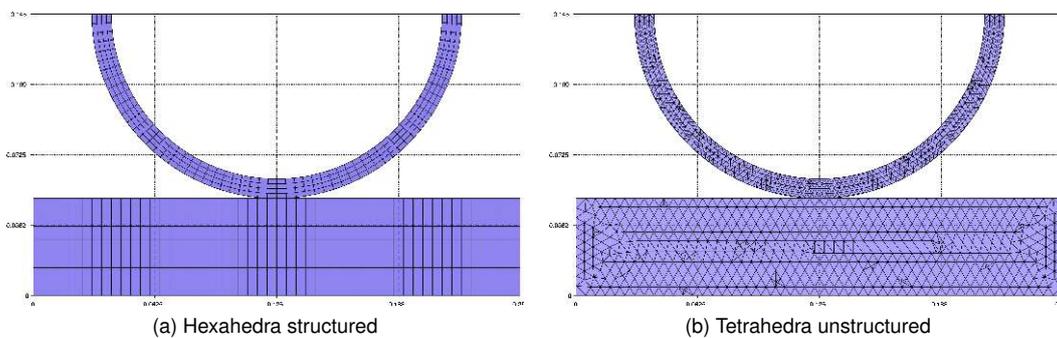


Figure 4.67: Double arc meshes

Body	E	ν
First arc	$3 \times 10^8 Pa$	0.32
Second arc	$1 \times 10^9 Pa$	0.32
Support material	$1 \times 10^{11} Pa$	0.3

Table 4.13: Parameters considered for double arc benchmark

We consider a hyperelastic half-ring composed of two materials (external 190mm and inner 170mm diameters) and a hyperelastic base of length 250mm and height 50mm, see Figure. A vertical displacement of $-90mm$ is imposed at each end of the half-ring and the base of the base is recessed. Two variables of interest are considered mainly: the displacement of the middle of the half-ring as a function of the loading pitch (60 steps of loading from the contact is used, ie about 1.16mm per loading step) and the contact pressure. On the other hand, an hyperelastic *Neo-Hookean* constitutive law is considered in both materials. The structure characteristic parameters are defined in

Table 4.13. The time step is 0.0005seconds , while the total simulation time is 0.2seconds . With an increment of the imposed displacement equal to $0.4t$. For the frictional case, the friction coefficient considered is $\mu = 0.5$.

4.5.7.1 Frictionless

The problem stated above has been solved using an unstructured mesh of tetrahedra and a structured mesh of hexahedron. The resulting deformation for the former case can be seen in the Figure 4.68.

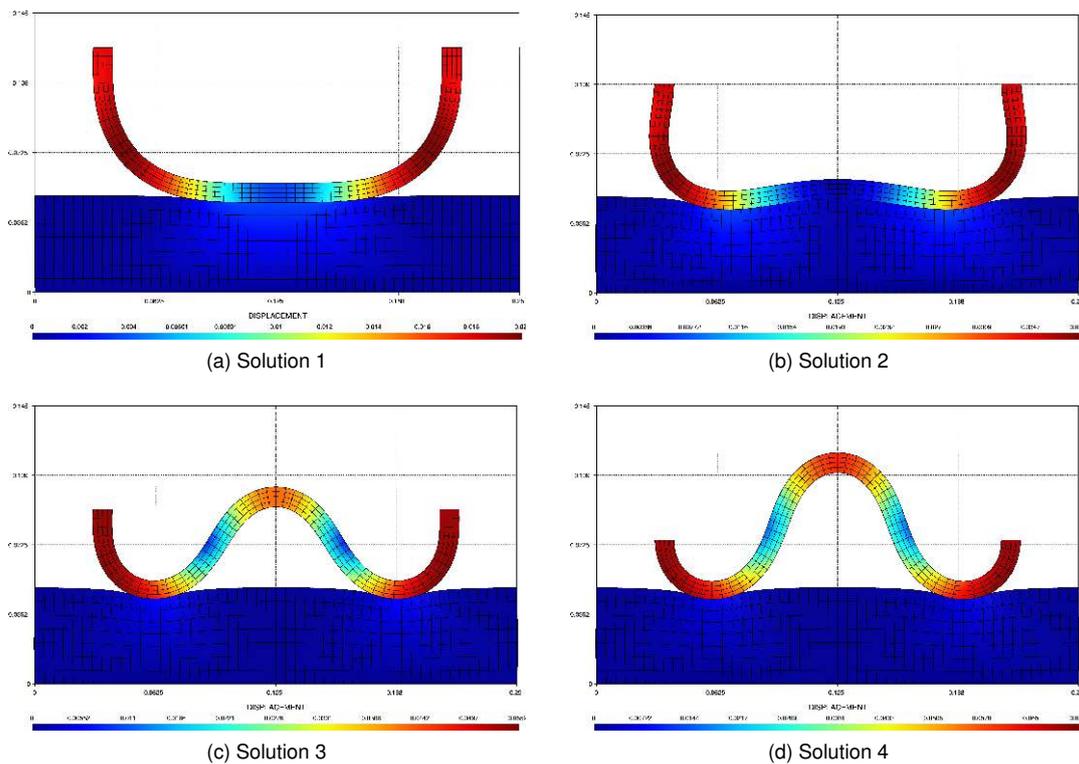


Figure 4.68: Displacement solution for frictionless in double arc benchmark

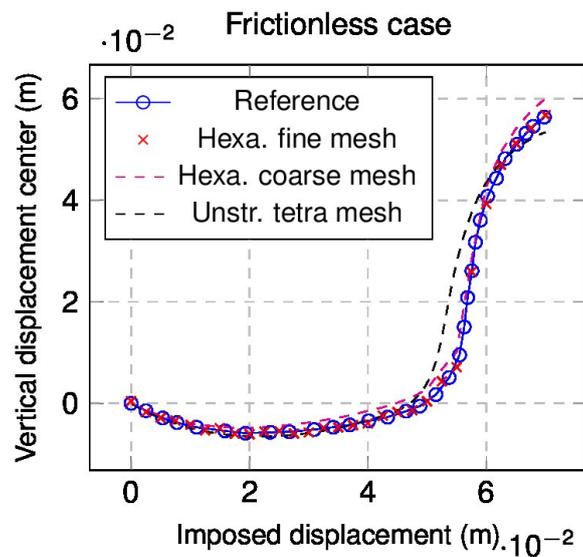


Figure 4.69: Compared solution for frictionless in double arc benchmark

We compare with the reference solution, Figure 4.69, where we have for the hexahedral meshes a deviation from the reference solution for the last steps, where we have a larger deformation and finally a stabilisation of the deformation. For the unstructured meshes of the tetrahedron, the results present more differences, respect the reference solution, higher deformation in general except for the last stages of the problem. In any case the results obtained are in very good agreement with the reference solution.

4.5.7.2 Frictional

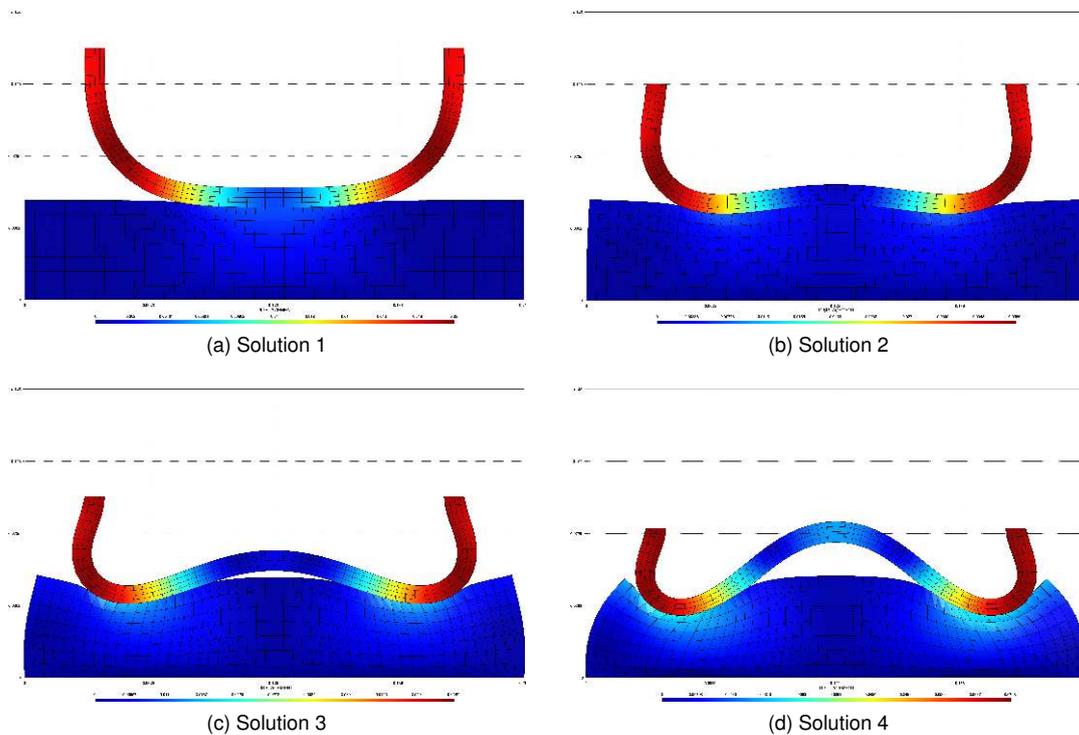


Figure 4.70: Displacement solution for frictional in double arc benchmark

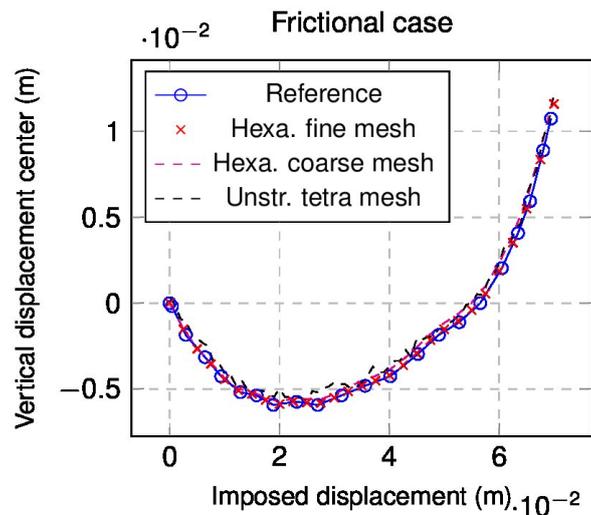


Figure 4.71: Compared solution for frictional in double arc benchmark

The solution for the frictional case can be seen in the following images, Figure 4.70, and the compared solution with the literature references (Figure 4.71), where the differences respect the reference are aligned with the ones

already appreciated on the frictionless case. Besides, the effects of the friction can be noted in the way the arc deforms in comparison of the frictionless case.

4.5.8 Arc pressing block

The problem consists in an arc and a block, in here the arc is pressed against the block. These two deformable solids are considered as hyperelastic (*Neo-Hookean*), and the solid is formulated in **UL** framework. We consider three cases, depending of the relationship of stiffness between the arc and the block, the first one with a rigid block, the second one with a deformable block and the final one for deformable block with the arc being rigid. These properties are specified on Table 4.14. The **BC** consists on a displacement imposed is $u_y = t$, with $t = [0.0, 1.0]$ for the two first cases, $t = [0.0, 1.775]$ for the rigid arc. The example can be found [here](#).

Body	Constitutive Law (CL)	E	ν
Arc (Rigid block)	Neo-Hookean	$68.96 \times 10^8 Pa$	0.32
Block (Rigid block)	Neo-Hookean	$68.96 \times 10^7 Pa$	0.32
Arc (Deformable block)	Neo-Hookean	$68.96 \times 10^8 Pa$	0.32
Block (Deformable block)	Neo-Hookean	$68.96 \times 10^5 Pa$	0.32
Arc (Deformable block-Rigid arc)	Neo-Hookean	$68.96 \times 10^9 Pa$	0.32
Block (Deformable block-Rigid arc)	Neo-Hookean	$68.96 \times 10^5 Pa$	0.32

Table 4.14: Parameters considered for arc pressing blocks

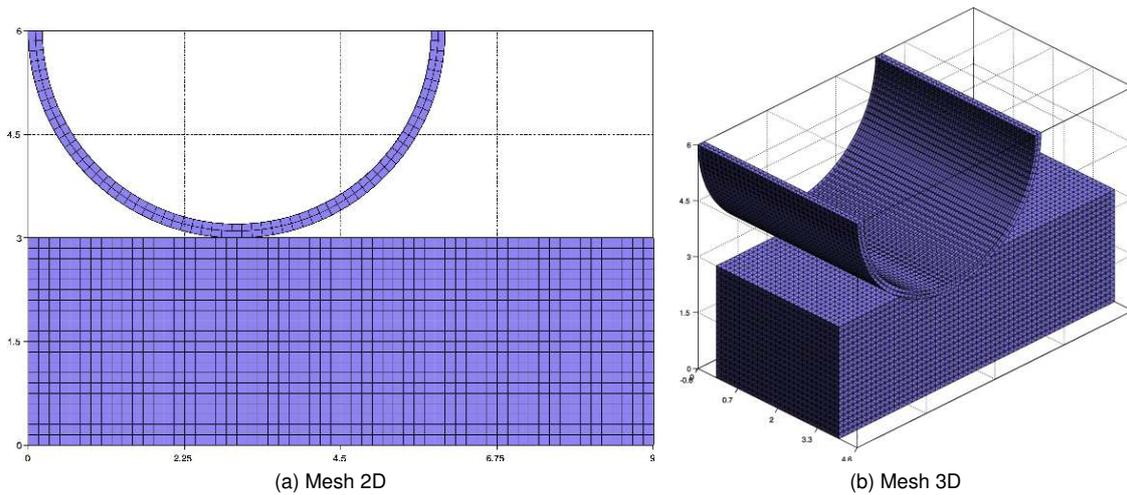


Figure 4.72: Arc pressing block

Figure 4.73 summarises the solution obtained for each one of the material properties previously presented. It can be seen the effects of the relative stiffness in each one of the cases. Particularly in the case where the block is defined and the arc is rigid (Figure 4.73b), and the case where the block is rigid (Figure 4.73c) and all the deformation lies in the arc.

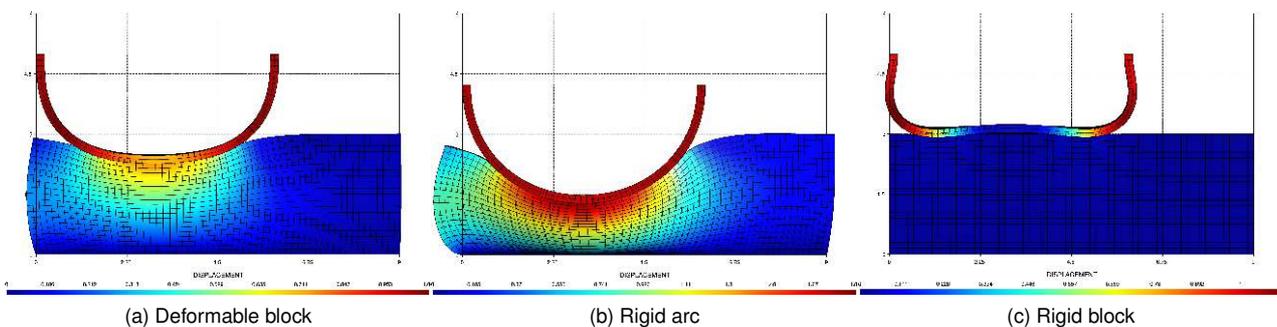


Figure 4.73: Solution for different stiffness between the arc and the block

4.5.9 Hyperelastic tubes

The problem[ArtPR15a] consists in two cylinders, Figure 4.74, with hyperelastic behaviour, as shown in the Table 4.15. We impose a vertical displacement $u_z = -0.01t$ in the upper cylinder, during a time interval of $t = [0.0, 4]$, in 100 steps. The two cylinders deal with finite deformations and are defined in a **UL** framework.

The two cylinders become into contact and deal with self-contact in the inner ring of the upper cylinder. For further details about this example, the following [link](#) expands and presents the necessary files.

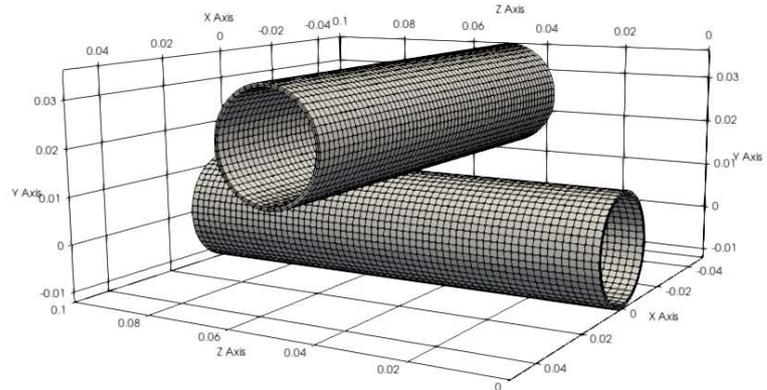


Figure 4.74: Mesh for hyperelastic tubes

Body	CL	E	ν
Upper cylinder	Neo-Hookean	10000Pa	0.3
Lower cylinder	Neo-Hookean	100000Pa	0.3

Table 4.15: Parameters considered for hyperelastic tubes

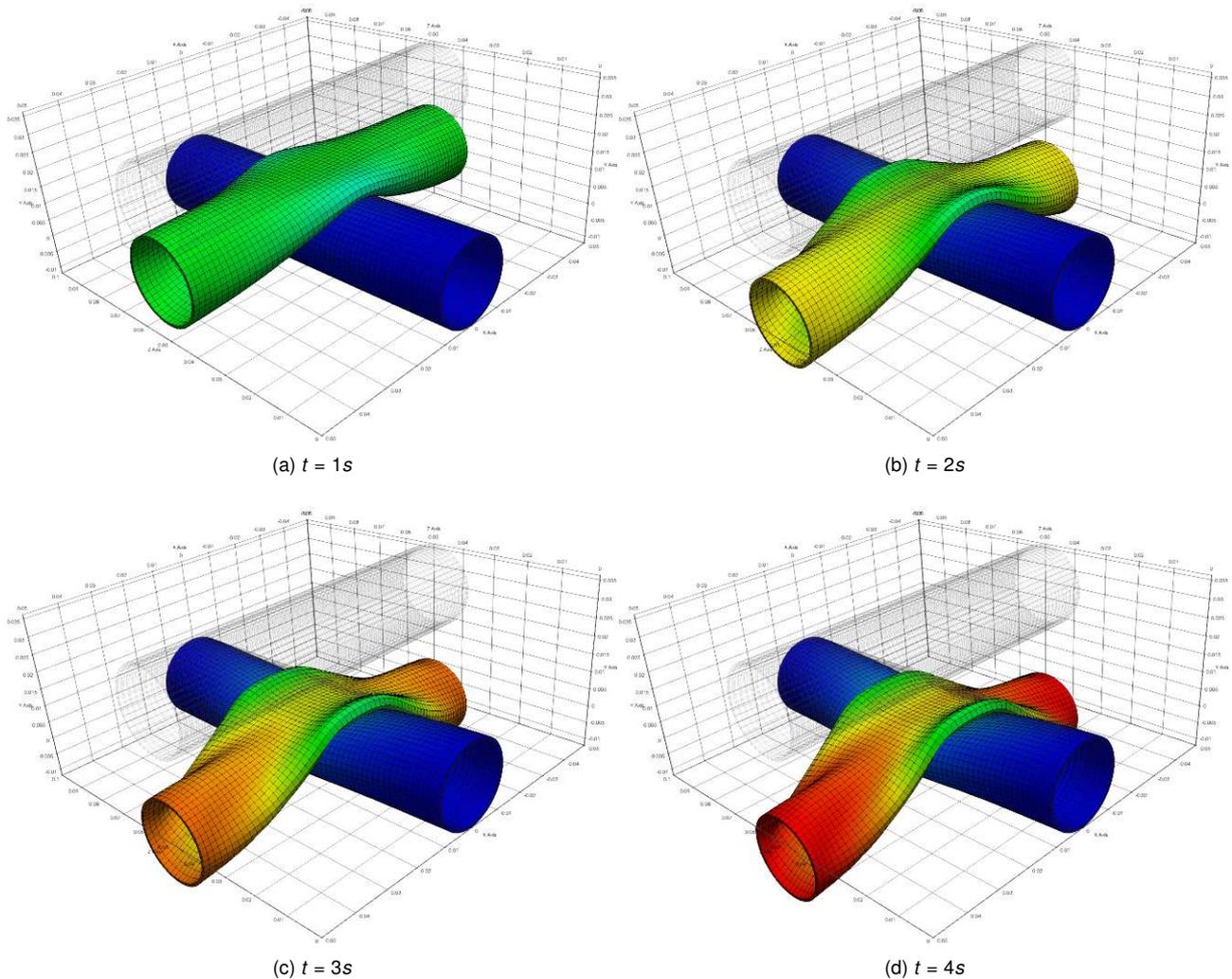
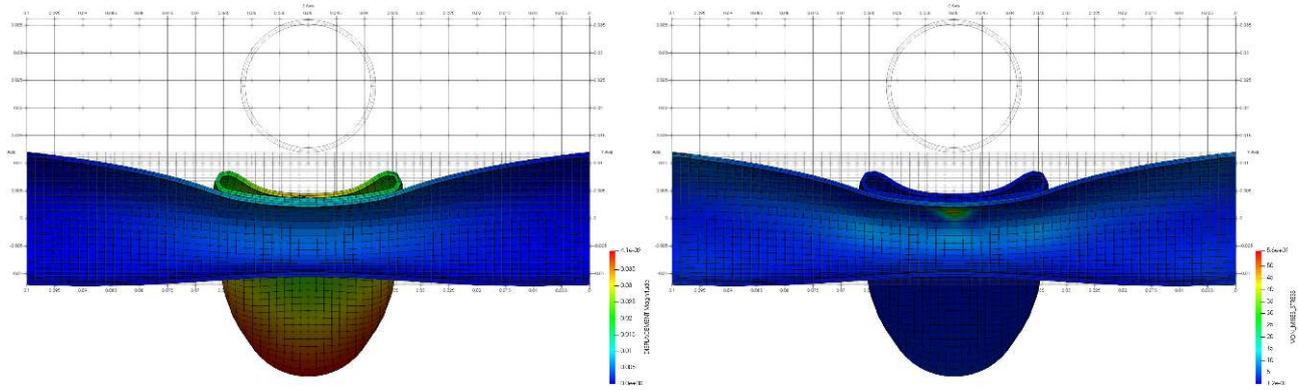


Figure 4.75: Displacement solution for hyperelastic tubes

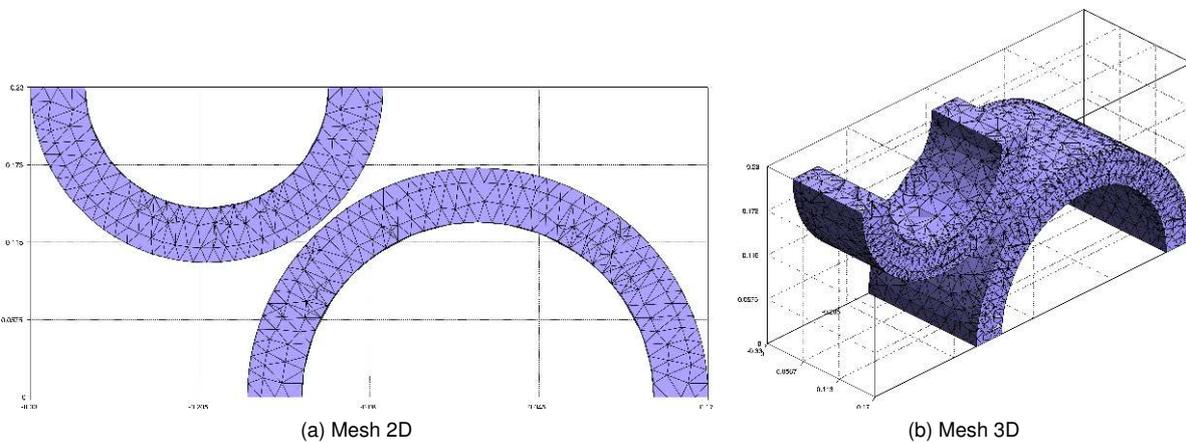
The Figure 4.75 presents the solution across different time steps, $t = [0, 4]s$. Additionally, Figure 4.76 shows a split of the final configuration, $t = 4s$, so the deformation at the end of the simulation can be appreciated at Figure 4.76a and the **VM** stress at Figure 4.76b. In this configuration the phenomenon of self-contact arises.



(a) Displacement (b) **VM** stress
Figure 4.76: Slice solution for hyperelastic tubes

4.5.10 Contacting cylinders

The following case consists in two deformable hemicylinders (Figure 4.77) which experiment large deformations with *Neo-Hookean* hyperelastic behaviour (Table 4.16). We applied two different movements in the base of the upper cylinder, first a horizontal movement (4.5.10.1.Horizontal movement) and later a vertical movement (4.5.10.2.Vertical movement). The relative position of the hemicylinders changes slightly between the two configurations. The problem is studied again in the test cases from the adaptive refinement chapter, see 6.10.5.Contacting cylinders with adaptive remeshing. These problems can be found [here](#).



(a) Mesh 2D (b) Mesh 3D
Figure 4.77: Contacting cylinders mesh

Body	CL	E	ν
Upper cylinder	Neo-Hookean	$2.1 \times 10^{11} Pa$	0.29
Lower cylinder	Neo-Hookean	$2.1 \times 10^{11} Pa$	0.29

Table 4.16: Parameters considered for contacting cylinders

4.5.10.1 Horizontal movement

A horizontal displacement is imposed $u_x = 0.2t$, for a time interval of $t = [0, 2.5]s$ in 1000 steps. Figure 4.78 shows the displacement solution in different time steps, resulting in the reaction at the bottom of the lower hemicylinder from Figure 4.79. The problem has been solved both considering frictionless (Figures 4.78a, 4.78b, 4.78c) and frictional ($\mu = 1$) (Figures 4.78d, 4.78e, 4.78f)cases. The difference of behaviour between the two cases is notorious.

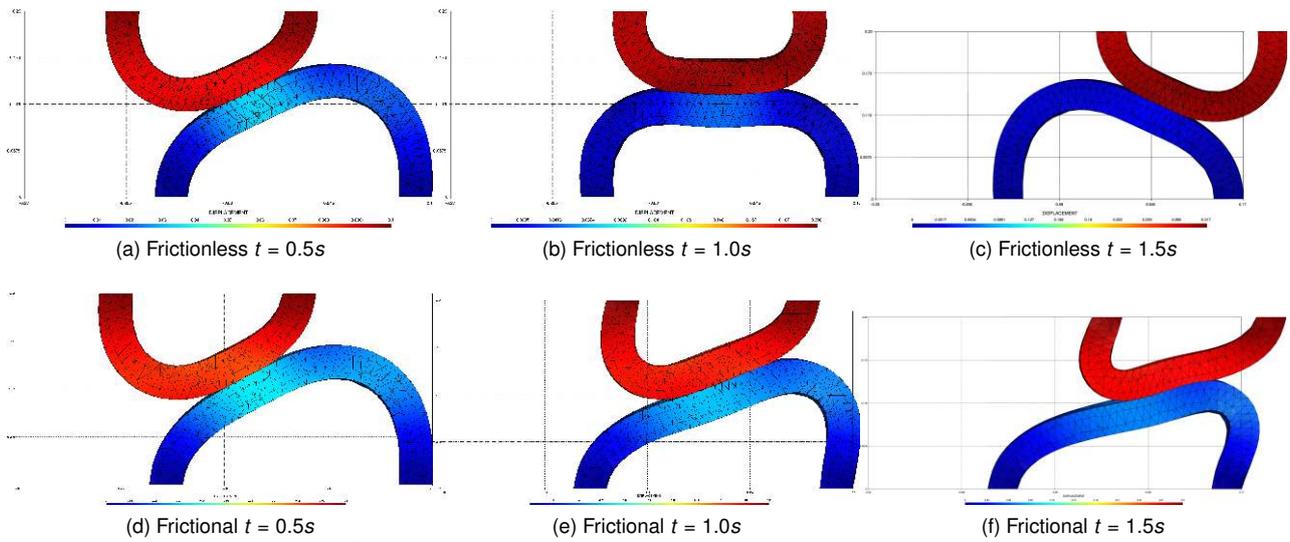


Figure 4.78: Solution for contacting cylinders with horizontal movement

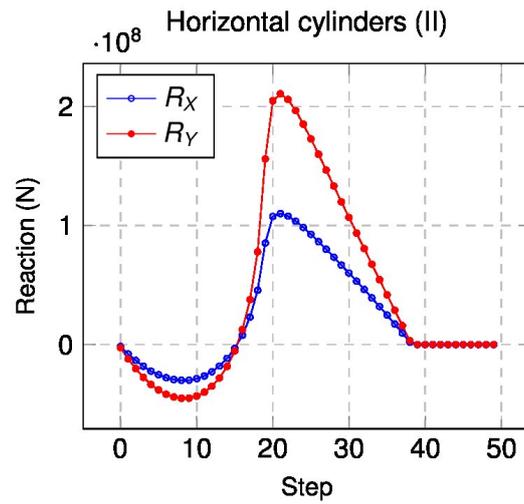


Figure 4.79: Frictionless reaction solution in the lower cylinder for the horizontal movement

4.5.10.2 Vertical movement

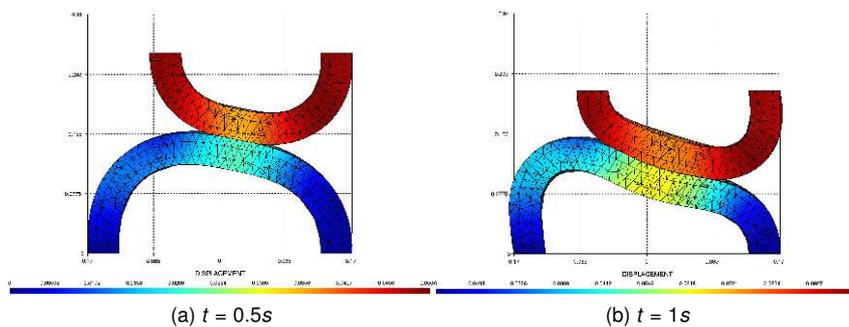


Figure 4.80: Solution for contacting cylinders with vertical movement

For the vertical displacement is imposed $u_y = 0.1t$, for a time interval of $t = [0, 1]s$ in 200 steps. At Figure 4.80 the deformation experimented is presented, as well as the reaction (Figure 4.81) in the base of the still hemicylinder.

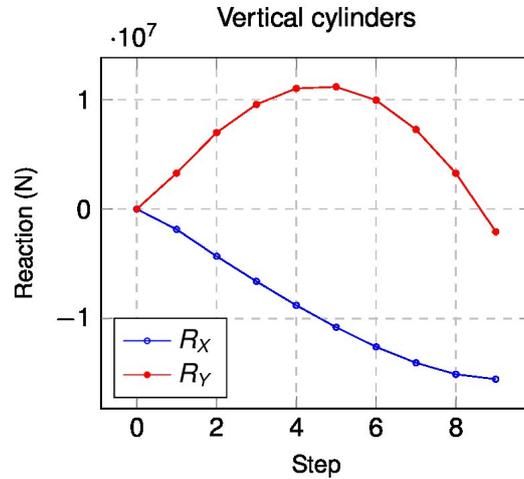


Figure 4.81: Reaction solution in the lower cylinder for the vertical movement

4.5.11 Press fit

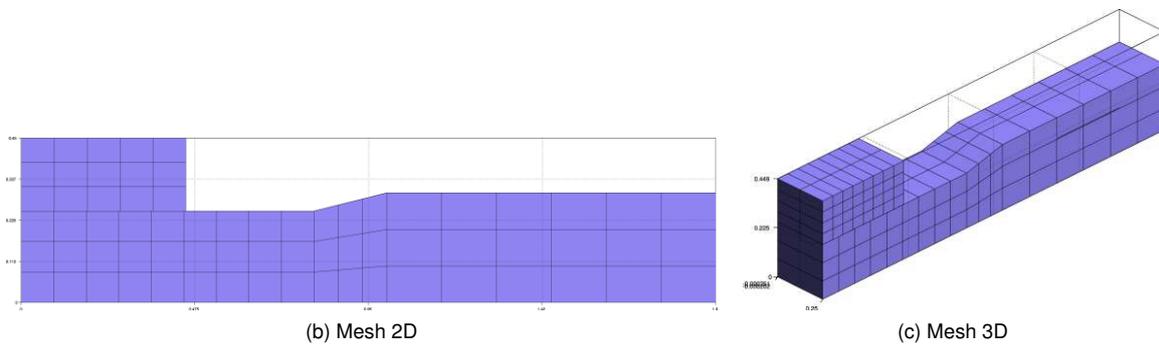
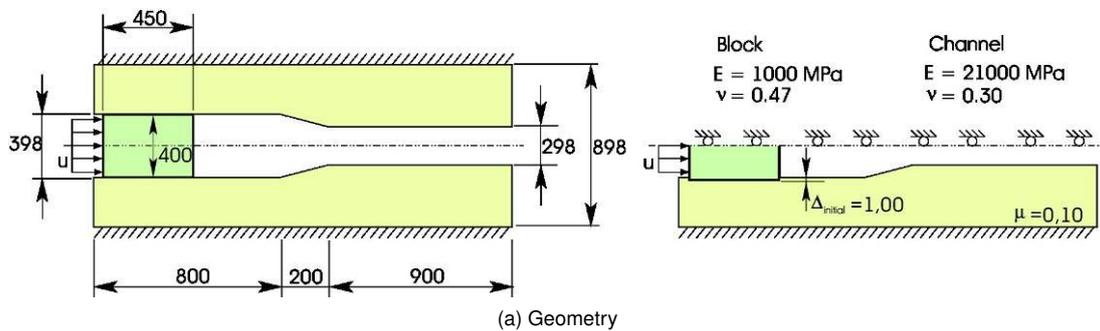


Figure 4.82: Press fit problem

The problem consists of the numerical simulation of a press-fit process of a block in a channel according to [ArtFW06; ArtDSB15]. The geometric and material parameters are shown in the Figures 4.82, where the symmetry of the problem is used and only half of the domain is simulated. It is considered *Neo-Hookean* compressible hyperelastic material for both components with the values from Table 4.17. With this problem, it is possible to evaluate the behaviour of the contact element in a friction contact problem with large deformation and sliding.

Body	E	ν	μ
Die	$68.96 \times 10^8 Pa$	0.32	0.1
Block	$68.96 \times 10^7 Pa$	0.32	0.1

Table 4.17: Parameters considered for press fit

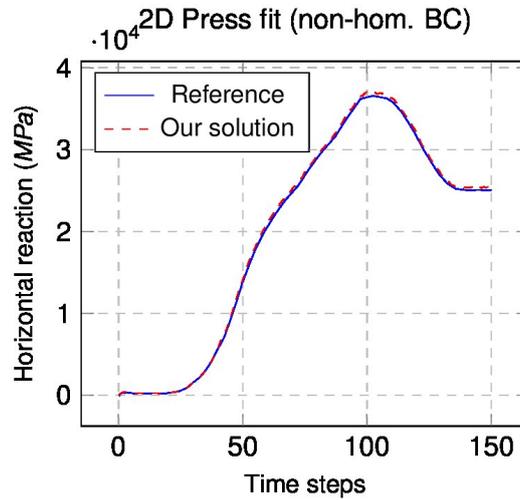


Figure 4.83: Solution for 2D press fit compared with reference

The process is modelled by applying a non-homogeneous boundary condition $u = 1000mm$ in the left face of the block. The height of the block is greater than the channel, imposing to the problem an initial penetration $\Delta_{initial} = 1mm$ and consequently an initial contact stress. The first time step uses $u = 0mm$ and the program generate the normal stress necessary to the non-penetration condition, separating the bodies in contact. After this step, we employed a non-homogeneous boundary condition $u = 1000mm$. It is considered plane strain state and **UL** formulation.

Additionally a 3D setup is also evaluated, extruding $250mm$, but this one cannot be compared with any reference. This can be found in together with the other case studies [shown](#). Comparing the results with the reference, we got slightly higher values than in the reference as seen in Figure 4.83. In the other hand, the evolution in the solution is presented on the Figures 4.84.

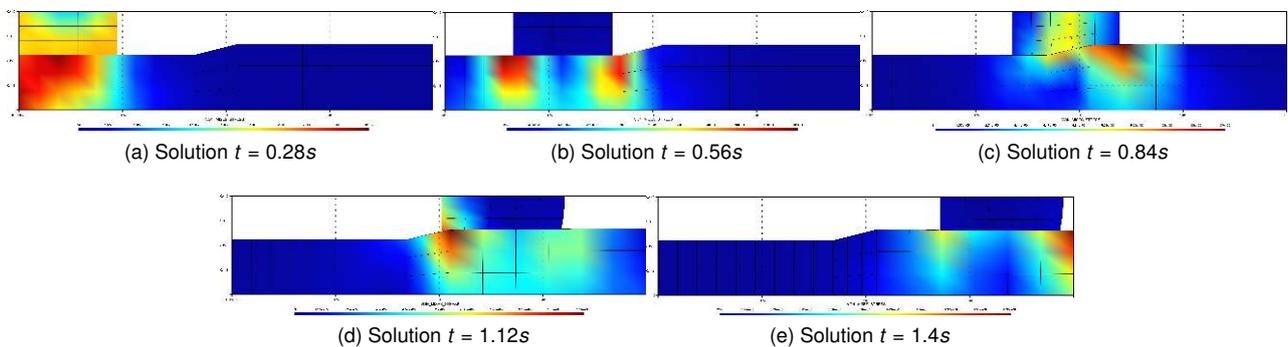
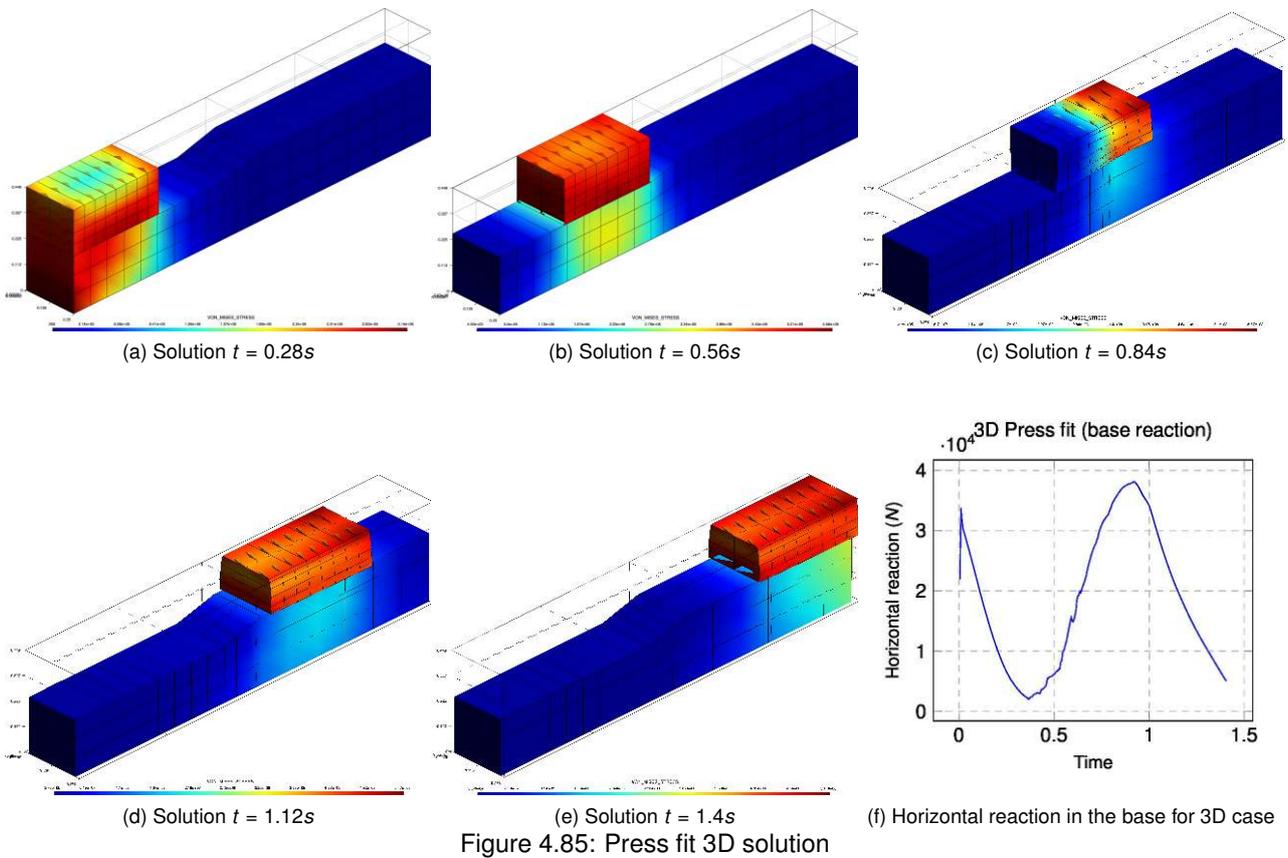


Figure 4.84: Press fit 2D solution

For the 3D case, where we do not have a reference solution to compare, therefore we can see the deformation evolution in the first five figures from Figure 4.85. The reaction in the base support most western points is shown in the last image of the set, the Figure 4.85f.



4.5.12 Ironing punch

The following are two examples extensively considered in the literature [ArtFW06; ArtHar+09; ArtPR15b], which are the ironing tests. These tests consist in moving die along a solid block, undergoing large deformation in the process.

1. This case cannot be found in literature, and it is a little bit more challenging as the curvature of the contacting die is higher than in the previous case, we will call this case as *circular ironing* (4.5.12.1.Circular ironing).
2. We will show the results obtained with the standard ironing test, called *shallow ironing*, which is the one commonly referred on the literature (4.5.12.2.Shallow ironing). In this second case, we will consider both a frictionless and a frictional case.

4.5.12.1 Circular ironing

In this case we will run the simulation just considering the frictionless case as our interest is to compare the behaviour between the two different geometries. The geometry of the problem is fully defined in the Figure 4.86a, and the hexahedral structured mesh considered in Figures 4.86b and 4.86c. The properties of the materials are identical for the shallow ironing problem, with a *Neo-Hookean* material behaviour (5.4.2.Neo-Hookean material). We impose a vertical displacement as $u_y = -t$ during the time interval $t = [0, 1]s$ and $u_x = t - 1$ at $t = [1, 10]s$ keeping the previous vertical displacement. The example is found in the repository.

Body	E	ν
Die	$68.96 \times 10^8 Pa$	0.32
Block	$68.96 \times 10^7 Pa$	0.32

Table 4.18: Parameters considered for circular ironing

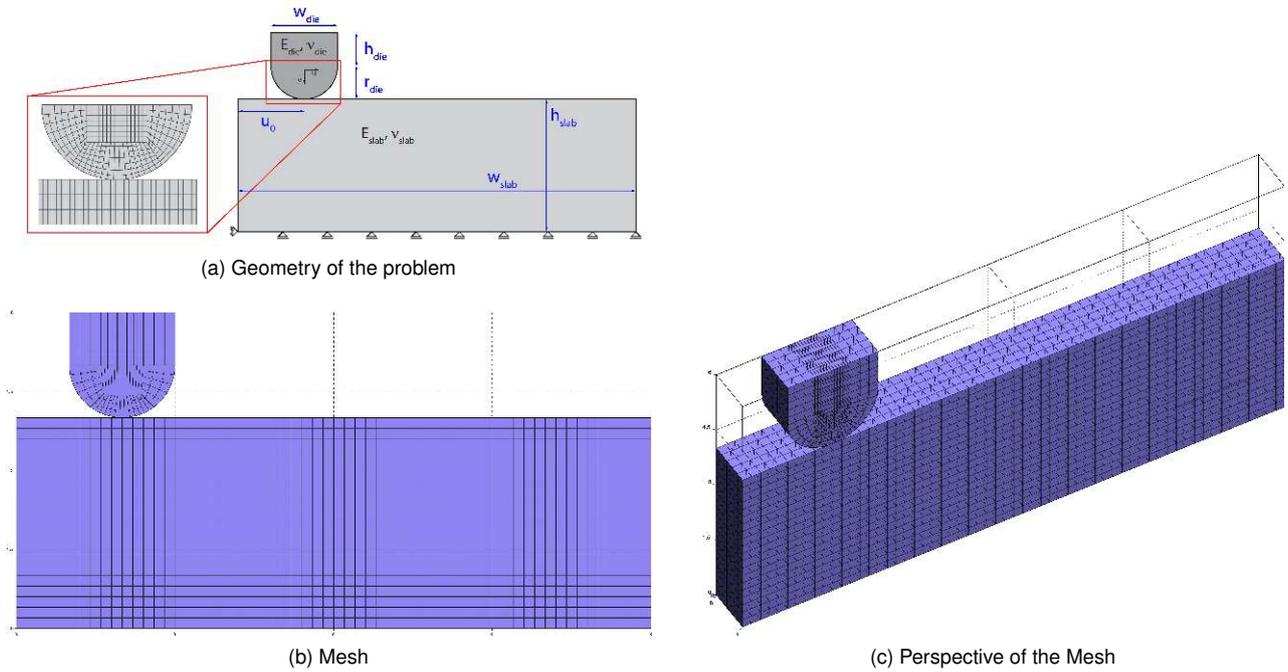


Figure 4.86: Circular ironing test

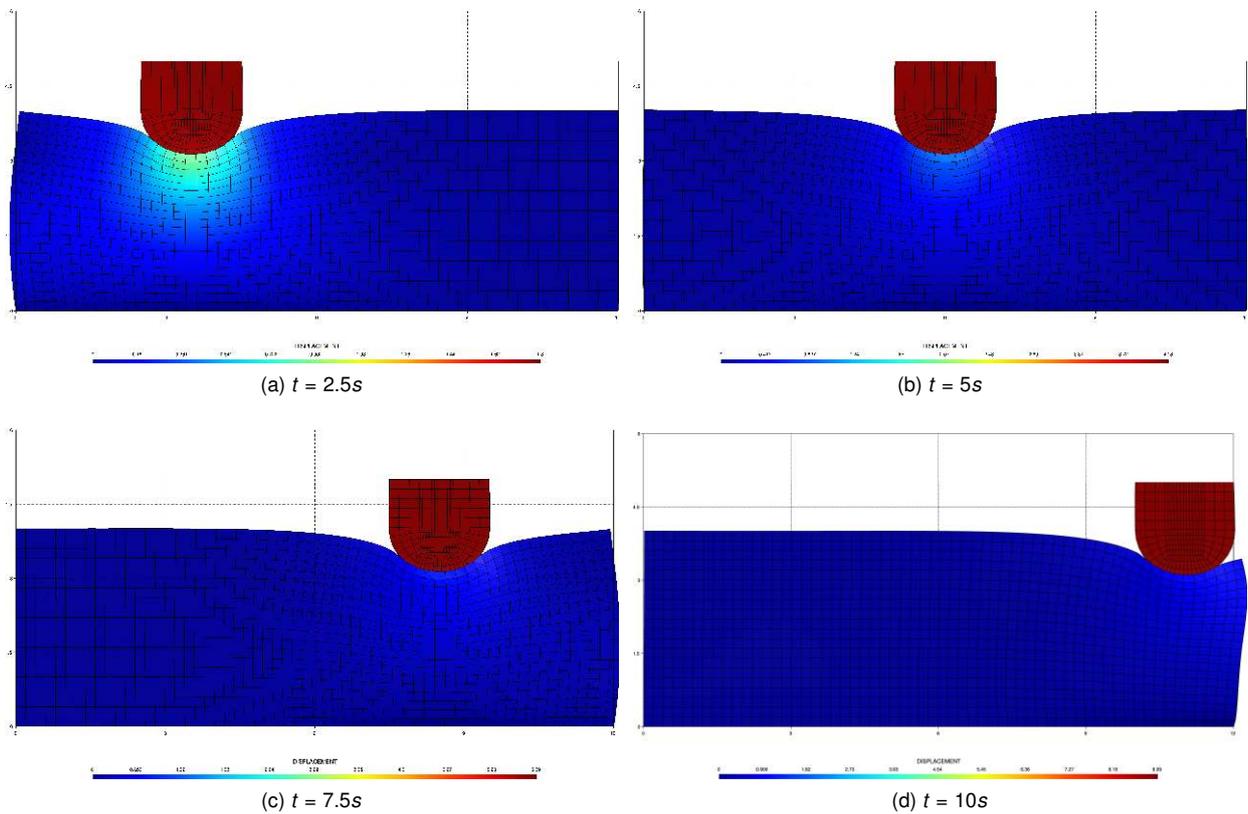


Figure 4.87: Circular ironing test solution

Figures 4.87 present the deformation evolution for this case in the interval going from $t = [0, 10]s$. The Figure 4.88 shows the different solution obtained with respect the shallow ironing example.

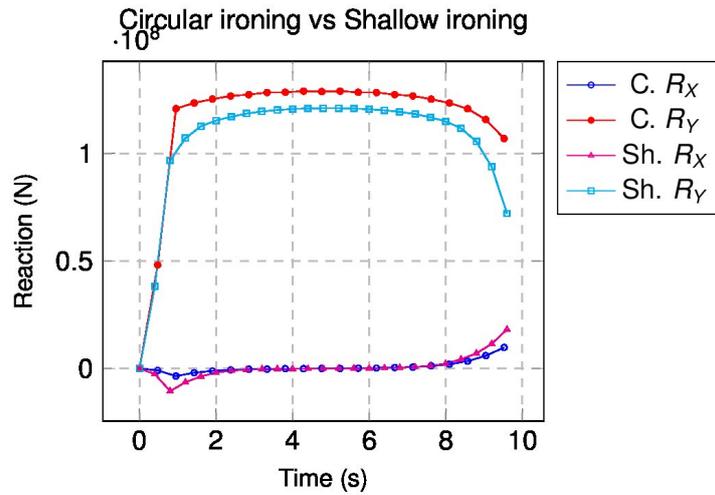


Figure 4.88: Comparison between the frictionless solutions of the shallow and the circular ironing

4.5.12.2 Shallow ironing

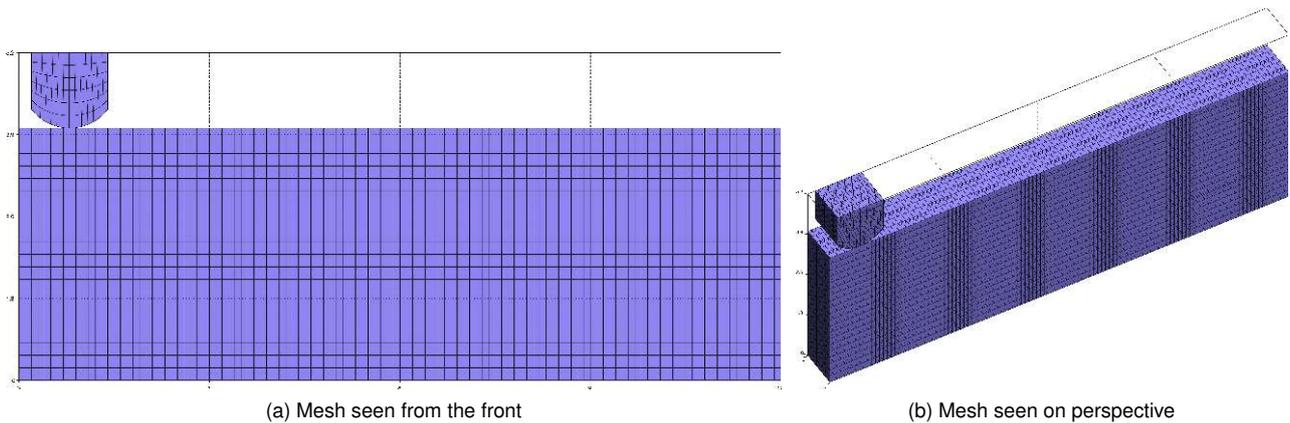


Figure 4.89: Shallow ironing test

On this case an indenter or die with a circular arc shaped bottom edge is pressed against a rectangular block and is forced to slide along the block length, see Figure 4.89 where the geometry and mesh considered can be appreciated. The contacting bodies exhibit a *Neo-Hookean* material behaviour (5.4.2. *Neo-Hookean material*), with the parameters from Table 4.19. The problem solved on this case is fully 3D with a deep of $1m$, but in many references can be found as a 2D case with plane strain hypothesis. The example can be found in [here](#).

Body	CL	E	ν	μ
Die	Neo-Hookean	$68.96 \times 10^8 Pa$	0.32	0.3
Block	Neo-Hookean	$68.96 \times 10^7 Pa$	0.32	0.3

Table 4.19: Parameters considered for shallow ironing

Although the performed simulation is *quasi-static*, the load steps are defined as a function of time for the sake of presentation of the results. From $t = [0, 1]s$, the indenter is moved vertically towards the block until the vertical displacement reaches $1m$. From $t = [1, 12]s$, the die is displaced horizontally by a total distance of $11m$. This loading differs sometimes in the literature, because of that the results obtained must be translated to the equivalent displacement evolution process.

The Figures 4.90 compare the solution to the shallow ironing test between the friction and the frictional solution at different times. We can appreciate that the frictional case indeed opposes the movement of the die. Figures 4.91

compares the solution obtained with the reference[ArtPR15b]. In general the agreement is good with [ArtPR15b], and then the solution can be considered as valid.

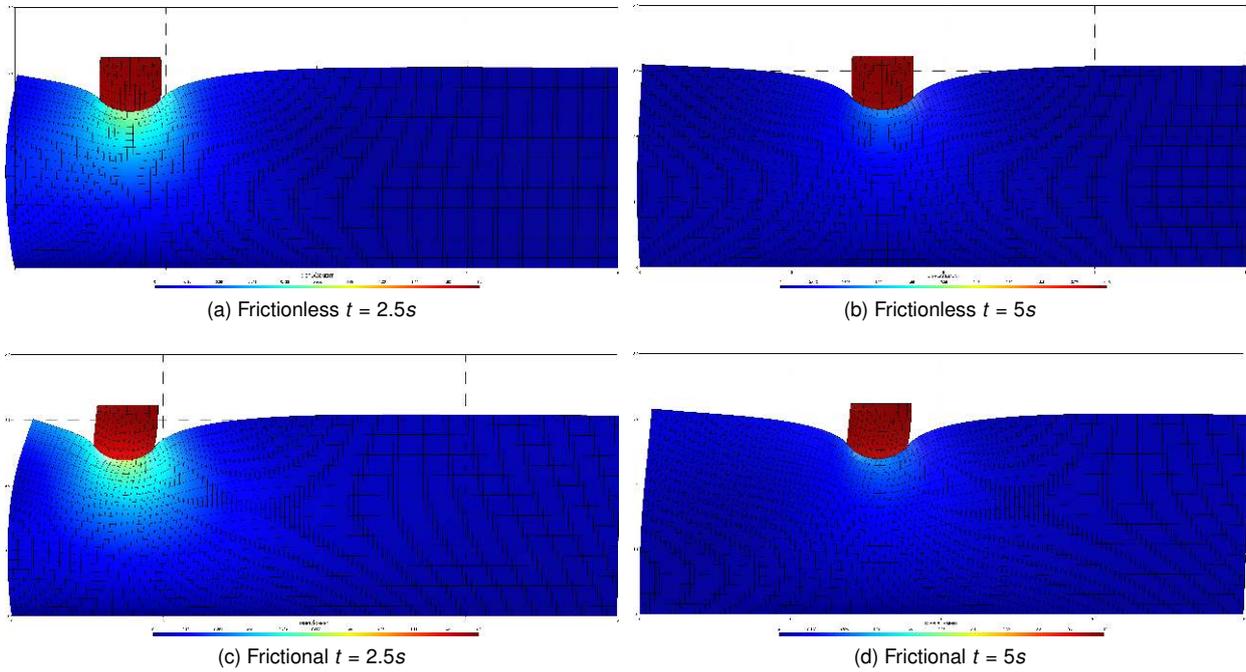
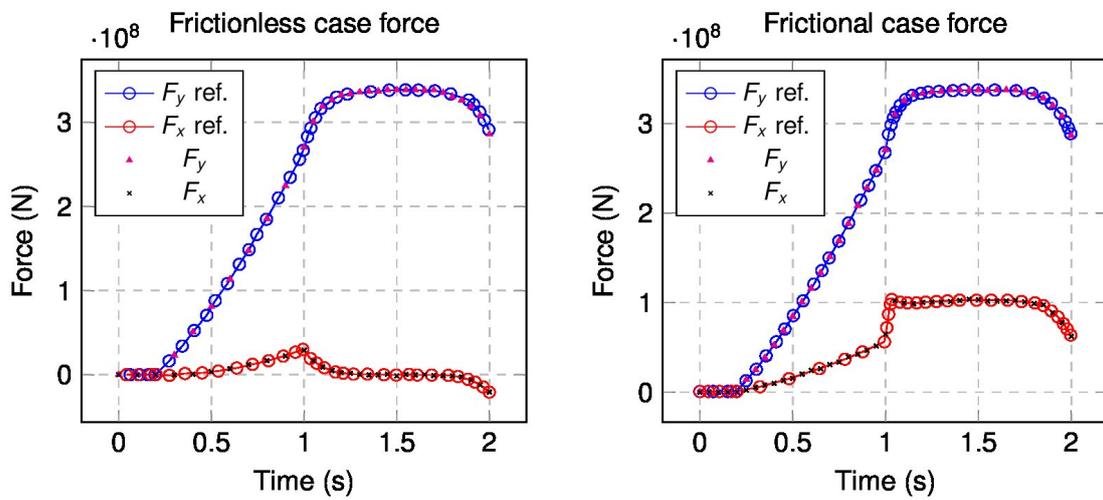


Figure 4.90: Shallow ironing test solution, comparing frictional and frictionless cases



(a) Frictionless solution

(b) Frictional solution

Figure 4.91: Reaction solution for shallow ironing test

4.6 Derivatives for contact mechanics linearization

On this section we can find in extension the definition of the directional derivatives, also called *Gateaux* derivatives [BookBel+14], associated with the mortar operators (4.92) and the slip and gap (4.93). These derivatives will be noted with Δ . With the components shown on (4.92), in addition to the derivatives of the normal and tangent (4.93), it is possible to define consistently the proper consistent linearisation of the tangent operator for the contact problem. The derivatives of the normal and tangent must be taken into account in order to define the derivatives of the gap and the slip. Similar developments can be found on other works, as Popp [PhDPop12; ArtPGW09; ArtPop+10], Yang [ArtYLM], Hammer [PhDHam13] and Cavalieri [ArtCFC12; ArtCC15; ArtCC13b] work.

First the definitions concerning the 2D problem will be presented (4.6.1. Derivatives for 2D contact), later the concerning to the 3D problem (4.6.2. Derivatives for 3D contact). The presenting work will focus particularly on the linear shape functions, as long as or work focuses on linear geometries.

$$(4.92a) \quad \Delta \mathbf{D}[j, k] = \sum_{g=1}^{n_{gp}} w_g \Delta \phi_{gj} N_{gk}^1 J_g^1 + \sum_{g=1}^{n_{gp}} w_g \phi_{gj} \Delta N_{gk}^1 J_g^1 + \sum_{g=1}^{n_{gp}} w_g \phi_{gj} N_{gk}^1 \Delta J_g^1$$

$$(4.92b) \quad \Delta \mathbf{M}[j, l] = \sum_{g=1}^{n_{gp}} w_g \Delta \phi_{gj} N_{gk}^2 J_g^1 + \sum_{g=1}^{n_{gp}} w_g \phi_{gj} \Delta N_{gk}^2 J_g^1 + \sum_{g=1}^{n_{gp}} w_g \phi_{gj} N_{gk}^2 \Delta J_g^1$$

Expressing the computation of the *weighted gap* in an algebraic way, based on the *mortar* operators definition.

$$(4.93a) \quad \tilde{g}_i = \mathbf{n}_i \cdot (\mathbf{D}_n \mathbf{x}_1 - \mathbf{M}_n \mathbf{x}_2)_i$$

Being i the node index and n the current time step, then

$$(4.93b) \quad \Delta \tilde{g}_i = \Delta \mathbf{n}_i \cdot (\mathbf{D}_n \mathbf{x}_1 - \mathbf{M}_n \mathbf{x}_2)_i + \mathbf{n}_i \cdot (\Delta \mathbf{D}_n \mathbf{x}_1 - \Delta \mathbf{M}_n \mathbf{x}_2 + \mathbf{D}_n \Delta \mathbf{x}_1 - \mathbf{M}_n \Delta \mathbf{x}_2)_i$$

In the same way, expressing the slip on an algebraic way, the derivative of the slip would be:

$$(4.93c) \quad \tilde{u}_{\tau i} = \tau_i \cdot ((\mathbf{D}_n - \mathbf{D}_{n-1}) \mathbf{x}_1 - (\mathbf{M}_n - \mathbf{M}_{n-1}) \mathbf{x}_2)_i$$

Being i the node index and $n, n-1$ the current and previous time steps, then

$$(4.93d) \quad \begin{aligned} \Delta \tilde{u}_{\tau i} = & \Delta \tau_i \cdot ((\mathbf{D}_n - \mathbf{D}_{n-1}) \mathbf{x}_1 - (\mathbf{M}_n - \mathbf{M}_{n-1}) \mathbf{x}_2)_i \\ & + \tau_i \cdot ((\Delta \mathbf{D}_n - \mathbf{D}_{n-1}) \mathbf{x}_1 - (\Delta \mathbf{M}_n - \mathbf{M}_{n-1}) \mathbf{x}_2)_i \\ & + \tau_i \cdot ((\mathbf{D}_n - \mathbf{D}_{n-1}) \Delta \mathbf{x}_1 - (\mathbf{M}_n - \mathbf{M}_{n-1}) \Delta \mathbf{x}_2)_i \end{aligned}$$

After the definition of the following directional derivatives (Δ), it is necessary it is algebraic representation of them on an equivalent matrix-vector format. This process is quite standard for **FE** formulations, and in our case it is done in an automatic way using the **AD**, for more details we address to the corresponding Appendix C. *Automatic differentiation*. The code for the tests for the quadratic convergence shown on the following sections are accessible in the public repository of *Kratos*, implemented on *Kratos C++* unittest format. The following [link](#) guides toward it.

4.6.1 Derivatives for 2D contact

In this section the derivatives concerning 2D linear lines, necessities in order to properly compute the contact are presented, see (4.92) and (4.93). The derivatives are tested using a set of pairs following the Figure 4.92 structure, the Table 4.20 contains the coordinates of the nodes of the actual configurations tested. The perturbations applied on each pair are defined on Table 4.21, which lists the nodes perturbed and its amplitude.

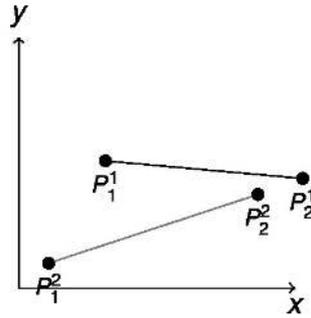


Figure 4.92: Line validation geometries

Pair	P_1^1	P_2^1	P_1^2	P_2^2
1	-1.0 0.0 0.0	1.0 0.0 0.0	1.2 0.0 0.0	-0.8 0.0 0.0
2	-1.0 0.0 0.0	1.0 0.0 0.0	1.2 1.0E ⁻³ 0.0	-0.8 1.0E ⁻³ 0.0
3	-1.0 0.0 0.0	-1.0 5.0E ⁻³ 0.0	-1.0 1.0E ⁻³ 0.0	-1.0 7.0E ⁻⁴ 0.0

Table 4.20: 2D lines pairs geometrical configuration

Case	Node perturbed	Perturbation amplitude
1	1	-5.0e - 2
2	1	-5.0e - 2
3	1	-1.0e - 1

Table 4.21: 2D lines perturbation amplitudes

4.6.1.1 Jacobians

4.6.1.1.1 Theory :

As the integration of the mortar formulation is performed only in the slave (mortar) domain, then the derivative of the *Jacobian* is defined only on the slave domain. Taking into consideration the definition of the *Jacobian* as (4.94a), it is possible to define the directional derivative of the slave *Jacobian* determinant at each *Gauss* point in a 2D linear line as (4.94b). In order to simplify and due to its extended use, we will call $\det(J)$ as J .

$$(4.94a) \quad J(\xi) = \left\| \sum_{k=1}^n N_{k,\xi}(\xi) \mathbf{x}_k \right\|$$

$$(4.94b) \quad \Delta J(\xi_g^1) = \frac{\sum_{k=1}^n N_{k,\xi}(\xi_g^1) \mathbf{x}_k}{\left\| \sum_{k=1}^n N_{k,\xi}(\xi_g^1) \mathbf{x}_k \right\|} \cdot \left(\sum_{k=1}^n N_{k,\xi}(\xi_g^1) \Delta \mathbf{x}_k \right)$$

4.6.1.1.2 Convergence study :

Finally, in the Figure 4.93, the convergence plot of the *Jacobian* derivatives for the 3 different geometrical pairs from Table 4.20 is shown. In this figure, and the followings, L_2 represents the *Euclidean* norm of the error, difference between the estimated value and the exact one. The plot is created considering in the x-axis the amplitude of the perturbation, and in the y the logarithm of L_2 of the error; therefore the slope indicates the convergence rate of the error in function of the perturbation.

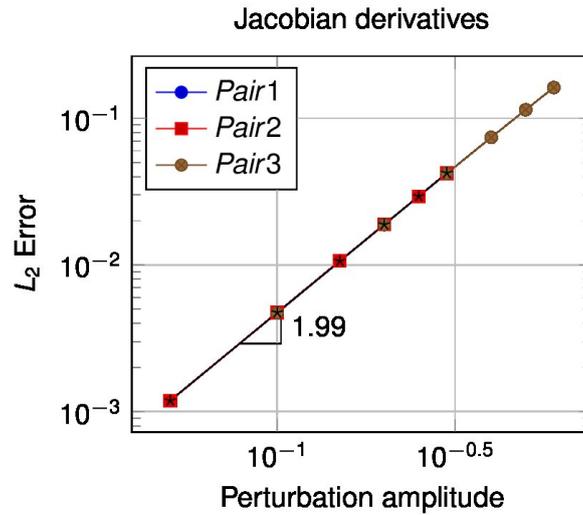


Figure 4.93: Convergence plot for the *Jacobian* derivatives for the 2D linear line

4.6.1.2 Shape functions

4.6.1.2.1 Theory :

The shape functions of both domains depend on the integration segments between the two domains on contact. In order to compute the shape functions of a 2D line, we do as (4.95a), Figure 4.94, where ξ corresponds with the local coordinate of the line. With this, its derivatives depend only on the derivative of the local coordinate ξ (4.95b). Despite the fact that lines in Figure 4.94 are not in contact, due to its relative position, will be affected by the perturbations, allowing to measure the derivative convergence.

$$(4.95a) \quad \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1 - \xi) \\ \frac{1}{2}(1 + \xi) \end{bmatrix}$$

$$(4.95b) \quad \begin{bmatrix} \Delta N_1 \\ \Delta N_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\Delta\xi \\ \frac{1}{2}\Delta\xi \end{bmatrix}$$

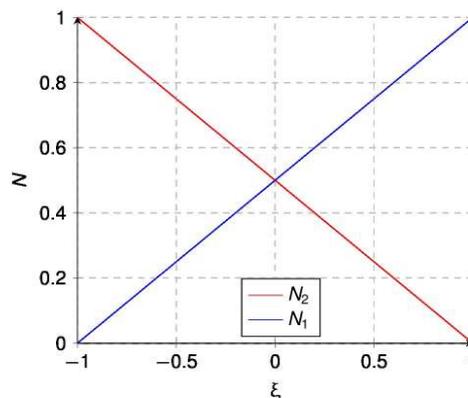


Figure 4.94: Shape functions for the 2D linear line

For this reason we need to compute on first place the derivatives of the local coordinates, which depends on the derivatives of the integration segments. We will separate in two small sections those derivations. The derivation of these terms is based on the work of *Laursen and Yang*[ArtYLM] and *Popp*[PhDPop12].

4.6.1.2.2 Integration segments :

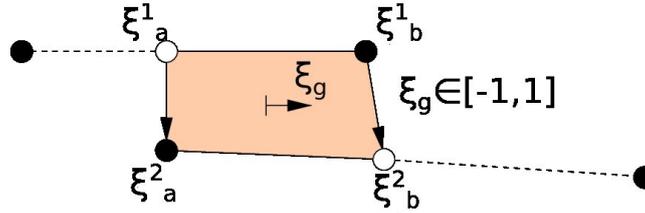


Figure 4.95: Integration segment for a linear line

The derivation of the integration points require the derivation of the integration segments. As previously presented this corresponds with the current and effective section of the line on contact, see Figure 4.95. As can be seen on this picture we have two possible scenarios, one where the projected slave node (ξ_a^1) corresponds with the original slave node and the second case corresponds with the counterpart scenario for the master side (ξ_a^2). The same logic is applied for the end integration segment coordinates (ξ_b^1 and ξ_b^2). If the integrated node corresponds with the original geometry the derivative therefore will be 0.

The derivative Δ for this first scenario is presented on (4.96), where the sub-index a for the coordinates (\mathbf{x}_a^1) and normal (\mathbf{n}_a) represent the first node on the slave side. On this case (4.96b) the shape functions and shape function gradients are computed on ξ_a^2 .

$$(4.96a) \quad \Delta \xi_a^1 = 0$$

$$(4.96b) \quad \Delta \xi_a^2 = -\frac{1}{\left(\sum_{l=1}^{n_m^c} N_{l,\xi}^2 x_l^2\right) n_a^y - \left(\sum_{l=1}^{n_m^c} N_{l,\xi}^2 y_l^2\right) n_a^x} \cdot \left[\left(\sum_{l=1}^{n_m^c} (N_l^2 \Delta x_l^2) - \Delta x_a^1 \right) n_a^y - \left(\sum_{l=1}^{n_m^c} (N_l^2 \Delta y_l^2) - \Delta y_a^1 \right) n_a^x \right] + \left[\left(\sum_{l=1}^{n_m^c} (N_l^2 x_l^2) - x_a^1 \right) \Delta n_a^y - \left(\sum_{l=1}^{n_m^c} (N_l^2 y_l^2) - y_a^1 \right) \Delta n_a^x \right]$$

On the second case (4.97), as the slave side, which is our reference domain, does not correspond with its projection the \mathbf{n} considered then is the one interpolated for the projected point. This means that the interpolation of the nodal normal will be computed. On this case (4.97a) the shape functions and shape function gradients are computed on ξ_a^1 .

$$(4.97a) \quad \Delta \xi_a^1 = \frac{num}{denom}$$

$$(4.97b) \quad denom = - \left[\sum_{k=1}^{n_s^c} N_{k,\xi}^1 x_k^1 \right] \left[\sum_{k=1}^{n_s^c} N_k^1 n_k^y \right] + \left[\sum_{k=1}^{n_s^c} N_{k,\xi}^1 y_k^1 \right] \left[\sum_{k=1}^{n_s^c} N_k^1 n_k^x \right] - \left[\sum_{k=1}^{n_s^c} (N_k^1 x_k^1) - x_a^2 \right] \left[\sum_{k=1}^{n_s^c} N_{k,\xi}^1 n_k^y \right] + \left[\sum_{k=1}^{n_s^c} (N_k^1 y_k^1) - y_a^2 \right] \left[\sum_{k=1}^{n_s^c} N_{k,\xi}^1 n_k^x \right]$$

$$(4.97c) \quad \begin{aligned} num = & \left[\sum_{k=1}^{n_s^c} (N_k^1 \Delta x_k^1) - \Delta x_a^2 \right] \left[\sum_{k=1}^{n_s^c} N_k^1 n_k^y \right] - \left[\sum_{k=1}^{n_s^c} (N_k^1 \Delta y_k^1) - \Delta y_a^2 \right] \left[\sum_{k=1}^{n_s^c} N_k^1 n_k^x \right] \\ & + \left[\sum_{k=1}^{n_s^c} (N_k^1 x_k^1) - x_a^2 \right] \left[\sum_{k=1}^{n_s^c} N_k^1 \Delta n_k^y \right] - \left[\sum_{k=1}^{n_s^c} (N_k^1 y_k^1) - y_a^2 \right] \left[\sum_{k=1}^{n_s^c} N_k^1 \Delta n_k^x \right] \end{aligned}$$

$$(4.97d) \quad \Delta \xi_a^2 = 0$$

4.6.1.2.3 Local coordinates (Gauss points) :

With the values obtained in the previous expression, it is possible to evaluate the Δ of the **Gauss Point (GP)** coordinates, and therefore calculate the values for the shape function derivatives. We evaluate those derivatives both on the slave side (ξ_g^1 , (4.98a)) and the master side (ξ_g^2 , (4.98b)).

For the slave side, we just need to apply the definition for the **GP** and derive it. As it consists on a simple sum, the derivative is straightforward.

$$(4.98a) \quad \Delta \xi_g^1 = \frac{1}{2} (1 - \xi_g) \Delta \xi_a^1 + \frac{1}{2} (1 + \xi_g) \Delta \xi_b^1$$

For the master side, as it depends on a projection, the linearisation requires to take that projection into consideration, resulting the following expression. All the shape functions and shape function gradients are computed on ξ_g^2 .

$$(4.98b) \quad \begin{aligned} \Delta \xi_g^2 = & - \frac{1}{\left(\sum_{k=1}^{n_m^c} N_k^2 x_k^2 \right) n_g^y - \left(\sum_{k=1}^{n_m^c} N_k^2 y_k^2 \right) n_g^x} \\ & \cdot \left[\left(\sum_{k=1}^{n_m^c} (N_k^2 \Delta x_k^2) - \Delta x_g^1 \right) n_g^y - \left(\sum_{k=1}^{n_m^c} (N_k^2 \Delta y_k^2) - \Delta y_g^1 \right) n_g^x \right. \\ & \left. + \left(\sum_{k=1}^{n_m^c} (N_k^2 x_k^2) - x_g^1 \right) \Delta n_g^y - \left(\sum_{k=1}^{n_m^c} (N_k^2 y_k^2) - y_g^1 \right) \Delta n_g^x \right] \end{aligned}$$

4.6.1.2.4 Convergence study :

Finally, in the Figure 4.96, the convergence plot of the shape function derivatives for the 3 different geometrical pairs from Table 4.20 is shown.

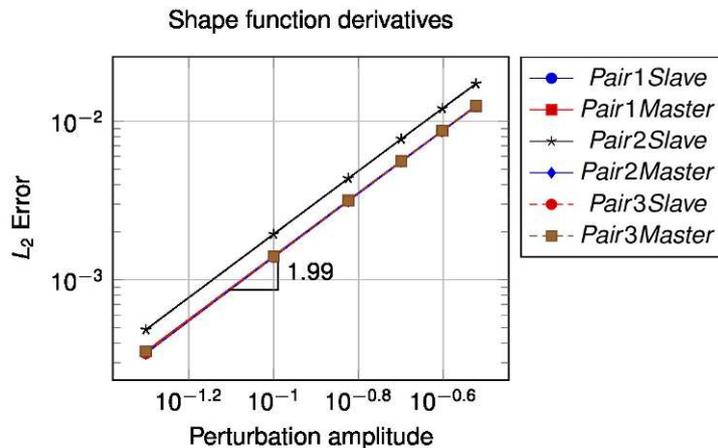


Figure 4.96: Convergence plot for the shape function derivatives for the 2D linear line

4.6.1.3 Dual shape functions

4.6.1.3.1 Theory :

On standard **FEM** shape functions defined on the reference configuration, by definition these are not dependant of the deformation of the domain. This is not the case for dual shape functions, this is due to its intrinsic dependence on the J of the slave side, see (4.99a). This means that our dual shape functions are deformation-dependant. In addition we shall consider its dependence with the standard shape function derivatives. All this together give us the following expression (4.99b). The \mathbf{A}_e definition is presented in the corresponding section, 4.3.3.4.1. *Dual Lagrange multipliers*, on the Equation (4.24).

$$(4.99a) \quad \Phi = \mathbf{A}_e N^1 \begin{cases} \mathbf{A}_e(N, J) \\ N(N) \end{cases}$$

$$(4.99b) \quad \Delta\Phi = \mathbf{A}_e \Delta N^1 + \Delta\mathbf{A}_e N^1$$

So in addition to the terms from the 4.6.1.2. *Shape functions* we need to evaluate the corresponding terms associated to \mathbf{A}_e (4.100).

$$(4.100) \quad \begin{aligned} \Delta\mathbf{A}_e &= \Delta\mathbf{D}_e \mathbf{M}_e^{-1} - \mathbf{D}_e \Delta\mathbf{M}_e \mathbf{M}_e^{-1} \\ \Delta\mathbf{D}_e &= \Delta[d_{jk}] \in \mathbb{R}^{m_e^1 \times m_e^1}, \Delta d_{jk} = \delta_{jk} \sum_{g=1}^{n_{gp}} w_g N_{gk}^1 \Delta J_g^1 \\ \Delta\mathbf{M}_e &= \Delta[m_{jk}] \in \mathbb{R}^{m_e^1 \times m_e^1}, \Delta m_{jk} = \sum_{g=1}^{n_{gp}} w_g \sum_{g=1}^{n_{gp}} w_g N_{gj}^1 N_{gk}^1 \Delta J_g^1 \end{aligned}$$

4.6.1.3.2 Convergence study :

Finally, in the Figure 4.97, the convergence plot of the dual shape function derivatives for the 3 different geometrical pairs from Table 4.20 is shown.

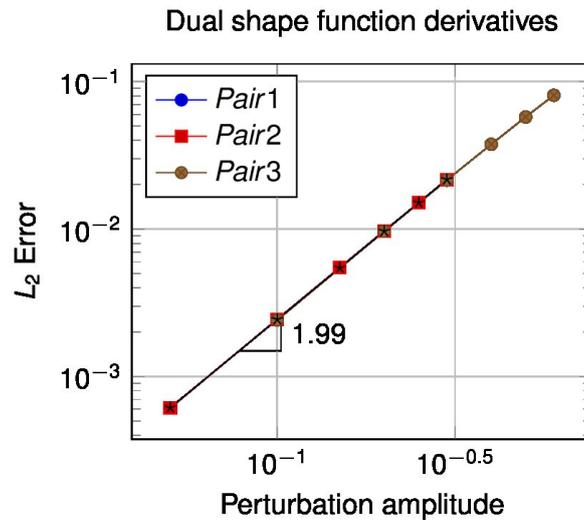


Figure 4.97: Convergence plot for the dual shape function derivatives for the 2D linear line

4.6.1.4 Normal and tangent vectors

4.6.1.4.1 Theory :

The linearisation of the tangent vectors is associated, as its definition depends on, on the linearisation of the normal vector. On these lines only the definition of the normal derivative, it is included as the definition of the tangent derivative is defined as the cross product between the normal derivative and the out of plane vector ($\Delta\tau = \mathbf{v}_z \times \Delta\mathbf{n}$).

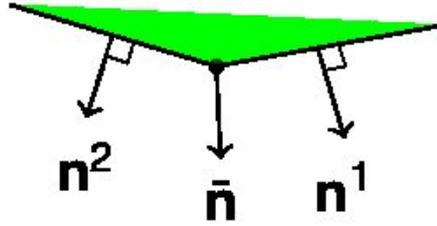


Figure 4.98: Normal average for a 2D line. Proposed by *Taylor and Papadopoulos*[ArtPT92]

The definition of the normal vector (\mathbf{n}) for a 2D line is defined on (4.101a), where x_i, y_i correspond with the coordinates of the i node of the line (1,2). The unit normal corresponding to (4.101a) can be obtained with (4.101b).

Then the average normal[ArtPT92] ($\bar{\mathbf{n}}$) is defined as (4.101c), this definition is general and can be extended to 3D. Figure 4.98 represents the normal average for a 2D line. It is relevant to mention that the average normal (4.101c) is already computed over unitary normals, in contrast with the average area normal proposed by *Popp*[PhDPop12], this is done due to practical reasons, as this approach was significantly more robust (better convergence) in several numeric studies, particularly when coarse meshes were considered.

$$(4.101a) \quad \mathbf{n}_{area} = \begin{bmatrix} y_2 \\ x_1 \end{bmatrix} - \begin{bmatrix} y_1 \\ x_2 \end{bmatrix}$$

$$(4.101b) \quad \mathbf{n} = \frac{\mathbf{n}_{area}}{\|\mathbf{n}_{area}\|}$$

In order to obtain the average normal we sum the normals of the neighbours entities (\mathbf{n}) to the node, then we make unitary dividing by its norm.

$$(4.101c) \quad \bar{\mathbf{n}} = \frac{\sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c}{\left\| \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \right\|}$$

So, once defined the normal (4.102) and its average value (4.103) we can define its directional derivative.

$$(4.102a) \quad \Delta\mathbf{n} = \frac{\Delta\mathbf{n}_{area} \|\mathbf{n}_{area}\| - \mathbf{n}_{area} \Delta\|\mathbf{n}_{area}\|}{\|\mathbf{n}_{area}\|^2}$$

For the 2D line the resulting expression for the \mathbf{n}_{area} derivative will be (4.102b), which corresponds with taken the corresponding unitary vector depending of the derivative evaluated.

$$(4.102b) \quad \Delta\mathbf{n}_{area} = \Delta \begin{bmatrix} y_2 \\ x_1 \end{bmatrix} - \Delta \begin{bmatrix} y_1 \\ x_2 \end{bmatrix}$$

For the directional derivative of the average normal, the resulting expression is decomposed in its different components.

$$(4.103a) \quad \Delta \bar{\mathbf{n}} = \frac{\Delta \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \left\| \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \right\| - \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \Delta \left\| \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \right\|}{\left\| \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \right\|^2}$$

The first components are easy to obtain, as the derivative of a sum is the sum of its derivatives.

$$(4.103b) \quad \Delta \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c = \sum_{c=1}^{n_j^{neigh}} \Delta \mathbf{n}^c$$

The second components require to calculate the derivative of a norm, proceeding the standard manner the expression obtained is (4.103c).

$$(4.103c) \quad \Delta \left\| \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \right\| = \frac{\sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \Delta \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c}{\left\| \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c \right\|} = \bar{\mathbf{n}} \Delta \sum_{c=1}^{n_j^{neigh}} \mathbf{n}^c$$

Additionally in order to evaluate the value of the normal derivative on a specific **GP**, calculated on the corresponding local coordinates ξ_g^1 . The procedure to obtain is straightforward as only requires to apply the chain rule on the shape function interpolation of the normal evaluated on the **GP** (4.104).

$$(4.104) \quad \Delta \mathbf{n}_g = \sum_{k=1}^{n_s^c} N_{k,\xi}^1(\xi_g^1) \Delta \xi_g^1 \mathbf{n}_k + \sum_{k=1}^{n_s^c} N_k^1(\xi_g^1) \Delta \mathbf{n}_k$$

4.6.1.4.2 Convergence study :

Finally, in the Figure 4.99, the convergence plot of the normal derivatives for the 3 different geometrical pairs from Table 4.20 is shown.

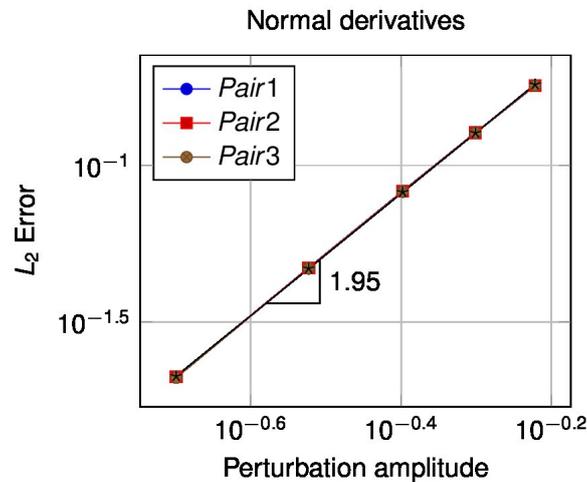
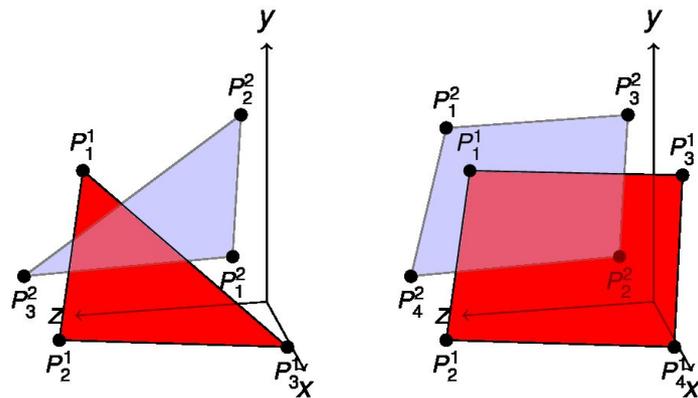


Figure 4.99: Convergence plot for the normal derivatives for the 2D linear line

4.6.2 Derivatives for 3D contact

In this section the derivatives concerning 3D linear triangles and bilinear quadrilaterals, as part of (4.92) and (4.93) are presented. The main difference between this approach and the one provided for 2D cases is the segmentation of the surfaces, where the actual integration is done. The derivatives are tested using a set of pairs following the Figure 4.100a structure and the Table 4.22 for the case of the triangle. Again, the perturbations applied on each pair are defined on Table 4.23.



(a) Triangle validation geometries (b) Quadrilateral validation geometries

Pair	P_1^1	P_2^1	P_3^1	P_1^2	P_2^2	P_3^2
1	0.0 0.0 0.0	1.0 0.0 0.0	0.0 1.0 0.0	0.0 1.0 1.0E-3	0.0 0.0 1.0E-3	1.0 0.0 1.0E-3
2	-0.1 0.1 1.0E-3	1.1 0.2 0.0	0.1 1.0 0.0	-0.1 1.3 1.0E-3	0.1 0.2 1.0E-3	1.2 0.2 2.0E-3
3	-0.1 0.1 1.0E-3	1.1 0.2 0.0	0.1 1.0 0.0	-0.1 1.3 1.0E-3	0.1 0.2 1.0E-3	1.2 0.2 2.0E-3
4	0.0 0.0 0.0	1.0 0.0 0.0	0.0 1.0 0.0	0.0 1.0 1.0E-3	0.0 0.0 1.0E-3	1.0 0.0 1.0E-3
5	-0.1 0.1 1.0E-3	1.1 0.2 0.0	0.1 1.0 0.0	-0.1 1.3 1.0E-3	0.1 0.2 1.0E-3	1.2 0.2 2.0E-3
6	0.0 0.0 0.0	1.0 0.0 0.0	0.0 1.0 0.0	-0.1 1.0 1.0E-3	0.0 0.0 1.0E-3	1.0 0.0 1.0E-3

Table 4.22: 3D triangles pairs geometrical configuration

Case	Node perturbed (I)	Perturbation amplitude (I)	Node perturbed (II)	Perturbation amplitude (II)
1	4	-5.0e-1	-	-
2	4	-5.0e-3	-	-
3	4	1.0e-1	5	1.0e-1
4	4	1.0e-1	5	5.0e-2
5	1	5.0e-2	-	-
6	4	-5.0e-2	-	-

Table 4.23: 3D triangles perturbation amplitudes

The derivative tests for quadrilateral pairs are defined following Figure 4.100b and Table 4.24. The Table 4.25 defines the perturbations.

Pair	P_1^1	P_2^1	P_3^1	P_4^1	P_1^2	P_2^2	P_3^2	P_4^2
1	0.0 0.2 1.0E-3	1.0 0.2 1.0E-3	1.1 1.1 0.0	0.2 1.0 0.0	-0.1 1.0 1.0E-3	1.0 1.1 1.0E-3	1.0 0.1 2.0E-3	0.0 0.1 2.0E-3
2	0.0 0.0 0.0	1.0 0.0 0.0	1.0 1.0 0.0	0.0 1.0 0.0	-0.1 1.0 1.0E-3	1.0 1.0 1.0E-3	1.0 0.0 1.0E-3	0.0 0.0 1.0E-3
3	0.0 0.3 2.0E-3	1.0 0.2 1.0E-3	1.2 1.1 0.0	0.2 1.1 0.0	-0.1 1.0 2.0E-3	1.2 1.1 2.0E-3	1.0 0.1 3.0E-3	0.1 0.1 3.0E-3

Table 4.24: 3D quadrilaterals pairs geometrical configuration

Case	Node perturbed	Perturbation amplitude
1	5	-5.0e-3
2	5	-5.0e-3
3	5	-5.0e-3

Table 4.25: 3D quadrilaterals perturbation amplitudes

4.6.2.1 Jacobians

4.6.2.1.1 Theory :

4.6.2.1.1.1 Integration segments derivatives :

Before introduce the theory that defines the directional derivatives of the J , we need to introduce the derivatives of the segments obtained during the integration. The resulting operation is presented on the Figure 4.100, and detailed on Figure 4.101. For more details about the method followed in order to evaluate the mortar integration, we address the corresponding section A.2.Mortar segmentation, where we debate the two different integration methods studied, and reason why we decided to consider a *segment base* approach.

The Figure 4.100 just represents the intersection and clipping technique, where the intersected points generate the triangles considered during the integration. The later, Figure 4.101, shows in deeper detail how this clipping technique works. The relevant part on this method is the consideration of the clipping coordinates, the resulting point coordinates (\mathbf{x}_{clip}) depend on the coordinates of the points from the slave segment ($\hat{\mathbf{x}}_1^1, \hat{\mathbf{x}}_2^1$) and master segment ($\hat{\mathbf{x}}_1^2, \hat{\mathbf{x}}_2^2$). This means that in order to compute properly the derivatives it is necessary to identify the corresponding nodes on the master and slave geometries when computing the clipping algorithm[BookSE02; BookHug+14].

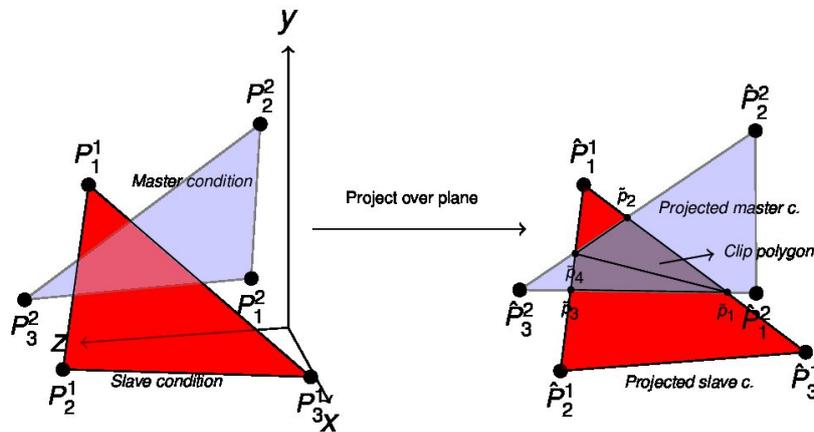


Figure 4.100: Intersection and clipping procedure during mortar segmentation

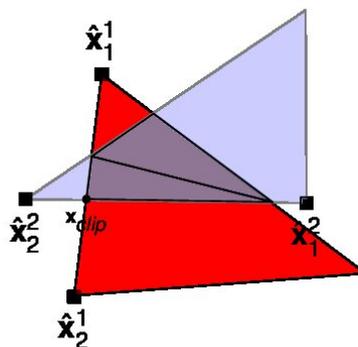


Figure 4.101: Detail on intersection on mortar segmentation

We need to differentiate two different scenarios when obtaining the intersections:

- The point of the integration triangle is part of the original master/slave geometry. This case is less complex to compute, as it takes directly the components of the master/slave side derivatives without direct dependence on the counterpart domain.

- The point which integrates the integration triangle belongs to an intersection (most general case). All points show on Figure 4.101 belong to this category.

For the first case, we need to differentiate if the node is a slave node (4.105a) or a master one (4.105b). Then its derivatives are computed respectively according to (4.106), (4.106a) for the slave and (4.106b) for the master.

$$(4.105a) \quad \mathbf{x}_{clip} = \mathbf{x}^1 - [(\mathbf{x}^1 - \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane}] \mathbf{n}_{plane}$$

$$(4.105b) \quad \mathbf{x}_{clip} = \mathbf{x}^2 - [(\mathbf{x}^2 - \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane}] \mathbf{n}_{plane}$$

$$(4.106a) \quad \begin{aligned} \Delta \mathbf{x}_{clip} &= \Delta (\mathbf{x}^1 - [(\mathbf{x}^1 - \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane}] \mathbf{n}_{plane}) \\ &= \Delta \mathbf{x}^1 - [(\Delta \mathbf{x}^1 - \Delta \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane} + (\mathbf{x}^1 - \mathbf{x}_{plane}^1) \cdot \Delta \mathbf{n}_{plane}] \mathbf{n}_{plane} - [(\mathbf{x}^1 - \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane}] \Delta \mathbf{n}_{plane} \end{aligned}$$

$$(4.106b) \quad \begin{aligned} \Delta \mathbf{x}_{clip} &= \Delta (\mathbf{x}^2 - [(\mathbf{x}^2 - \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane}] \mathbf{n}_{plane}) \\ &= \Delta \mathbf{x}^2 - [(\Delta \mathbf{x}^2 - \Delta \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane} + (\mathbf{x}^2 - \mathbf{x}_{plane}^1) \cdot \Delta \mathbf{n}_{plane}] \mathbf{n}_{plane} - [(\mathbf{x}^2 - \mathbf{x}_{plane}^1) \cdot \mathbf{n}_{plane}] \Delta \mathbf{n}_{plane} \end{aligned}$$

For the intersection case, we need to define the clip algorithm in order to obtain the corresponding coordinates. We will consider the *Foley*[BookHug+14] clipping algorithm can be defined as follows (4.107), an alternative approach can be to project $\hat{\mathbf{x}}_1^1$, $\hat{\mathbf{x}}_2^1$, $\hat{\mathbf{x}}_1^2$ and $\hat{\mathbf{x}}_2^2$ into the auxiliary plane defined by \mathbf{n}_{plane} prior to compute the clipping coordinates. The corresponding derivative, originally deduced by *Puso and Laursen*[ArtPL04], can be found in (4.108), where we need to decompose into its different components due to long expression deduced.

$$(4.107) \quad \mathbf{x}_{clip} = \hat{\mathbf{x}}_1^1 - \frac{(\hat{\mathbf{x}}_1^1 - \hat{\mathbf{x}}_1^2) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}}{(\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}} (\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1)$$

$$(4.108a) \quad \begin{aligned} \Delta \mathbf{x}_{clip} &= \Delta \hat{\mathbf{x}}_1^1 - \Delta \left(\frac{(\hat{\mathbf{x}}_1^1 - \hat{\mathbf{x}}_1^2) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}}{(\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}} (\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \right) \\ &= \Delta \hat{\mathbf{x}}_1^1 - \frac{num}{denom} (\Delta \hat{\mathbf{x}}_2^1 - \Delta \hat{\mathbf{x}}_1^1) - \frac{\Delta num \cdot denom - num \cdot \Delta denom}{denom^2} (\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \end{aligned}$$

The second term is quite complex, that's why we have decomposed into different components. The expression for the numerator corresponds with (4.108d), and (4.108e) for the denominator.

$$(4.108b) \quad num = (\hat{\mathbf{x}}_1^1 - \hat{\mathbf{x}}_1^2) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}$$

$$(4.108c) \quad denom = (\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}$$

Then the respective derivatives are defined as:

$$(4.108d) \quad \begin{aligned} \Delta num &= \Delta ((\hat{\mathbf{x}}_1^1 - \hat{\mathbf{x}}_1^2) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}) \\ &= ((\hat{\mathbf{x}}_1^1 - \hat{\mathbf{x}}_1^2) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2)) \cdot \Delta \mathbf{n}_{plane} + ((\Delta \hat{\mathbf{x}}_1^1 - \Delta \hat{\mathbf{x}}_1^2) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) + (\hat{\mathbf{x}}_1^1 - \hat{\mathbf{x}}_1^2) \times (\Delta \hat{\mathbf{x}}_2^2 - \Delta \hat{\mathbf{x}}_1^2)) \cdot \mathbf{n}_{plane} \end{aligned}$$

$$(4.108e) \quad \begin{aligned} \Delta denom &= \Delta ((\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) \cdot \mathbf{n}_{plane}) \\ &= ((\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2)) \cdot \Delta \mathbf{n}_{plane} + ((\Delta \hat{\mathbf{x}}_2^1 - \Delta \hat{\mathbf{x}}_1^1) \times (\hat{\mathbf{x}}_2^2 - \hat{\mathbf{x}}_1^2) + (\hat{\mathbf{x}}_2^1 - \hat{\mathbf{x}}_1^1) \times (\Delta \hat{\mathbf{x}}_2^2 - \Delta \hat{\mathbf{x}}_1^2)) \cdot \mathbf{n}_{plane} \end{aligned}$$

4.6.2.1.1.2 Jacobian derivatives :

Once obtained the integration cell vertex derivatives we can define the corresponding derivative of J . As the integration occurs on the cell vertex the proper *Jacobian* to consider is the J on the integration cell, J_{clip} , see (4.109a).

Considering the three vertices of the integration triangle as \mathbf{x}_{clip}^1 , \mathbf{x}_{clip}^2 and \mathbf{x}_{clip}^3 , and its respective derivatives, then derivative of J_{clip} can be calculated using (4.109b).

$$\begin{aligned}
 (4.109a) \quad J_{clip} &= \left\| (\mathbf{x}_{clip}^2 - \mathbf{x}_{clip}^1) \times (\mathbf{x}_{clip}^3 - \mathbf{x}_{clip}^1) \right\| \\
 \Delta J_{clip} &= \Delta \left\| (\mathbf{x}_{clip}^2 - \mathbf{x}_{clip}^1) \times (\mathbf{x}_{clip}^3 - \mathbf{x}_{clip}^1) \right\| \\
 (4.109b) \quad &= \frac{(\mathbf{x}_{clip}^2 - \mathbf{x}_{clip}^1) \times (\mathbf{x}_{clip}^3 - \mathbf{x}_{clip}^1)}{\left\| (\mathbf{x}_{clip}^2 - \mathbf{x}_{clip}^1) \times (\mathbf{x}_{clip}^3 - \mathbf{x}_{clip}^1) \right\|} \cdot [(\Delta \mathbf{x}_{clip}^2 - \Delta \mathbf{x}_{clip}^1) \times (\mathbf{x}_{clip}^3 - \mathbf{x}_{clip}^1)] \\
 &+ \frac{(\mathbf{x}_{clip}^2 - \mathbf{x}_{clip}^1) \times (\mathbf{x}_{clip}^3 - \mathbf{x}_{clip}^1)}{\left\| (\mathbf{x}_{clip}^2 - \mathbf{x}_{clip}^1) \times (\mathbf{x}_{clip}^3 - \mathbf{x}_{clip}^1) \right\|} \cdot [(\mathbf{x}_{clip}^2 - \mathbf{x}_{clip}^1) \times (\Delta \mathbf{x}_{clip}^3 - \Delta \mathbf{x}_{clip}^1)]
 \end{aligned}$$

4.6.2.1.2 Convergence study :

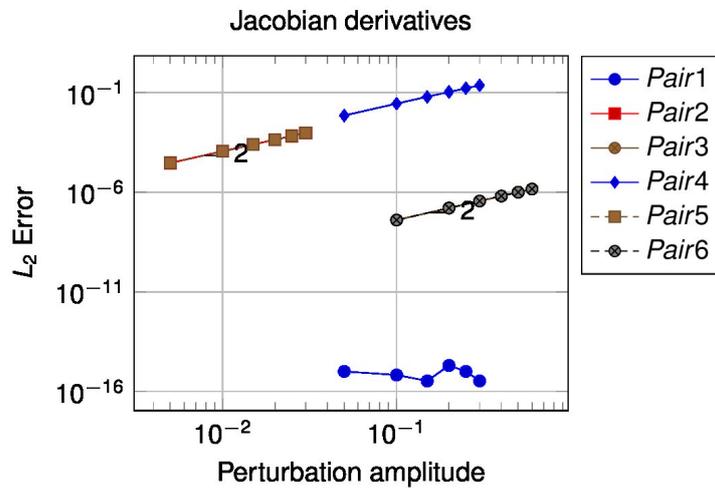


Figure 4.102: Convergence plot for the *Jacobian* derivatives for the 3D linear triangle

The convergence plot of the *Jacobian* derivatives for the 6 different geometrical pairs of the 3D linear triangles from Table 4.22 can be found in the Figure 4.102.

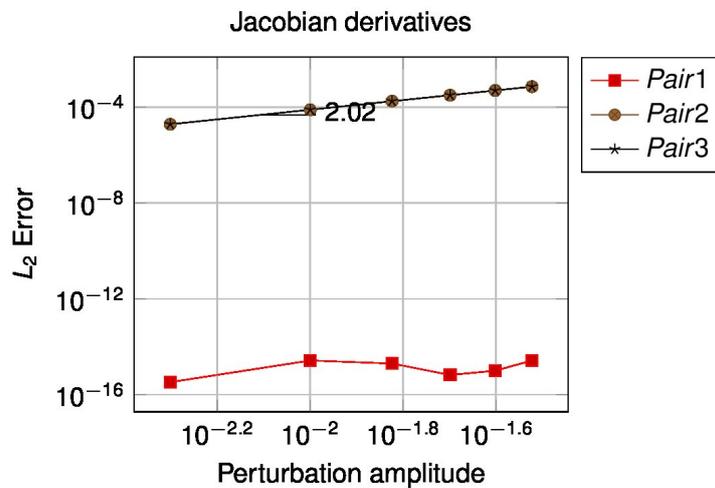


Figure 4.103: Convergence plot for the *Jacobian* derivatives for the 3D bilinear quadrilateral

The convergence plot of the *Jacobian* derivatives for the 3 different geometrical pairs of the 3D bilinear quadrilaterals from Table 4.24 can be found in the Figure 4.103.

4.6.2.2 Shape functions

4.6.2.2.1 Theory :

The same reasoning applied on the 2D case 4.6.1.2. Shape functions is applied here. This means we need to define first the shape functions and its derivatives. Those shape functions depend on its local coordinates (ξ, η) , then the derivatives depend on the derivatives of the local coordinates. Those derivatives will be obtained in a independent subsection. First, we define the shape functions of the 3D linear triangle (4.110a), and its derivatives (4.110b). On the Figure 4.104 we represent on 3D the shape functions according to ξ and η .

$$(4.110a) \quad \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} = \begin{bmatrix} 1 - \xi - \eta \\ \xi \\ \eta \end{bmatrix}$$

$$(4.110b) \quad \begin{bmatrix} \Delta N_1 \\ \Delta N_2 \\ \Delta N_3 \end{bmatrix} = \begin{bmatrix} -\Delta\xi & -\Delta\eta \\ \Delta\xi & 0 \\ 0 & \Delta\eta \end{bmatrix}$$

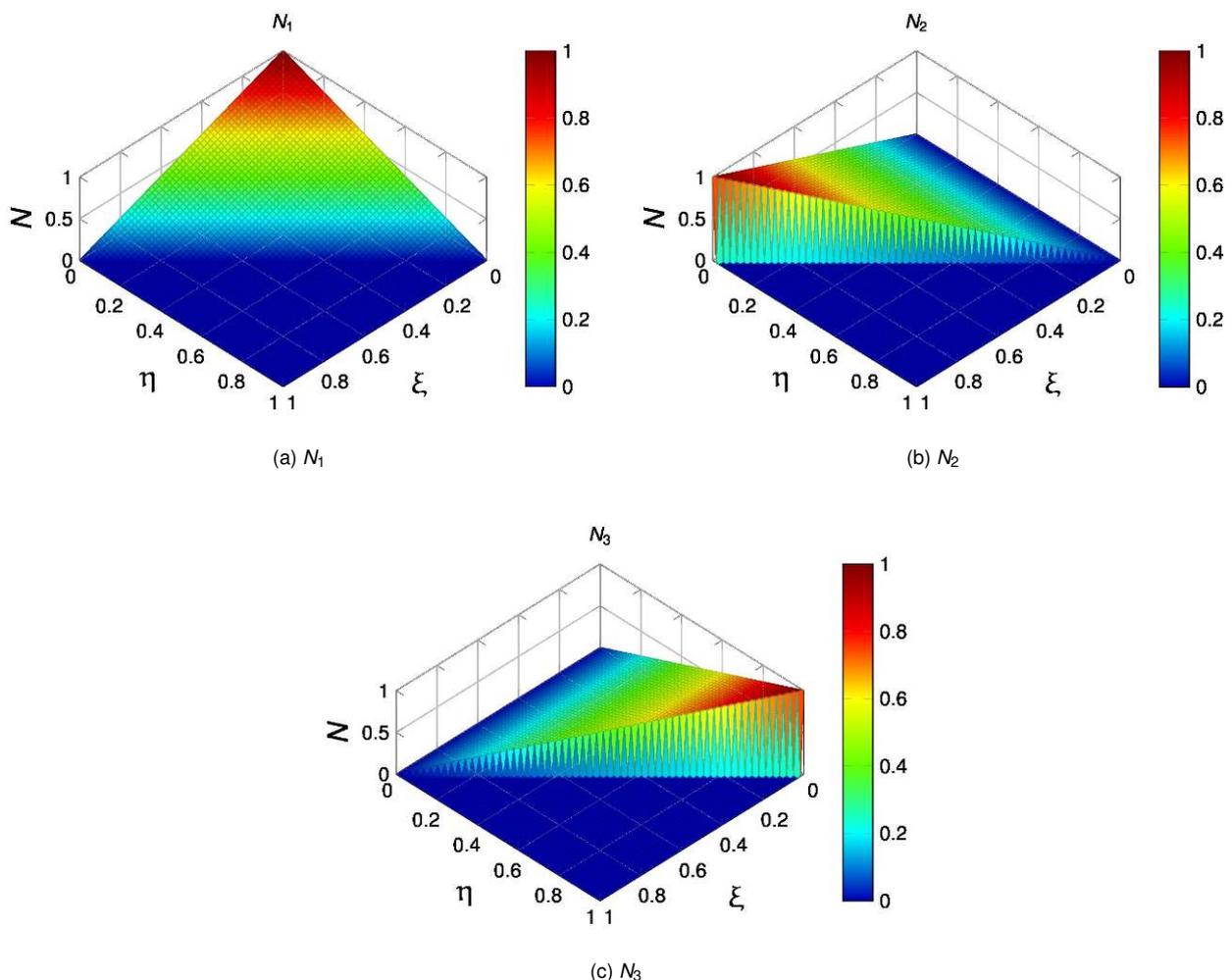


Figure 4.104: Shape functions for the 3D linear triangle

Second, we define the shape functions of the 3D bilinear quadrilateral and its derivatives at (4.111). The graphic representation of the shape functions are shown in the Figure 4.105.

$$(4.111a) \quad \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4}((1-\xi)(1-\eta)) \\ \frac{1}{4}((1+\xi)(1-\eta)) \\ \frac{1}{4}((1+\xi)(1+\eta)) \\ \frac{1}{4}((1-\xi)(1+\eta)) \end{bmatrix}$$

$$(4.111b) \quad \begin{bmatrix} \Delta N_1 \\ \Delta N_2 \\ \Delta N_3 \\ \Delta N_4 \end{bmatrix} = \begin{bmatrix} -\frac{\Delta\xi}{4} & -\frac{\Delta\eta}{4} \\ \frac{\Delta\xi}{4} & -\frac{\Delta\eta}{4} \\ \frac{\Delta\xi}{4} & \frac{\Delta\eta}{4} \\ -\frac{\Delta\xi}{4} & \frac{\Delta\eta}{4} \end{bmatrix}$$

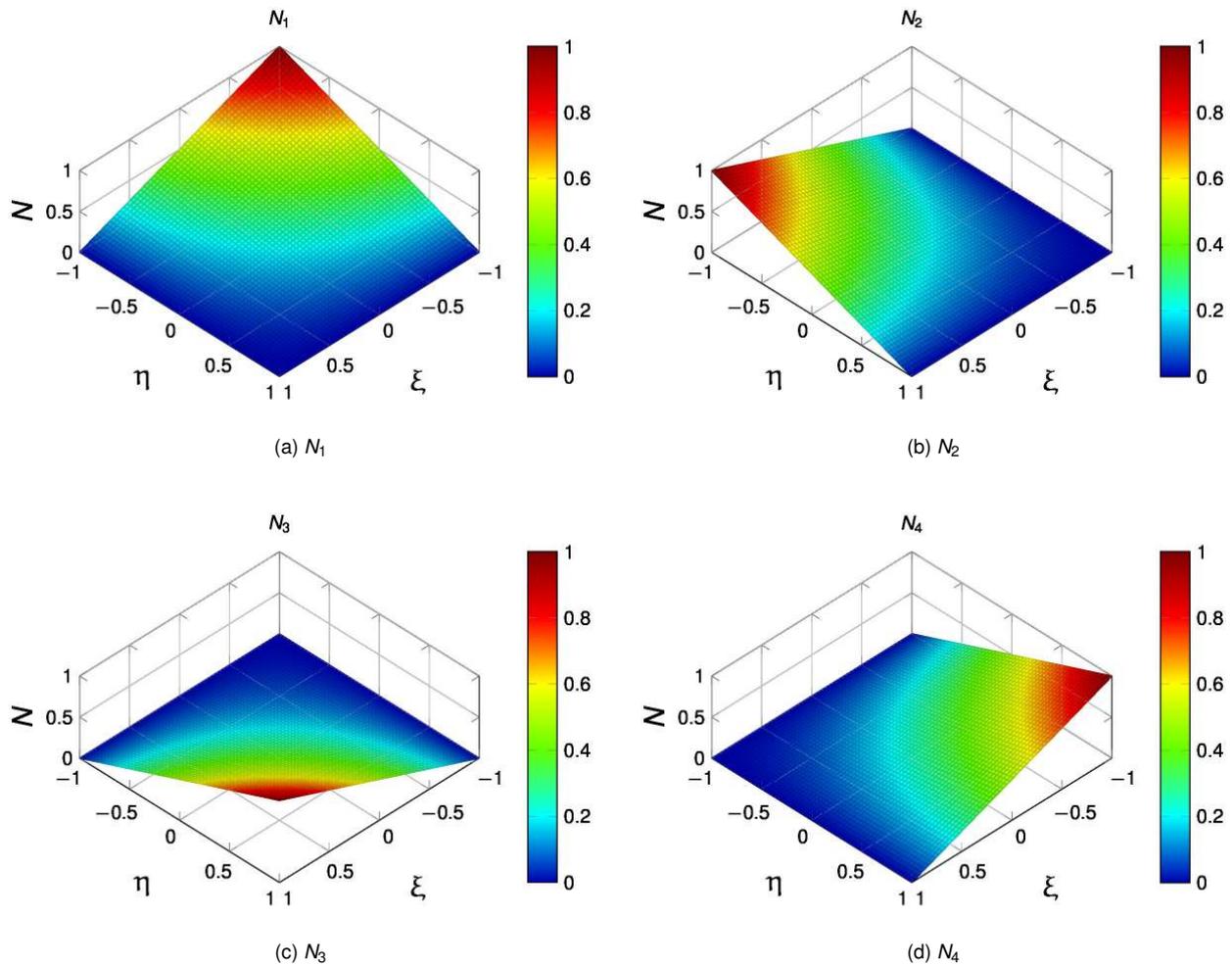


Figure 4.105: Shape functions for the 3D bilinear quadrilateral

4.6.2.2.2 Local coordinates (*Gauss points*) :

Here we will define the derivatives of the corresponding **GP** local coordinates of the integration triangles, Figure 4.101. Important to highlight the fact that the complexity of this task came from the fact that the inner **GP** from the integration triangles must be projected into the master and slave domains. This, in the most general case requires the resolution of a **NL** problem with a **NR** iterative process.

To obtain the local coordinates ξ and η we will consider the expression proposed by Popp[ArtPGW09], but simplified as our geometries are linear and consequently reduces the complexity of the system of equations. In order to obtain the coordinates of the **GP** we use (4.112a), where we can see as the coordinates of the **GP** in the integration triangle (represented with \tilde{N}) has an equivalent on the master/slave geometries. With this expression we can define this as a residual (4.112b) of a linear system of equations, which converges in one **NL** iteration. The complete system of equations (4.112c) require to compute the corresponding *Jacobian* (J) (4.112d).

$$(4.112a) \quad \begin{aligned} \mathbf{x}_g^1 &= \sum_{k=1}^{n_s^c} N_k^1 \mathbf{x}_k^1 = \sum_{i=1}^3 \tilde{N}_i \mathbf{x}_{clip} \\ \mathbf{x}_g^2 &= \sum_{i=1}^{n_m^c} N_i^2 \mathbf{x}_i^2 = \sum_{i=1}^3 \tilde{N}_i \mathbf{x}_{clip} \end{aligned}$$

$$(4.112b) \quad \begin{aligned} \mathbf{RHS}_1 &= \mathbf{x}_g^1 \\ \mathbf{RHS}_2 &= \mathbf{x}_g^2 \end{aligned}$$

$$(4.112c) \quad \mathbf{LHS}_i \begin{bmatrix} \xi_i \\ \eta_i \end{bmatrix} = \mathbf{RHS}_i, i = 1, 2$$

Where \mathbf{LHS}_i corresponds with the *Jacobian*:

$$(4.112d) \quad \mathbf{LHS}_i = J^i = \begin{bmatrix} \sum_{k=1}^{n_s^c} N_{k,\xi}^i \mathbf{x}_k^i \\ \sum_{k=1}^{n_m^c} N_{k,\eta}^i \mathbf{x}_k^i \end{bmatrix}, i = 1, 2$$

With this we can define the equivalent expressions for the derivatives of the local coordinates (ξ and η), we redefine the **RHS** (4.113a), and using the same **LHS** (J) (4.112d), we can solve (4.113b) the new system of equations in order to obtain our target derivatives.

$$(4.113a) \quad \begin{aligned} \mathbf{RHS}_1 &= \Delta \mathbf{x}_g^1 = \sum_{i=1}^3 \tilde{N}_i \Delta \mathbf{x}_{clip} \\ \mathbf{RHS}_2 &= \Delta \mathbf{x}_g^2 = \sum_{i=1}^3 \tilde{N}_i \Delta \mathbf{x}_{clip} \end{aligned}$$

$$(4.113b) \quad \begin{bmatrix} \Delta \xi_i \\ \Delta \eta_i \end{bmatrix} = \mathbf{LHS}_i^{-1} \mathbf{RHS}_i, i = 1, 2$$

4.6.2.2.3 Convergence study :

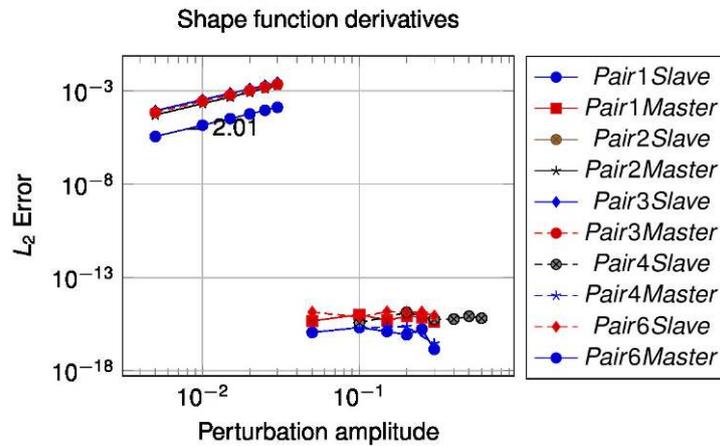


Figure 4.106: Convergence plot for the shape function derivatives for the 3D linear triangle

The convergence plot of the shape function derivatives for the 6 different geometrical pairs of the 3D linear triangles from Table 4.22 can be found in the Figure 4.106.

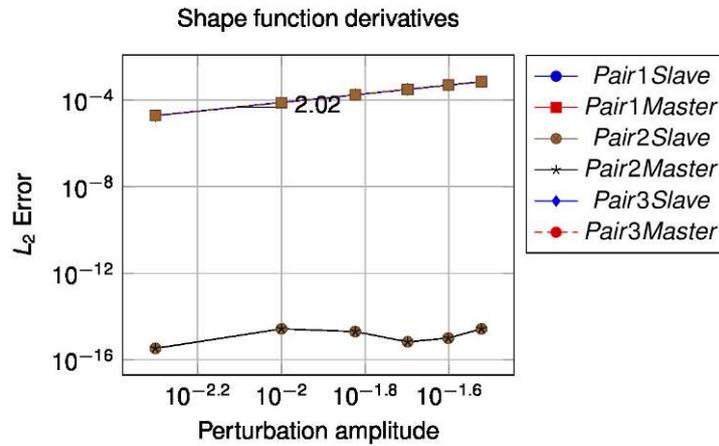


Figure 4.107: Convergence plot for the shape function derivatives for the 3D bilinear quadrilateral

The convergence plot of the shape function derivatives for the 3 different geometrical pairs of the 3D bilinear quadrilaterals from Table 4.24 can be found in the Figure 4.107.

4.6.2.3 Dual shape functions

4.6.2.3.1 Theory :

The procedure to follow is identical to the one presented for the 2D case in 4.6.1.3. Dual shape functions. No additional modification is needed, except, of course, the corresponding computation on the integration triangles, which means of the J_{clip} (4.6.2.1. Jacobians) and the use of the shape functions and derivatives already presented on the previous section 4.6.2.2. Shape functions. Then we proceed with the convergence study.

4.6.2.3.2 Convergence study :

The convergence plot of the dual shape function derivatives for the 6 different geometrical pairs of the 3D linear triangles from Table 4.22 can be found in the Figure 4.108.

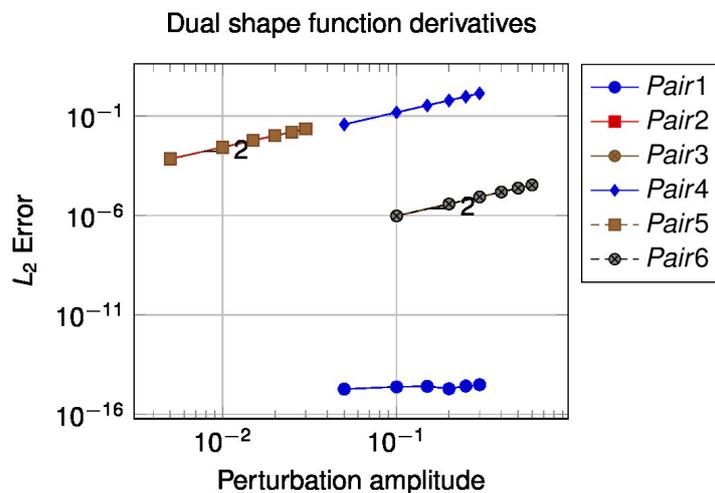


Figure 4.108: Convergence plot for the dual shape function derivatives for the 3D linear triangle

The convergence plot of the dual shape function derivatives for the 3 different geometrical pairs of the 3D bilinear quadrilaterals from Table 4.24 can be found in the Figure 4.109.

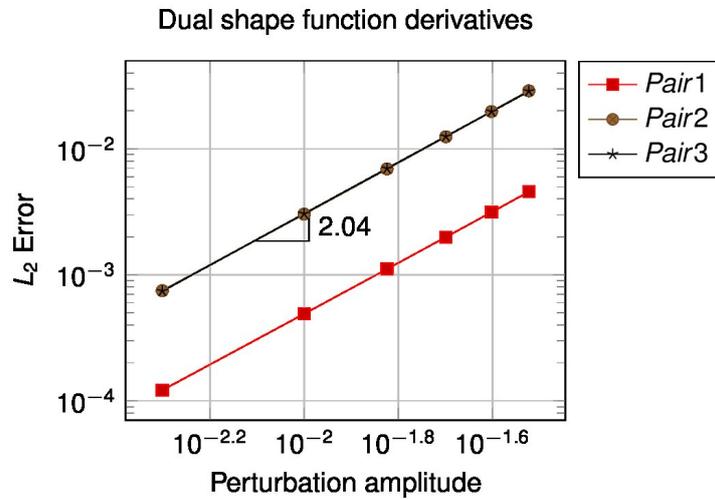


Figure 4.109: Convergence plot for the dual shape function derivatives for the 3D bilinear quadrilateral

4.6.2.4 Normal and tangent vectors

4.6.2.4.1 Theory :

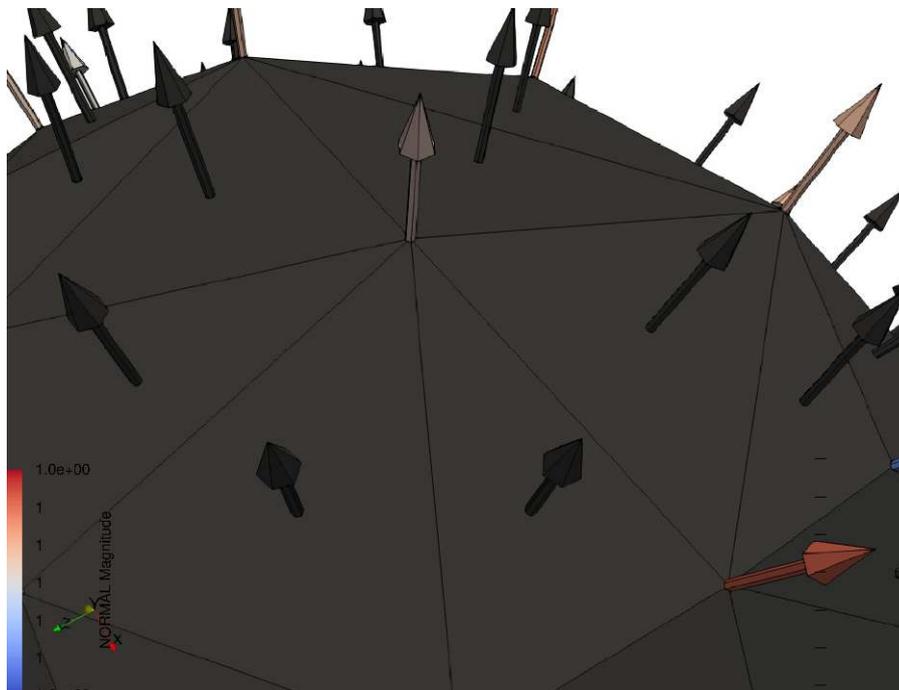


Figure 4.110: Normal average for a 3D geometry

As previously introduced, the tangent vector is taken as the complementary direction to the normal (4.114a), then in order to define its derivative we just need to compute the normal derivative, then the computation of the tangent derivative will be automatically taken into account with the AD procedure. The definition of the normal for 3D surface geometries corresponds with (4.114b), where can be defined as the cross product between the rows of the Jacobian matrix. Then we compute the unit normal the same way we do for the 2d case (4.101b).

The average normal behaves exactly the same to the definition presented for the 2D case (4.101c), the main

difference in the practical case came with the fact that the number of neighbours in 3D is not bounded, see Figure 4.110, in contrast with the 2D case where there are only two potential neighbours.

$$(4.114a) \quad \boldsymbol{\tau} = \mathbf{I} - \mathbf{n} \times \mathbf{n}$$

$$(4.114b) \quad \mathbf{n}_{area} = \mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}$$

Then in order to evaluate Δ for the normal on 3D surfaces we can apply the chain rule to the previous definition on (4.114b) and (4.101b). The resulting derivative (4.115) results as an addition.

$$(4.115a) \quad \Delta \mathbf{n} = \frac{\Delta \mathbf{n}_{area} \|\mathbf{n}_{area}\| - \mathbf{n}_{area} \Delta \|\mathbf{n}_{area}\|}{\|\mathbf{n}_{area}\|^2} = \frac{\Delta (\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}) \|\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}\| - (\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}) \Delta \|\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}\|}{\|\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}\|^2}$$

Where the first derivative is:

$$(4.115b) \quad \Delta (\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}) = \left(\sum_{k=1}^{n_s^c} N_{k,\xi} \Delta \mathbf{x}_k \right) \times \left(\sum_{k=1}^{n_s^c} N_{k,\eta} \mathbf{x}_k \right) + \left(\sum_{k=1}^{n_s^c} N_{k,\xi} \mathbf{x}_k \right) \times \left(\sum_{k=1}^{n_s^c} N_{k,\eta} \Delta \mathbf{x}_k \right)$$

And the second term:

$$(4.115c) \quad \Delta \|\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}\| = \frac{(\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}) \Delta (\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta})}{\|\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}\|} = \mathbf{n} \Delta (\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta})$$

Finally, in order to obtain the derivative of the average normal we proceed following the same equation presented for the 2D case (4.103). The normal derivative for the **GP** also behaves the same way as in 2D (4.104).

4.6.2.4.2 Convergence study :

The convergence plot of the normal vector derivatives for the 6 different geometrical pairs of the 3D linear triangles from Table 4.22 can be found in the Figure 4.111. We can appreciate that the convergence is greater than cubic.

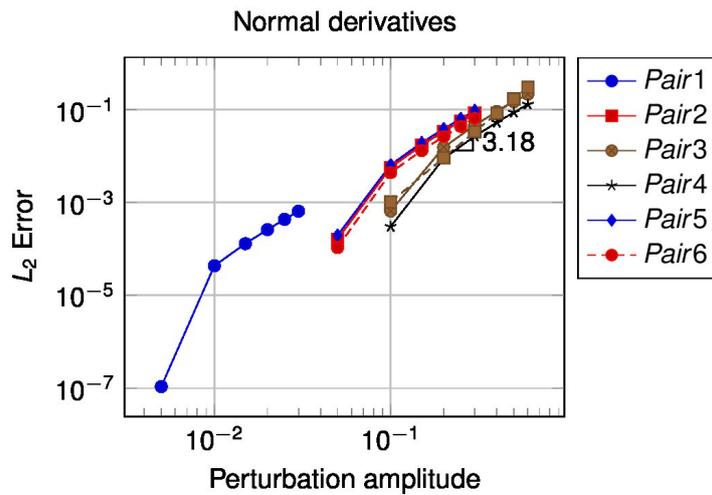


Figure 4.111: Convergence plot for the normal vector derivatives for the 3D linear triangle

The convergence plot of the normal vector derivatives for the 3 different geometrical pairs of the 3D bilinear quadrilaterals from Table 4.24 can be found in the Figure 4.112.

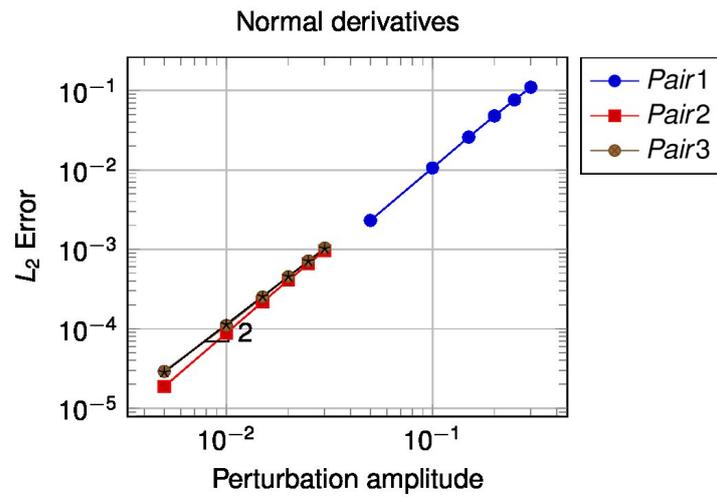


Figure 4.112: Convergence plot for the normal vector derivatives for the 3D bilinear quadrilateral

Bibliography

Books

- [BookAnd01] *From Convexity to Nonconvexity*. Anders Klarbring (auth.) R. P. Gilbert P. D. Panagiotopoulos P. M. Pardalos (eds.) Lars-Erik Andersson. Springer US. 1st ed. 2001.
- [BookBAL03] *Friction & flow stress in forming & cutting*. Philippe Boisse, Taylan Altan, and Kees van Luttervelt. Sterling, VA: Kogan Page. 2003.
- [BookBBT01] *The friction and lubrication of solids*. Frank Philip Bowden, Frank Philip Bowden, and David Tabor. Oxford university press. 2001.
- [BookBel+14] *Nonlinear Finite Elements for Continua and Structures*. Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. Wiley. 2nd ed. 2014.
- [BookBHS07] *Encyclopedia of Computational Mechanics (3 Volume Set)*. René de Borst, T.J.R. Hughes, and Erwin Stein. Wiley. 1st ed. 2007.
- [BookBou85] *Application des potentiels à l'étude de l'équilibre et du mouvement des solides élastiques*. Joseph Boussinesq. Gauthier-Villars, Imprimeur-Libraire. 1885.
- [BookEbe99] *Dynamic collision detection using oriented bounding boxes*. David Eberly. 1999.
- [BookEri04] *Real-time collision detection*. Christer Ericson. CRC Press. 2004.
- [BookFSS12] *OpenDSA: a creative commons active-ebook (abstract only)*. Eric Fouh, Maoyuan Sun, and Clifford Shaffer. 2012. Pp. 721–721.
- [BookHBC09] *Isogeometric Analysis: Toward Integration of CAD and FEA*. T.J.R. Hughes, Yuri Bazilevs, and J. Austin Cottrell. Wiley. 1st ed. 2009.
- [BookHug+14] *Computer graphics: principles and practice*. John F Hughes, Andries Van Dam, James D Foley, Morgan McGuire, Steven K Feiner, and David F Sklar. Pearson Education. 2014.
- [BookKO88] *Contact problems in elasticity: a study of variational inequalities and finite element methods*. Noboru Kikuchi and John Tinsley Oden. siam. 1988.
- [BookLau10] *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Tod A. Laursen. Springer. 2010.
- [BookMau13] *Contact, adhesion and rupture of elastic solids*. Daniel Maugis. Springer Science & Business Media. 2013.
- [BookPhD06] *R-Trees: Theory and Applications*. Alexandros Nanopoulos PhD Apostolos N. Papadopoulos PhD Yannis Theodoridis PhD (auth.) Yannis Manolopoulos PhD. Springer-Verlag London. 1st ed. 2006.
- [BookPop10] *Contact Mechanics and Friction: Physical Principles and Applications*. Valentin L. Popov. Springer-Verlag Berlin Heidelberg. 1st ed. 2010.
- [BookSE02] *Geometric tools for computer graphics*. Philip Schneider and David H Eberly. Elsevier. 2002.
- [BookSK13] *Computational Contact Mechanics: Geometrically Exact Theory for Arbitrary Shaped Bodies*. Karl Schweizerhof and Alexander Konyukhov. Springer-Verlag Berlin Heidelberg. 1st ed. 2013.
- [BookTos05] *Domain Decomposition Methods — Algorithms and Theory*. Andrea Toselli Olof B. Widlund (auth.) Springer-Verlag Berlin Heidelberg. 1st ed. 2005.

- [BookWoh01] *Discretization Methods and Iterative Solvers Based on Domain Decomposition*. Barbara I. Wohlmuth. Springer-Verlag Berlin Heidelberg. 1st ed. 2001.
- [BookWP99] *New Developments in Contact Problems*. Peter Wriggers and Panagiotis Panagiotopoulos. Springer-Verlag Wien. 1st ed. 1999.
- [BookWri06] *Computational Contact Mechanics*. Peter Wriggers. Springer. 2nd. 2006.
- [BookYas13] *Numerical Methods in Contact Mechanics*. Vladislav A. Yastrebov. Wiley-ISTE. 1st ed. 2013.
- [BookZZT13] *The Finite Element Method: its Basis and Fundamentals*. O. C. Zienkiewicz, J.Z. Zhu, and Robert L. Taylor. Butterworth-Heinemann. 7th ed. 2013.

Articles

- [ArtAC91] “A mixed formulation for frictional contact problems prone to Newton like solution methods”. P. Alart and A. Curnier. In: *Computer Methods in Applied Mechanics and Engineering*. No. 3, Vol. 92, 1991, pp. 353–375. DOI: [10.1016/0045-7825\(91\)90022-x](https://doi.org/10.1016/0045-7825(91)90022-x).
- [ArtAnt17] “A further analysis on the analogy between friction and plasticity in Solid Mechanics”. Nicolas Antoni. In: *International Journal of Engineering Science*. Vol. 121, 2017, pp. 34–51. Elsevier. DOI: [10.1016/j.ijengsci.2017.08.012](https://doi.org/10.1016/j.ijengsci.2017.08.012).
- [ArtBar08] “Adhesive elastic contacts – JKR and more”. Etienne Barthel. In: *Journal of Physics D: Applied Physics*. Vol. 41, 2008, p. 163001. IOP Publishing. DOI: [10.1088/0022-3727/41/16/163001](https://doi.org/10.1088/0022-3727/41/16/163001).
- [ArtBen+11] “A large deformation, rotation-free, isogeometric shell”. D.J. Benson, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes. In: *Computer Methods in Applied Mechanics and Engineering*. No. 13, Vol. 200, 2011, pp. 1367–1378. Elsevier. DOI: [10.1016/j.cma.2010.12.003](https://doi.org/10.1016/j.cma.2010.12.003).
- [ArtBMP] “The adapted augmented Lagrangian method: a new method for the resolution of the mechanical frictional contact problem”. Philippe Busetta, Daniel Marceau, and Jean-Philippe Ponthot. In: *Computational Mechanics*. No. 2, , pp. 259–275. DOI: [10.1007/s00466-011-0644-z](https://doi.org/10.1007/s00466-011-0644-z).
- [ArtBT39] “The area of contact between stationary and moving surfaces”. Frank Philip Bowden and David Tabor. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*. No. 938, Vol. 169, 1939, pp. 391–413. The Royal Society London.
- [ArtCai+14] “Collision detection using axis aligned bounding boxes”. Panpan Cai, Chandrasekaran Indhumathi, Yiyu Cai, Jianmin Zheng, Yi Gong, Teng Sam Lim, and Peng Wong. In: . 2014, pp. 1–14. Springer. DOI: [10.1007/978-981-4560-32-0_1](https://doi.org/10.1007/978-981-4560-32-0_1).
- [ArtCam15] “Computational modeling and simulation of rupture of membranes and thin films”. Eduardo MB Campello. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering*. No. 6, Vol. 37, 2015, pp. 1793–1809. Springer. DOI: [10.1007/s40430-014-0273-5](https://doi.org/10.1007/s40430-014-0273-5).
- [ArtCC12] “An augmented lagrangian method to solve three-dimensional nonlinear contact problems”. FJ Calavalieri and Alberto Cardona. In: *Latin American applied research*. No. 3, Vol. 42, 2012, pp. 281–289. SciELO Argentina.
- [ArtCC13a] “Three-dimensional numerical solution for wear prediction using a mortar contact algorithm”. Federico José Cavalieri and Alberto Cardona. In: *International Journal for Numerical Methods in Engineering*. No. 8, Vol. 96, 2013, pp. 467–486. Wiley Online Library. DOI: [10.1002/nme.4556](https://doi.org/10.1002/nme.4556).
- [ArtCC13b] “An augmented Lagrangian technique combined with a mortar algorithm for modelling mechanical contact problems”. FJ Cavalieri and A Cardona. In: *International Journal for Numerical Methods in Engineering*. No. 4, Vol. 93, 2013, pp. 420–442. Wiley Online Library. DOI: [10.1002/nme.4391](https://doi.org/10.1002/nme.4391).
- [ArtCC15] “Numerical solution of frictional contact problems based on a mortar algorithm with an augmented Lagrangian technique”. F. J. Cavalieri and A. Cardona. In: *Multibody System Dynamics*. No. 4, Vol. 35, 2015, pp. 353–375. DOI: [10.1007/s11044-015-9449-8](https://doi.org/10.1007/s11044-015-9449-8).

- [ArtCFC12] “Un algoritmo de contacto mortar aplicable a problemas tridimensionales”. F.J. Cavaliere, V.D. Fachinotti, and A. Cardona. In: *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*. No. 2, Vol. 28, 2012, pp. 80–92. DOI: [10.1016/j.rimni.2012.03.001](https://doi.org/10.1016/j.rimni.2012.03.001).
- [ArtCho+17] “An overview of recent results on Nitsche’s method for contact problems”. Franz Chouly, Mathieu Fabre, Patrick Hild, Rabii Mlika, Jerome Pousin, and Yves Renard. In: . 2017, pp. 93–141. Springer.
- [ArtCou85] “The theory of simple machines”. CA Coulomb. In: *Mem. Math. Phys. Acad. Sci.* No. 161-331, Vol. 10, 1785, p. 4.
- [ArtDB15] “A literature review of bounding volumes hierarchy focused on collision detection”. Simena Dinas and José M Bañón. In: *Ingeniería y competitividad*. No. 1, Vol. 17, 2015, pp. 49–62. Universidad del Vall.
- [ArtDem+12] “Computational comparison of the bending behavior of aortic stent-grafts”. Nicolas Demanget, Stéphane Avril, Pierre Badel, Laurent Orgéas, Christian Geindreau, Jean-Noël Albertini, and Jean-Pierre Favre. In: *Journal of the mechanical behavior of biomedical materials*. No. 1, Vol. 5, 2012, pp. 272–282. Elsevier. DOI: [10.1016/j.jmbbm.2011.09.006](https://doi.org/10.1016/j.jmbbm.2011.09.006).
- [ArtDFS98] “Solution of coercive and semicoercive contact problems by FETI domain decomposition”. Zdenek Dostál, Ana Friedlander, and Sandra A Santos. In: *Contemporary Mathematics*. Vol. 218, 1998, pp. 82–93. Providence, RI: American Mathematical Society. DOI: [10.1090/conm/218/03003](https://doi.org/10.1090/conm/218/03003).
- [ArtDon99] “A simple benchmark problem to test frictional contact”. CY Dong. In: *Computer methods in applied mechanics and engineering*. No. 1-2, Vol. 177, 1999, pp. 153–162. Elsevier. DOI: [10.1016/s0045-7825\(98\)00377-6](https://doi.org/10.1016/s0045-7825(98)00377-6).
- [ArtDPS] “A frictional mortar contact approach for the analysis of large inelastic deformation problems”. T Doca, FM Andrade Pires, and JMA Cesar de Sa. In: *International Journal of Solids and Structures*. No. 9, , pp. 1697–1715. Elsevier. DOI: [10.1016/j.ijsolstr.2014.01.013](https://doi.org/10.1016/j.ijsolstr.2014.01.013).
- [ArtDSB15] “High-order mortar-based element applied to nonlinear analysis of structural contact mechanics”. APC Dias, AL Serpa, and ML Bittencourt. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 294, 2015, pp. 19–55. Elsevier. DOI: [10.1016/j.cma.2015.05.013](https://doi.org/10.1016/j.cma.2015.05.013).
- [ArtEul50a] “Sur la diminution de la resistance du frottement”. Leonhard Euler. In: *Mémoires de l’académie des sciences de Berlin*. 1750, pp. 133–148.
- [ArtEul50b] “Sur le frottement des corps solides”. Leonhard Euler. In: *Mémoires de l’Académie des Sciences de Berlin*. 1750, pp. 122–132.
- [ArtFF03] “Collision Detection for Virtual Prototyping Environments”. Mauro Figueiredo and T Fernando. In: . 2003, pp. 157–166.
- [ArtFic63] “Sul problema elastostatico di Signorini con ambigue condizioni al contorno”. Gaetano Fichera. In: *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur.(8)*. Vol. 34, 1963, pp. 138–142.
- [ArtFig10] “S-cd: surface collision detection toolkit for virtual prototyping”. Mauro Figueiredo. In: *IADIS International Journal on Computer Science and Information Systems*. Vol. 5, 2010, pp. 72–86.
- [ArtFW06] “Mortar based frictional contact formulation for higher order interpolations using the moving friction cone”. Kathrin A Fischer and Peter Wriggers. In: *Computer methods in applied mechanics and engineering*. No. 37-40, Vol. 195, 2006, pp. 5020–5036. Elsevier. DOI: [10.1016/j.cma.2005.09.025](https://doi.org/10.1016/j.cma.2005.09.025).
- [ArtFZ75] “A note on numerical computation of elastic contact problems”. A Francavilla and OC Zienkiewicz. In: *International Journal for Numerical Methods in Engineering*. No. 4, Vol. 9, 1975, pp. 913–924. Wiley Online Library. DOI: [10.1002/nme.1620090410](https://doi.org/10.1002/nme.1620090410).
- [ArtHar+09] “A contact domain method for large deformation frictional contact problems. Part 2: Numerical aspects”. S. Hartmann, J. Oliver, R. Weyler, J.C. Cante, and J.A. Hernández. In: *Computer Methods in Applied Mechanics and Engineering*. No. 33–36, Vol. 198, 2009, pp. 2607–2631. DOI: [10.1016/j.cma.2009.03.009](https://doi.org/10.1016/j.cma.2009.03.009).

- [ArtHar+10] “A 3D frictionless contact domain method for large deformation problems”. S Hartmann, R Weyler, J Oliver, JC Cante, and JA Hernández. In: *Computer Modeling in Engineering and Sciences (CMES)*. No. 3, Vol. 55, 2010, p. 211.
- [ArtHer82] “Über die berührung fester elastische Körper und über die Harte”. H Hertz. In: *Verhandlungen des Vereins zur Beförderung des Gewerbefleißes*. 1882,
- [ArtHTK77] “A finite element method for large displacement contact and impact problems”. TJR Hughes, RL Taylor, and W Kanoknukulchai. In: *Formulations and Computational Algorithms in FE Analysis*. 1977, pp. 468–495. MIT Press: Boston.
- [ArtHWP18] “A truly variationally consistent and symmetric mortar-based contact formulation for finite deformation solid mechanics”. Michael Hiermeier, Wolfgang A. Wall, and Alexander Popp. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 342, 2018, pp. 532–560. DOI: [10.1016/j.cma.2018.07.020](https://doi.org/10.1016/j.cma.2018.07.020).
- [ArtJKR71] “Surface energy and the contact of elastic solids”. Kenneth Langstreth Johnson, Kevin Kendall, and AD Roberts. In: *Proceedings of the royal society of London. A. mathematical and physical sciences*. No. 1558, Vol. 324, 1971, pp. 301–313. The Royal Society London.
- [ArtKar39] “Minima of functions of several variables with inequalities as side constraints”. William Karush. In: *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*. 1939,
- [ArtKor94] “Monotone multigrid methods for elliptic variational inequalities I”. Ralf Kornhuber. In: *Numerische Mathematik*. No. 2, Vol. 69, 1994, pp. 167–184. Springer. DOI: [10.1007/s002110050178](https://doi.org/10.1007/s002110050178).
- [ArtKra68] “Friction and wear”. IV Kragelsky. In: *Mechanical Engineering, Moscow*. 1968, Elsevier.
- [ArtLWZ12a] “A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method”. L. de Lorenzis, Peter Wriggers, and G. Zavarise. In: *Computational Mechanics*. No. 1, Vol. 49, 2012, pp. 1–20. Springer. DOI: [10.1007/s00466-011-0623-4](https://doi.org/10.1007/s00466-011-0623-4).
- [ArtLWZ12b] “A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method”. L. De Lorenzis, Peter Wriggers, and G. Zavarise. In: *Computational Mechanics*. No. 1, Vol. 49, 2012, pp. 1–20. Springer-Verlag. DOI: [10.1007/s00466-011-0623-4](https://doi.org/10.1007/s00466-011-0623-4).
- [ArtMor73] “Systèmes élastoplastiques de liberté finie”. JEAN JACQUES Moreau. In: *Travaux du Séminaire d'Analyse Convexe*. Vol. 3, 1973, p. 33.
- [ArtNag05] “Simple local interpolation of surfaces using normal vectors”. Takashi Nagata. In: *Computer Aided Geometric Design*. No. 4, Vol. 22, 2005, pp. 327–347. DOI: [10.1016/j.cagd.2005.01.004](https://doi.org/10.1016/j.cagd.2005.01.004).
- [ArtNet+14] “Applying Nagata patches to smooth discretized surfaces used in 3D frictional contact problems”. D. M. Neto, M. C. Oliveira, L. F. Menezes, and J. L. Alves. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 271, 2014, pp. 296–320. Elsevier. DOI: [10.1016/j.cma.2013.12.008](https://doi.org/10.1016/j.cma.2013.12.008).
- [ArtNit71] “Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind”. Joachim Nitsche. In: . No. 1, Vol. 36, 1971, pp. 9–15.
- [ArtNOM17] “Surface Smoothing Procedures in Computational Contact Mechanics”. D. M. Neto, M. C. Oliveira, and L. F. Menezes. In: *Archives of Computational Methods in Engineering*. No. 1, Vol. 24, 2017, pp. 37–87. DOI: [10.1007/s11831-015-9159-7](https://doi.org/10.1007/s11831-015-9159-7).
- [ArtOde81] “Exterior penalty methods for contact problems in elasticity”. JT Oden. In: . 1981, pp. 655–665. Springer. DOI: [10.1007/978-3-642-81589-8_33](https://doi.org/10.1007/978-3-642-81589-8_33).
- [ArtOli+09a] “A contact domain method for large deformation frictional contact problems. Part 1: Theoretical basis”. J. Oliver, S. Hartmann, J. C. Cante, R. Weyler, and J. A. Hernández. In: *Computer Methods in Applied Mechanics and Engineering*. No. 33–36, Vol. 198, 2009, pp. 2591–2606. DOI: [10.1016/j.cma.2009.03.006](https://doi.org/10.1016/j.cma.2009.03.006).
- [ArtOli+09b] “A New Approach in Computational Contact Mechanics: The Contact Domain Method”. X Oliver, S Hartmann, JC Cante, R Weyler, and JA Hernández. In: *CIMNE, Barcelona*. 2009,

- [ArtPGW09] "A finite deformation mortar contact formulation using a primal–dual active set strategy". Alexander Popp, Michael W. Gee, and Wolfgang A. Wall. In: *International Journal for Numerical Methods in Engineering*. No. 11, Vol. 79, 2009, pp. 1354–1391. John Wiley & Sons, Ltd. DOI: [10.1002/nme.2614](https://doi.org/10.1002/nme.2614).
- [ArtPL04] "A mortar segment-to-segment contact method for large deformation solid mechanics". Michael A. Puso and Tod A. Laursen. In: *Computer Methods in Applied Mechanics and Engineering*. No. 6–8, Vol. 193, 2004, pp. 601–629. Elsevier. DOI: [10.1016/j.cma.2003.10.010](https://doi.org/10.1016/j.cma.2003.10.010).
- [ArtPop+10] "A dual mortar approach for 3D finite deformation contact with consistent linearization". Alexander Popp, Markus Gitterle, Michael W. Gee, and Wolfgang A. Wall. In: *International Journal for Numerical Methods in Engineering*. No. 11, Vol. 83, 2010, pp. 1428–1465. John Wiley & Sons, Ltd. DOI: [10.1002/nme.2866](https://doi.org/10.1002/nme.2866).
- [ArtPP15] "The research works of Coulomb and Amontons and generalized laws of friction". Elena Popova and Valentin L Popov. In: *Friction*. No. 2, Vol. 3, 2015, pp. 183–190. Springer. DOI: [10.1007/s40544-015-0074-6](https://doi.org/10.1007/s40544-015-0074-6).
- [ArtPPS03] "Vibration damping by friction forces: theory and applications". Karl Popp, Lars Panning, and Walter Sextro. In: *Modal Analysis*. No. 3-4, Vol. 9, 2003, pp. 419–448. Sage Publications Sage CA: Thousand Oaks, CA. DOI: [10.1177/107754603030780](https://doi.org/10.1177/107754603030780).
- [ArtPR15a] "An unconstrained integral approximation of large sliding frictional contact between deformable solids". Konstantinos Poullos and Yves Renard. In: *Computers & Structures*. Vol. 153, 2015, pp. 75–90. DOI: [10.1016/j.compstruc.2015.02.027](https://doi.org/10.1016/j.compstruc.2015.02.027).
- [ArtPR15b] "An unconstrained integral approximation of large sliding frictional contact between deformable solids". Konstantinos Poullos and Yves Renard. In: *Computers & Structures*. Vol. 153, 2015, pp. 75–90. Elsevier. DOI: [10.1016/j.compstruc.2015.02.027](https://doi.org/10.1016/j.compstruc.2015.02.027).
- [ArtPT92] "A mixed formulation for the finite element solution of contact problems". Panayiotis Papadopoulos and Robert L Taylor. In: *Computer Methods in Applied Mechanics and Engineering*. No. 3, Vol. 94, 1992, pp. 373–389. Elsevier. DOI: [10.1016/0045-7825\(92\)90061-n](https://doi.org/10.1016/0045-7825(92)90061-n).
- [ArtRao16] "Art of Modeling in Contact Mechanics". Michel Raous. In: . Vol. 570, 2016, Springer. DOI: [10.1007/978-3-319-40256-7_4](https://doi.org/10.1007/978-3-319-40256-7_4).
- [ArtRKC02] "Fast Continuous Collision Detection between Rigid Bodies". Stephane Redon, Abderrahmane Kheddar, and Sabine Coquillart. In: *Computer Graphics Forum*. No. 3, Vol. 21, 2002, pp. 279–287. DOI: [10.1111/1467-8659.t01-1-00587](https://doi.org/10.1111/1467-8659.t01-1-00587).
- [ArtRom11] "Advanced methods for robot-environment interaction towards an industrial robot aware of its volume". Fabrizio Romanelli. In: *Journal of Robotics*. Vol. 2011, 2011, Hindawi. DOI: [10.1155/2011/389158](https://doi.org/10.1155/2011/389158).
- [ArtSig33] "Sopra alcune questioni di elastostatica". Antonio Signorini. In: *Atti della Societa Italiana per il Progresso delle Scienze*. No. li, Vol. 21, 1933, pp. 143–148.
- [ArtSN17] "Woven fabrics computational simulation using beam-to-beam contacts formulation". Mauro Takayama Saito and Alfredo Gay Neto. In: *Revista Interdisciplinar De Pesquisa Em Engenharia*. No. 22, Vol. 2, 2017, pp. 09–25.
- [ArtSpe] "The Hertz contact problem with finite friction". DA Spence. In: *Journal of elasticity*. No. 3-4, , pp. 297–319. Springer. DOI: [10.1007/bf00126993](https://doi.org/10.1007/bf00126993).
- [ArtSWP18] "A computational approach for thermo-elasto-plastic frictional contact based on a monolithic formulation using non-smooth nonlinear complementarity functions". Alexander Seitz, Wolfgang A Wall, and Alexander Popp. In: *Advanced Modeling and Simulation in Engineering Sciences*. No. 1, Vol. 5, 2018, p. 5. Springer. DOI: [10.1186/s40323-018-0098-3](https://doi.org/10.1186/s40323-018-0098-3).
- [ArtSWT85] "A perturbed Lagrangian formulation for the finite element solution of contact problems". Juan C Simo, Peter Wriggers, and Robert L Taylor. In: *Computer methods in applied mechanics and engineering*. No. 2, Vol. 50, 1985, pp. 163–180. Elsevier. DOI: [10.1016/0045-7825\(85\)90088-x](https://doi.org/10.1016/0045-7825(85)90088-x).

- [ArtTC03] "A model of adhesion coupled to contact and friction". C. Talon and A. Curnier. In: *European Journal of Mechanics - A/Solids*. No. 4, Vol. 22, 2003, pp. 545–565. DOI: [10.1016/s0997-7538\(03\)00046-9](https://doi.org/10.1016/s0997-7538(03)00046-9).
- [ArtTK14] "Real-Time Hybrid Virtuality for Prevention of Excavation Related Utility Strikes". Sanat Talmaki and Vineet R. Kamat. In: *Journal of Computing in Civil Engineering*. No. 3, Vol. 28, 2014, p. 04014001. DOI: [10.1061/\(asce\)cp.1943-5487.0000269](https://doi.org/10.1061/(asce)cp.1943-5487.0000269).
- [ArtTP] "On a patch test for contact problems in two dimensions". Robert L. Taylor and Panagiotis Panagiotopoulos. In: . ,
- [ArtWI93] "On the treatment of nonlinear unilateral contact problems". P. Wriggers and M. Imhof. In: *Archive of Applied Mechanics*. No. 2, Vol. 63, 1993, pp. 116–129. Springer. DOI: [10.1007/bf00788917](https://doi.org/10.1007/bf00788917).
- [ArtWie+18] "Algebraic multigrid methods for dual mortar finite element formulations in contact mechanics". T. A. Wiesner, A. Popp, M. W. Gee, and W. A. Wall. In: *International Journal for Numerical Methods in Engineering*. No. 4, Vol. 114, 2018, pp. 399–430. DOI: [10.1002/nme.5748](https://doi.org/10.1002/nme.5748).
- [ArtWoh02] "A Comparison of Dual Lagrange Multiplier Spaces for Mortar Finite Element Discretizations". Barbara I. Wohlmuth. In: *ESAIM: Mathematical Modelling and Numerical Analysis*. No. 6, Vol. 36, 2002, pp. 995–1012. EDP Sciences. DOI: [10.1051/m2an:2003002](https://doi.org/10.1051/m2an:2003002).
- [ArtWZ13] "Hertz Theory: Contact of Spherical Surfaces". Q. Jane Wang and Dong Zhu. In: . 2013, pp. 1654–1662. Springer US. Boston, MA. DOI: [10.1007/978-0-387-92897-5_492](https://doi.org/10.1007/978-0-387-92897-5_492).
- [ArtYCF11] "A local contact detection technique for very large contact and self-contact problems: sequential and parallel implementations". VA Yastrebov, Georges Cailletaud, and Frédéric Feyel. In: . 2011, pp. 227–251. Springer. DOI: [10.1007/978-3-642-22167-5_13](https://doi.org/10.1007/978-3-642-22167-5_13).
- [ArtYL08] "A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations". Bin Yang and TodA. Laursen. In: *Computational Mechanics*. No. 2, Vol. 41, 2008, pp. 189–205. Springer-Verlag. DOI: [10.1007/s00466-006-0116-z](https://doi.org/10.1007/s00466-006-0116-z).
- [ArtYLM] "Two dimensional mortar contact methods for large deformation frictional sliding". Bin Yang, Tod A. Laursen, and Xiaonong Meng. In: *International Journal for Numerical Methods in Engineering*. No. 9, , pp. 1183–1225. Chichester, New York, Wiley [etc.] 1969-. DOI: [10.1002/nme.1222](https://doi.org/10.1002/nme.1222).
- [ArtZhu12] "Tutorial on hertz contact stress". Xiaoyin Zhu. In: . Vol. 521, 2012, pp. 1–8.
- [ArtZK12] "k-IOS: Intersection of spheres for efficient proximity query". Xinyu Zhang and Young J. Kim. In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 354–359. DOI: [10.1109/icra.2012.6224889](https://doi.org/10.1109/icra.2012.6224889).
- [ArtZL09] "A modified node-to-segment algorithm passing the contact patch test". Giorgio Zavarise and Laura De Lorenzis. In: *International journal for numerical methods in engineering*. No. 4, Vol. 79, 2009, p. 379. DOI: [10.1002/nme.2559](https://doi.org/10.1002/nme.2559).

Ph.D.'s thesis

- [PhDBru08] "Contact analysis and overlapping domain decomposition methods for dynamic and nonlinear problems". Stephan BrunSSen. 2008.
- [PhDBus] "Modélisation et résolution du problème de contact mécanique et son application dans un contexte multiphysique". Philippe Bussetta.
- [PhDDoc] "Energy wear methods for dual-mortar contact analysis of frictional problems at finite inelastic strains". Thiago de Carvalho Rodrigues Doca.
- [PhDDro15] "Local mortar method for contact problems with high order non-matching meshes". Guillaume Drouet. 2015.
- [PhDGit12] "A dual mortar formulation for finite deformation frictional contact problems including wear and thermal coupling". Markus Gitterle. 2012.

- [PhDHam13] “Frictional mortar contact for finite deformation problems with synthetic contact kinematics”. Michael E. Hammer. 2013.
- [PhDKra01] “Monotone multigrid methods for Signorini’s problem with friction”. Rolf H Krause. 2001.
- [PhDPop12] “Mortar Methods for Computational Contact Mechanics and General Interface Problems”. Alexander Popp. 2012.
- [PhDYas11] “Computational contact mechanics: geometry, detection and numerical techniques”. Vladislav A. Yastrebov. 2011.

Online resources

- [OnlCho] *Collision Detection Using the Separating Axis Theorem*. URL: <https://gamedevelopment.tutsplus.com/tutorials/collision-detection-using-the-separating-axis-theorem-gamedev-169>.
- [OnlJac] *Jacobi elliptic functions*. URL: https://www.encyclopediaofmath.org/index.php/Jacobi_elliptic_functions.
- [OnlRtr] *Wikipedia. R-tree*. URL: <https://en.wikipedia.org/wiki/R-tree>.

Chapter 5

Plasticity

“As the water shapes itself to the vessel that contains it, so a wise man adapts himself to circumstances.”

Confucius
(551 - 479 BC, Chinese philosopher)

5.1 Introduction

The mathematical modelling of plasticity in solids and structures are essential for a good modulation of the forming processes[BookBan10]. As well as other industrial processes and applications, like the design of steel and concrete civil structures, or the understanding of the response of the soil or rocks in geotechnical problems[BookBHS07]. We can shortly introduce[BookNPO09] that in contrast with the elastic constitutive laws, the elasticplastic laws are path-dependent and dissipative[BookZTF14], this means that large part of the work expended in plastically deforming the material is irreversibly converted to other forms of energy, generally heat[BookHil98]. The stress depends on the entire history of the deformation[BookHS98], and cannot be written as a single-valued function of the strain; rather it can only be specified as a relation between rates of stress and strain[BookBel+14].

5.1.1 Historical outline

Plasticity is one of the most developed theories for describing material **NL** behaviour. The history of the studies in this scientific discipline, which involves mathematics, physics and engineering applications is rich. For a complete overview of the history of plasticity, we address directly to the overview presented on the book of Hill[BookHil98] and Zyczkowski[BookZyc81], much more complete than the one presented here.

Having said that, the first developments came from the *Coulomb*[ArtCou76] contributions in 1776, when he presented a work stating the dependence of the sliding resistance on a plane between two bodies, being a function of the adhesion and the frictional properties¹. This work, together with the later developments published by *Poncelet* and *Rankine*, was useful for the calculation of retaining walls.

We can identify as the firsts[BookWei] properly speaking works on plasticity the contributions of *Tresca*[BookTre65] in 1864 on the extrusion of metals, this was the later denominated *Tresca* yield criterion, which states that the metal yields when the maximal shear stress surpasses a threshold. During that time, the also-French mathematician *St. Venant*[ArtSai70] introduced basic constitutive relations for perfectly plastic materials, where it related the direction on the strain increase and the principal stresses. These developments were later extended to 3D by *Lévy*.

¹We discuss in the contact chapter 4. [Contact mechanics](#), the similarities between the frictional contact theory and the plasticity models.

In 1886 *Bauschinger*[ArtBau86] introduced the effect that now carries its name, which it consists in the fact that by deforming a metal in one direction until its elasticity limit has been exceeded, and then deforming it in the opposite direction, its proportionality limit in the latter direction is lower due to the material imperfections. It was before the **First World War (WWI)**, in 1913, when *Von Mises*[ArtMis13] introduced the known as J_2 theory. Later in 1924 *Prandtl*[ArtPra24] extended the *St. Venant* work for elastic-plastic behaviour, which *Reuss* extended later to 3D. It was not until 1928 when *Von Mises*[ArtMis28] extended his previous theory considering the yield function as a plastic potential in the incremental stress-strain relations of the flow theory. That very same year *Prandtl* tried to formulate general relations for hardening behaviour.

In the period between wars, and especially during the **WWII**, many relevant developments were done on this field. The war stimulated research through problems such as the calculation of the stresses in autofrettaged gun barrels and of the forces resisting a shot penetrating armour plate[BookHil98].

Additionally to this, it is particularly known and interesting to highlight the problems originated during the production of the *Liberty ships*[ArtZha16]. These were cargo ships built in the **United States (U.S.)** during **WWII** in order to provide goods and carry troops. The initial design was modified by the **U.S.** Maritime Commission to conform to make it quicker and cheaper to build, this new design consisted on replace part of the riveting operations with welding. This was due because riveting represented the 30% of the workforce costs. Unfortunately, no further studies were carried to understand to how welding would affect the structure[ArtGer12]. The rapid and massive construction requirements originated several issues that many notes and trigger the alarm about a study the reasons that induce these problems. Almost 1500 instances of significant brittle fractures were recorded (Figure 5.1). Usually this problem is studied as a fracture phenomenon, but studies from *Altstetter et al.*[ArtAA95] proposed an alternative approach that enhanced localised plasticity might nucleate either ductile fracture or quasi-cleavage. In any case, these problems lead deeper studies on the **NL** behaviour of the materials and benefited therefore the understanding of the plasticity.



Figure 5.1: *Liberty ships* suddenly broke in half while moored at the dock

So in 1949 *Prager*[ArtPra49] proposed a framework for the plastic constitutive relations for hardening materials with *smooth* yield functions. Two years later *Drucker* his material stability postulate. Together, they presented in 1952[ArtDP52] its criterion, which tries to represent the plastic deformation of soils. This is a pressure-dependent model that determines whether a material has exceeded the elastic limit. On 1953 *Koiter*[ArtKoi53] generalised the plastic stress-strain relations for *non-smooth* yield functions.

The following developments, due to the rise of the computational power were more focus on the research on the numerical analysis, especially after the 1970s, so it will be detailed in the next section 5.2.State of the Art in numerical plasticity.

5.2 State of the Art in numerical plasticity

5.2.1 Historical outline

We want to convey that the developments that will be shown in this work are within the classical theory, and we will not be innovative in this matter. The work done in this field was needed in order to be able to address the forming process problem. So the first works that can be considered as reference, the foundations for the numerical plasticity including infinitesimal theory, can be found in the book from the 1950s of *Hill*[BookHil98] and later in the article of *Koiter*[ArtKoi60].

Having said that, we can attribute[BookWei] to *Duvaut and Lions*[ArtDL76] the first study of the **Boundary Value Problem (BVP)** of elasto-plasticity, who defined the problem as variational inequality². Later *Johnson*[ArtJoh76] split

²See the Appendix D.Constrained optimisation problems for further detail. On this Appendix the inequality is treated in order to introduce the mathematical concept necessities to understand the **CCM**, but the theory can be applied on plasticity too.

this variational inequality in two steps, one to solve the steady state problem after removing the velocity and a second step to solve the velocity of the problem.

In 1985 *J.C. Simo* and *R.L. Taylor*[ArtST85], derived the consistent elasto-plastic tangent tensors. \mathbb{C}^{ep} . This result can be seen in the corresponding section at 5.5.3.Numerical implementation tangent constitutive tensor.

Later, incremental finite strain elasto-plasticity for stable states of equilibrium and the coercivity of internal energy is presented by using multiplicative decomposition of strain, *Frechet-derivatives* and the chain rule, yielding corresponding consistent tangent operators with the same formal structure as for linearised strains, published by *J.C. Simo* in 1988[ArtSim88a; ArtSim88b]. The theory related to this point is shown at 5.3.Finite strain elasto-plastic models. Additionally, in the range of large deformations, the original proposition for the multiplicative decomposition of the deformation gradient considered on this work can be attributed to *Lee and Liu*[ArtLL67].

In the last 20 years significant advances have been done, particularly on the development of finite deformation theories in order to solve practical and industrial problems which involve large strains. In order to applied the corresponding **NR** strategy, we need the proper linearisation of the problem. In general these problems require to work with discrete equations, where only a few cases as the J_2 flow theory can provide a closed-form solution. So general algorithms for the *return-mapping* algorithms are needed. Denominating these procedures as **Closest Point Projection Method (CPPM)**, complete monographs about this subject can be found in the work of *Lubliner*[BookLub08] and *Simo and Hughes*[BookHS98].

The main works we want to address for a deeper understanding for the reader, which analyze from a extended point of view and with a general purpose, are the books of *Neto*[BookNPO09], *Lubliner*[BookLub08], *Simo*[BookSim93] and *Simo and Hughes*[BookHS98].

5.2.2 Constitutive models

The following is a brief summary of the most commonly employed *yield criteria* (5.2.2.1.Yield criteria) and *yield surface hardening* (5.2.2.2.Yield surface hardening).

5.2.2.1 Yield criteria

The *discontinuity* or *yield criterion* is a scalar function of tensor arguments that defines the elastic domain. It is regularly presented in the form $\Phi(\boldsymbol{\sigma}, \mathbf{q})$, where $\boldsymbol{\sigma}$ is the *Cauchy* stress tensor, but can be represented with any stress tensor, and \mathbf{q} are the (*stress-like*) internal variables, grouped as "*back-stress*". The surface limits the stress state, in such a way that is not admissible any state outside the defined surface.

The following[BookNPO09; BookOll14] summarises some of the most common yield criteria used in engineering practice. We briefly summarise them, but for deeper understanding we recommend to attend to the literature.

- **Tresca yield criterion:** Originally presented in 1864[BookTre65]. It assumes that plastic yielding begins when the maximum shear stress reaches a critical value.
- **Barret de Saint Venant yield criterion:** *Barret de Saint Venant*, published in 1871[ArtSai70] a constitutive equation for inelastic of an elastically rigid, perfectly plastifying solid material in plane stress state with the hypothesis of isotropic deformations, stating that the main axes of strain coincide with the main axes of stress.
- **Von-Mises yield criterion:** According to it the plastic yielding begins when the J_2 stress deviator reaches a critical value. Where J_2 cam is defined as (5.1).

$$(5.1) \quad J_2 = \frac{1}{2} \text{tr} (\boldsymbol{\sigma}_{dev})^2$$

This is probably the most extended yield criterion, especially for metals.

- **Mohr-Coulomb yield criterion:** It is based on the assumption that the phenomenon of macroscopic plastic yielding is the result of frictional sliding between material particles.
- **Drucker-Prager yield criterion:** It states that plastic yielding begins when the J_2 invariant of the deviatoric stress and the hydrostatic stress, p , reach a critical combination.

It is relevant to introduce here the concept of *yield surface* (Φ) and *plastic potential* (Γ) so we can define what is an **associated plastic flow rule** and a **non-associated plastic flow rule**. When both potentials coincide, we have an **associated plastic flow rule**, and a **non-associated plastic flow rule** otherwise. It can be interpreted as saying that the plastic strain increment vector is normal to the *yield surface* (Φ), this is denominated the *normality rule*[BookOll14].

All the yield criteria presented here, except for *Barret de Saint Venant*, are available in *Kratos* at the module `StructuralMechanicsApplication`. This includes all the **non-associated plastic flow rule** combinations. See [B.2.Kratos Multiphysics](#) for further details of the *Kratos* structure and [5.5.Implementation details](#) for the implementation details of the constitutive models. For more detailed description on the yield criteria, we address directly to the literature[BookOll14; BookNPO09].

5.2.2.2 Yield surface hardening

Essentially[BookNPO09], hardening is characterised by a dependence of yield stress level upon the history of plastic straining to which the body has been subjected. The following are the main hardening behaviours (Figure 5.2 summarises the two last types of hardening behaviours):

- **Perfect plasticity:** A material model is said to be perfectly plastic if no hardening is allowed, that is, the yield stress level does not depend in any way on the degree of plastification. In this case, the yield surface remains fixed regardless of any deformation process the material may experience.
- **Isotropic hardening:** A plasticity model is said to be isotropic hardening if the evolution of the yield surface is such that, at any state of hardening, it corresponds to a uniform (isotropic) expansion/contraction of the initial yield surface, without translation. The movements of the yield surface can be:
 - *Positive:* When the surface experiences an expansion of the initial yield surface. This is also denominated an isotropic hardening elasto-plastic process.
 - *Null:* When the plastic loading surface does not experience any evolution. Also referred as isotropic perfectly elasto-plastic process.
 - *Negative:* When there is a contraction in the initial yield surface. This corresponds with an isotropic softening elasto-plastic process.
- **Kinematic hardening:** Appears when the yield surfaces preserve their shape and size but translate in the stress space as a rigid body. It is frequently observed in experiments that, after being loaded (and hardened) in one direction, many materials show a decreased resistance to plastic yielding in the opposite direction. This phenomenon is known as the *Bauschinger* effect and can be modelled with the introduction of kinematic hardening.

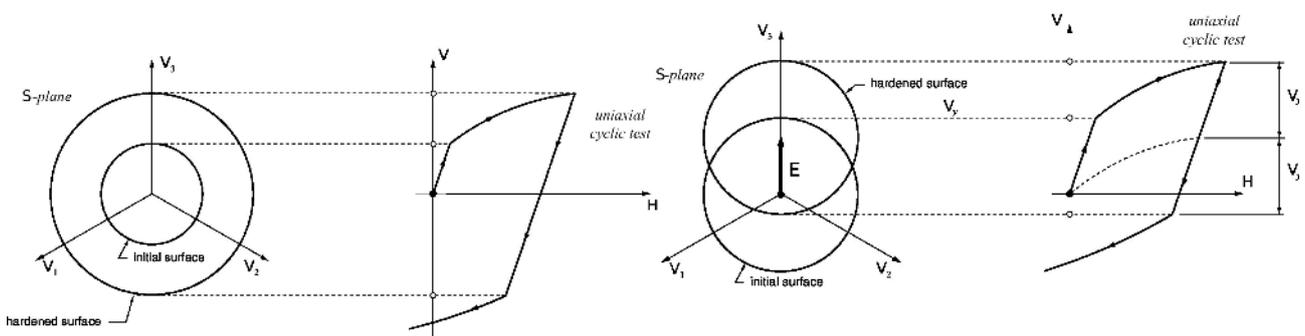


Figure 5.2: Types of hardening. Left: Isotropic hardening. Right: Kinematic hardening

5.3 Finite strain elasto-plastic models

In many industrial applications, such as the metal forming processes, we cannot take into account the infinitesimal deformation hypothesis for the plasticity theory, without a significant loss in accuracy or even completely wrong

results[BookNPO09]. So, in order to tackle this kind of problem we need to formulate the problem in finite deformations. The following introduces the basic theoretical fundamentals and applied concepts in order to formulate the finite deformation elasto-plastic model.

5.3.1 Fundamentals

The following introduces the base theoretical fundamentals required to formulate the finite strain elasto-plastic problem. The Equation (5.2) introduces the base formulation required for a finite strain elasto-plasticity[BookBHS07; BookNPO09] models.

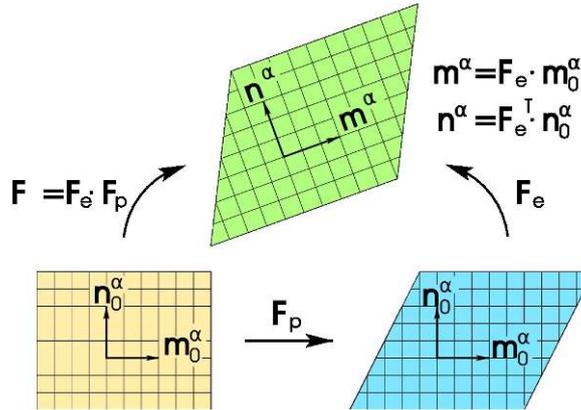


Figure 5.3: Multiplicative decomposition of the deformation gradient \mathbf{F}

1. Multiplicative decomposition of the deformation gradient \mathbf{F} , being \mathbf{F}^e and \mathbf{F}^p the elastic and plastic strains (Figure 5.3) originally proposed in [ArtLL67].

$$(5.2a) \quad \mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p$$

2. The *free-energy potential* $\bar{\Psi}(\mathbf{C}^e, \alpha)$ and elastic strain-stress relation for the stresses ($\bar{\mathbf{S}}$) in the intermediate configuration. In here \mathbf{C}^e correspond with the elastic right *Cauchy-Green* tensor and the (strain-like) internal variables (α) of the current constitutive model. With the *free-energy potential* it is possible to derive the hyperelastic law considered.

$$(5.2b) \quad \begin{cases} \bar{\mathbf{S}} = 2 \frac{\partial \bar{\Psi}}{\partial \mathbf{C}^e} \\ \bar{\mathbf{S}} := \mathbf{F}^{e-1 T} \boldsymbol{\tau} \mathbf{F}^{e-T} = \mathbf{F}^p \mathbf{S} \mathbf{F}^{p T} \end{cases}$$

3. *Yield surface* $\Phi(\mathbf{T}, \mathbf{q})$ for the stress-like hardening variables $\mathbf{q} = -\partial \Psi / \partial \alpha$ and the *Mandel stress* $\mathbf{T} = \mathbf{F}^{e T} \boldsymbol{\tau} \mathbf{F}^{e-T} = \mathbf{C}^e \bar{\mathbf{S}}$. The yield function will define the onset of the plastic yielding.
4. Plastic evolution equations, or *dissipation potential*, from which the flow rule and hardening law (evolution laws) for internal variables are derived. In order to define that we require of a *plastic potential* (Γ). Being \mathbf{L}^p the plastic velocity gradient and the general flow functions, \mathbf{M}_p the plastic flow function and \mathbf{M}_h the general hardening flow function.

$$(5.2c) \quad \begin{cases} \mathbf{L}^p = \dot{\gamma} (\mathbf{R}^e)^T \frac{\partial \Gamma}{\partial \boldsymbol{\tau}} \mathbf{R}^e = \dot{\gamma} \mathbf{M}_p(\mathbf{T}, \mathbf{q}) \text{ (This expression is also known as the normality rule)} \\ \dot{\alpha} = \dot{\gamma} \mathbf{M}_h(\mathbf{T}, \mathbf{q}) \end{cases}$$

For $\mathbf{L}^p = \dot{\mathbf{F}}^p \mathbf{F}^{p-1}$ and the plastic multiplier ($\dot{\gamma}$), with $\mathbf{M}_p = \frac{\partial \Gamma}{\partial \boldsymbol{\tau}}$ and $\mathbf{M}_h = \frac{\partial \Gamma}{\partial \mathbf{q}}$ for the associative case.

5. Loading/unloading conditions, for the elasto-plastic model the **KKT** complementary conditions and the consistency condition during plastic loading.

$$(5.2d) \quad \begin{cases} \dot{\gamma} \geq 0, \Phi \leq 0, \dot{\gamma} \Phi = 0 \\ \dot{\gamma} \dot{\Phi} = 0 \text{ (Consistency condition)} \end{cases}$$

From these **KKT** conditions we can deduce[BookOll14]:

$$(5.2e) \quad \begin{cases} \Phi < 0 \Rightarrow \dot{\gamma} = 0 \text{ Elastic behaviour or unloading,} \\ \Phi = 0 \Rightarrow \begin{cases} \dot{\gamma} > 0 \\ \dot{\gamma} = 0 \end{cases} \text{ Neutral plastic load} \\ \Phi > 0 \Rightarrow \text{Incompatible state} \end{cases}$$

Additionally in the case of numerical implementation it is necessary to implement **return-mapping** algorithm to obtain the current state of the material. In our case we take into consideration the implicit *Backward Euler* algorithm, well known in the literature[ArtMT94]. More details concerning the **return-mapping** algorithm in the following section, 5.3.3. Finite strain return mapping operations.

5.3.2 Plastic loading surface

5.3.2.1 Introduction

In order to evaluate the plastic multiplier ($\dot{\gamma}$), we will adopt the formulation from Oller[BookOll14; PhDoll88]. This will be evaluated considering the *plastic loading function* in addition to the *yield surface* (Φ). In order to understand the difference between these two functions, we need to define the influence in the constitutive behaviour by each one of the functions[BookOll14]. The limit between the elastic and the plastic areas is set through the *yield surface* (Φ) and from such limit this surface can move in the stress space, follow the evolution of the plastic process and transform itself into the so-called *plastic loading surface*. This function representing the plastic loading surface is simply the discontinuity or yield limit function updated for each of the (stress-like) internal variables value (\mathbf{q}) at every moment of the pseudo-time (t) of the elasto-plastic process. The phenomenon governing the *yield surface* change of position in the stress space is known as plastic hardening, where some models have been introduced in the state-of-the-art section, 5.2.2.2. Yield surface hardening, and now we detail the formulation for isotropic (5.3.2.2. Isotropic hardening) and kinematic hardening (5.3.2.3. Kinematic hardening).

5.3.2.2 Isotropic hardening

We define the isotropic hardening considering a *plastic hardening function* (\mathcal{K}), which we generally define with the (stress-like) internal variables (\mathbf{q}). In classic plasticity, it is defined depending on the *internal variable of plastic hardening*, Equation (5.3).

$$(5.3) \quad \mathcal{K}(\kappa^P) = f(\kappa^P)$$

Defining the *hardening function* (\mathcal{K}) as an internal variable of the plastic process, we can obtain a more general formulation, as seen in (5.4). In here the tensor function ($\mathbf{h}_\kappa(\boldsymbol{\sigma}, \mathbf{q})$) and the scalar function ($h_\kappa(\boldsymbol{\sigma}, \mathbf{q})$) depends on the updated tensor state and the internal variables.

$$(5.4) \quad \begin{aligned} \dot{\kappa}^P &= \dot{\gamma} H_\kappa(\boldsymbol{\sigma}, \mathbf{q}) = \dot{\gamma} \left[\mathbf{h}_\kappa(\boldsymbol{\sigma}, \mathbf{q}) : \frac{\partial \Gamma(\boldsymbol{\sigma}; \kappa^P)}{\partial \boldsymbol{\sigma}} \right] \\ \dot{\mathcal{K}} &= \dot{\gamma} H_\mathcal{K}(\boldsymbol{\sigma}, \mathbf{q}) = h_\mathcal{K}(\boldsymbol{\sigma}, \mathbf{q}) \dot{\kappa}^P \end{aligned}$$

5.3.2.3 Kinematic hardening

As previously stated, the kinematic hardening consists in a translational motion of the plastic loading surface. The motion depends on the *kinematic plastic hardening internal variable* ($\boldsymbol{\eta}$), which defines the stress space origin, or *back stress*. The continuous change of position of this coordinate origin during the elasto-plastic process induces a translational movement of the yield surface (Φ), which can be combined with an isotropic movement. We can express this in a general form as in the Equation (5.5).

$$(5.5) \quad \Phi(\boldsymbol{\sigma}, \mathbf{q}) = f(\boldsymbol{\sigma} - \boldsymbol{\eta}) - \mathcal{K} = 0$$

In here the plastic hardening can be defined according to *Prager* and *Melan* as $\dot{\eta} = \beta \dot{\kappa}^p$, with $\beta = \sqrt{c_k} \frac{\dot{\epsilon}^p}{\dot{\epsilon}^p}$. c_k depends on the type of plastic potential considered, being $c_k = \frac{2}{3} h_k$ for **VM** and Equation (5.6) in general.

$$(5.6) \quad c_k = \left[\frac{1}{\dot{\epsilon}_{rs}^p \dot{\epsilon}_{rs}^p} \cdot \sqrt{\frac{\sigma_{ij} \dot{\epsilon}_{ij}^p}{f(\sigma_{kl} - \eta_{kl})}} \right] \cdot h_k$$

5.3.2.4 Stress-strain relation

This can be obtained by the plastic yield general criterion and *Pragers* consistency condition, see Equation (5.7).

$$(5.7a) \quad \left. \begin{aligned} \Phi(\boldsymbol{\sigma}, \mathbf{q}) = f(\boldsymbol{\sigma} - \boldsymbol{\eta}) - \mathcal{K} = 0 \\ \dot{\Phi} = \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} + \frac{\partial \Phi}{\partial \boldsymbol{\eta}} : \dot{\boldsymbol{\eta}} + \frac{\partial \Phi}{\partial \mathcal{K}} \dot{\mathcal{K}} = 0 \end{aligned} \right\} \Rightarrow \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} + \frac{\partial \Phi}{\partial \boldsymbol{\eta}} : \dot{\boldsymbol{\eta}} - \dot{\mathcal{K}} = 0$$

From this latter expression, and replacing $\dot{\eta} = \beta \dot{\kappa}^p = c_k \dot{\epsilon}^p$ then we can derive Equation (5.7b).

$$(5.7b) \quad \begin{aligned} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\boldsymbol{\epsilon}}^e + c_k \frac{\partial \Phi}{\partial \boldsymbol{\eta}} : \dot{\boldsymbol{\epsilon}}^p - h_{\mathcal{K}} (\mathbf{h}_{\kappa} : \dot{\boldsymbol{\epsilon}}^p) = 0 \\ \left[\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\boldsymbol{\epsilon}} \right] - \dot{\gamma} \left[\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} - c_k \frac{\partial \Phi}{\partial \boldsymbol{\eta}} : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} + h_{\mathcal{K}} \mathbf{h}_{\kappa} : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} \right] = 0 \end{aligned}$$

From the latter expression, the *plastic consistency factor* ($\dot{\gamma}$). The plastic multiplier defines a factor which evaluates the distance between an inadmissible tensor state outside the domain and the plastic loading surface. With this we define $\dot{\gamma}$ as in Equation (5.7c).

$$(5.7c) \quad \dot{\gamma} = \frac{\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \dot{\boldsymbol{\epsilon}}}{\underbrace{\left[-c_k \frac{\partial \Phi}{\partial \boldsymbol{\eta}} : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} + h_{\mathcal{K}} \mathbf{h}_{\kappa} : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} \right]}_A + \left(\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathbb{C} : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} \right)} \quad \text{being } \dot{\gamma} \geq 0$$

In here we have isolated A . If we consider a zero kinematic hardening ($c_k = 0$) then A can be defined as:

$$(5.7d) \quad A = - \frac{\partial \Phi}{\partial \kappa^p} \bar{\sigma} = h_{\mathcal{K}} \mathbf{h}_{\kappa} : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}}$$

5.3.2.5 Stability condition

It is relevant to introduce here the concept of *local stability condition*, or *Druckers* stability condition. This concept is only meaningful for the stability behaviour of a point in a solid, therefore in order to ensure the stability of the whole solid we require to check the stability in each point if using this criterion. Besides, if the stability is loss in some points this does not necessarily mean that the global stability has been lost. For example, in softening materials, in which the local stability condition may not be satisfied in some points, but globally may not be the case. The stability of the whole solid can be proved by a weaker condition known as *global stability condition*.

5.3.2.5.1 Local stability :

The *Druckers* second postulates requires (5.8), and due to the fact that $\dot{\gamma}$ is always positive and defined in (5.7c), therefore can be ensured if the convexity of Γ , plastic potential, is required. One simple way to ensure $\dot{\gamma} > 0$ and the convexity of Γ is to impose an **associated plastic flow rule** ($\frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} \propto \frac{\partial \Phi}{\partial \boldsymbol{\sigma}}$).

$$(5.8) \quad \dot{\boldsymbol{\epsilon}} : \mathbb{C} : \dot{\boldsymbol{\epsilon}}^p = \dot{\boldsymbol{\epsilon}} : \mathbb{C} : \left(\dot{\gamma} \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} \right) \geq 0$$

5.3.2.5.2 Global stability :

For softening materials the previous postulate is sufficient, but not necessarily, stability condition. In order to formulate a global stability condition we will consider a weak form, as we do in order to formulate **FEM**. Defining Π as the total energy in the system in the final configuration, Π^* the total energy in the initial configuration and $\delta\mathbf{u}$ the virtual displacement from the initial configuration to the final one. With this we define Equation (5.9), which defines the equilibrium while applying the virtual displacement. In here, the variation of the total potential energy developed is null.

$$(5.9) \quad \Pi = \Pi^* + \delta\Pi + \frac{1}{2!}\delta^2\Pi + \dots \Rightarrow \Delta\Pi = \Pi - \Pi^* \cong \underbrace{\delta\Pi}_0 + \frac{1}{2!}\delta^2\Pi + \dots$$

As $\delta\Pi \approx 0$, the total increase of the virtual work is equal to the second variation of the functional and it is, therefore, the concave or convex stability condition of the functional. With this if $\Delta\Pi > 0$ the original configuration is stable for any $\delta\mathbf{u}$, and $\Delta\Pi < 0$ is unstable. With this calling Ω_0 the initial configuration and Ω_p the plastic configuration, and as $\Delta\Pi \cong \frac{1}{2} \int_{\Omega} \delta\boldsymbol{\sigma} : \delta\varepsilon d\Omega$ we can define the global stability condition as (5.10).

$$(5.10) \quad \Delta\Pi \cong \frac{1}{2} \int_{\Omega} \delta\boldsymbol{\sigma} : \delta\varepsilon d\Omega = \left[\frac{1}{2} \int_{\Omega_0} \delta\boldsymbol{\sigma} : \delta\varepsilon d\Omega + \frac{1}{2} \int_{\Omega_p} \delta\boldsymbol{\sigma} : \delta\varepsilon d\Omega \right] \cong [\Delta\Pi_{\Omega_0} + \Delta\Pi_{\Omega_p}] > 0$$

5.3.2.6 Condition of unicity of solution

Considering two different virtual displacements, $\delta\mathbf{u}_1$ and $\delta\mathbf{u}_2$, the potential energy difference is equivalent to the potential energy for a virtual displacement equivalent to the difference of those virtual displacements ($\Delta(\delta\mathbf{u}) = \delta\mathbf{u}_2 - \delta\mathbf{u}_1$), with the corresponding increment on the strain ($\Delta(\delta\varepsilon)$) and stress ($\Delta(\delta\boldsymbol{\sigma})$). The corresponding increment in potential energy is Equation (5.11).

$$(5.11) \quad \Delta(\delta^2\Pi) = \int_{\Omega} \Delta(\delta\boldsymbol{\sigma}) : \Delta(\delta\varepsilon) d\Omega \begin{cases} = 0 & \text{There is no unicity of the solution} \\ \neq 0 & \text{There is unicity of the solution} \end{cases}$$

If $\Delta(\delta^2\Pi) = 0$ during the virtual displacement change $\Delta(\delta\mathbf{u})$, it means that the stress in both final configurations are the same $\delta\boldsymbol{\sigma}_1 = \delta\boldsymbol{\sigma}_2$; in consequence, $\Delta(\delta\boldsymbol{\sigma}) = \delta\boldsymbol{\sigma}_1 - \delta\boldsymbol{\sigma}_2 = 0$. Therefore, there are two admissible and independent of each other kinematic states $\delta\mathbf{u}_1 \neq \delta\mathbf{u}_2$, but the stress increment is identical for both configurations $\delta\boldsymbol{\sigma}_1 = \delta\boldsymbol{\sigma}_2$, which implies that there is not a single solution, but there exists a bifurcation. Considering the previously stated kinematic states, if $\Delta(\delta^2\Pi) \neq 0$, the unicity of the solution is guaranteed.

5.3.3 Finite strain return mapping operations

In addition to the formerly presented, we need to consider some additional concepts. Besides the multiplicative decomposition between the elastic (\mathbf{F}^e) and plastic (\mathbf{F}^p) deformation gradient from (5.2a), we must decompose these deformation gradient in its stretch ($\mathbf{U}^e, \mathbf{V}^e, \mathbf{U}^p, \mathbf{V}^p$) and rotation tensors ($\mathbf{R}^e, \mathbf{R}^p$) as seen in (5.12).

$$(5.12) \quad \begin{aligned} \mathbf{F}^e &= \mathbf{R}^e \mathbf{U}^e = \mathbf{V}^e \mathbf{R}^e \\ \mathbf{F}^p &= \mathbf{R}^p \mathbf{U}^p = \mathbf{V}^p \mathbf{R}^p \end{aligned}$$

With this decomposition we are able to define the general elastic predictor/return-mapping algorithm. The crucial difference between the discretisation of the large strain problem and the infinitesimal one lies in the numerical approximation of the *plastic flow equation*, see (5.2c). In order to be consistent with the presented multiplicative

decomposition, we will consider an **exponential map backward discretisation**[BookNPO09]. Considering an implicit scheme algorithm, the resulting update formula for the plastic deformation gradient (\mathbf{F}^p) results in (5.13).

$$(5.13a) \quad \mathbf{F}_{n+1}^p = \exp \left[\mathbf{R}_{n+1}^{eT} \frac{\partial \Gamma}{\partial \boldsymbol{\tau}} \Big|_{n+1} \mathbf{R}_{n+1}^e \right] \mathbf{F}_n^p$$

In view of the isotropy of the tensor exponential function can be simplified as:

$$(5.13b) \quad \mathbf{F}_{n+1}^p = \mathbf{R}_{n+1}^{eT} \exp \left[\frac{\partial \Gamma}{\partial \boldsymbol{\tau}} \Big|_{n+1} \right] \mathbf{R}_{n+1}^e \mathbf{F}_n^p$$

The incompressibility of the *plastic flow* for pressure insensitive flow potentials is carried over exactly to the incremental rule (5.13b).

As we have adopted, a standard *Backward Euler* difference scheme to discretise the *plastic flow equation*, the updating formula for the plastic deformation gradient will be refactored to (5.14). This formula in general is not volume-preserving, therefore this solution will result in an accuracy loss in the resolution of elasto-plastic constitutive equations of plastically incompressible models.

$$(5.14) \quad \mathbf{F}_{n+1}^p = \left(\mathbf{I} - \mathbf{R}_{n+1}^{eT} \frac{\partial \Gamma}{\partial \boldsymbol{\tau}} \Big|_{n+1} \mathbf{R}_{n+1}^e \right)^{-1} \mathbf{F}_n^p$$

With this, in order to update the elastic deformation gradient (\mathbf{F}^e) we need to combine the definition of the plastic deformation gradient (\mathbf{F}^p) with the multiplicative elasto-plastic split, we obtain the following equivalent kinematic update expression in terms of the elastic deformation gradient (5.15).

$$(5.15a) \quad \mathbf{F}_{n+1}^e = \mathbf{F}_\Delta \mathbf{F}_n^e \mathbf{R}_{n+1}^{eT} \exp \left[\frac{\partial \Gamma}{\partial \boldsymbol{\tau}} \Big|_{n+1} \right] \mathbf{R}_{n+1}^e$$

The definition of the incremental deformation gradient is the following:

$$(5.15b) \quad \mathbf{F}_\Delta \equiv \mathbf{F}_{n+1} (\mathbf{F}_n)^{-1}$$

With this last expression we can compute the trial elastic deformation gradient, if we save the deformation gradient from the previously converged time step. This will help us to compute a consistent update of the elastic deformation gradient.

$$(5.15c) \quad \mathbf{F}_{trial}^e = \mathbf{F}_\Delta \mathbf{F}_n$$

In order to be consistent with the expression from (5.14), therefore we calculate the update of the elastic deformation gradient as:

$$(5.15d) \quad \mathbf{F}_{n+1}^e = \mathbf{F}_{trial}^e \left(\mathbf{I} + \mathbf{R}_{n+1}^{eT} \frac{\partial \Gamma}{\partial \boldsymbol{\tau}} \Big|_{n+1} \mathbf{R}_{n+1}^e \right)$$

5.4 Large deformation elastic models

5.4.1 Introduction

In order to consider *elastic-plastic* models with large deformations, we need to consider elastic models in large deformations that will act as the predictors of our plastic behaviour, derived from the *free-energy potential* from (5.2b). The most commonly used models for that purpose are the *hyperelastic* ones. We will present the two hyperelastic models considered on this work, the *Neo-Hookean* and *Kirchhoff* material. Before introduce these models, we will present the common theory that defines the *hyperelastic* materials. Hyperelastic materials[BookBel+14] are

characterised by the existence of a stored (or strain) energy function that is a potential for the stress (\mathbf{S}) (5.16a). In here Ψ corresponds with the stored energy potential, on the other hand, w is a potential expressed in terms of the *Green-Lagrange* strain.

$$(5.16a) \quad \mathbf{S} = 2 \frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{C}} = \frac{\partial w(\mathbf{E})}{\partial \mathbf{E}}$$

We can obtain in a similar manner the consistent tangent moduli \mathbb{C}^{SE} as (5.16b).

$$(5.16b) \quad \mathbb{C}^{SE} = 4 \frac{\partial^2 \psi(\mathbf{C})}{\partial \mathbf{C} \partial \mathbf{C}} = \frac{\partial^2 w(\mathbf{E})}{\partial \mathbf{E} \partial \mathbf{E}}$$

5.4.2 Neo-Hookean material

This model is an extension of the isotropic linear *Hooke's* law to finite strain. The material can be used to model materials such as plastics and rubber-like substances. The corresponding potential is shown in the Equation (5.17a). With this and considering (5.16a) we can obtain the corresponding stresses (5.17b). In a similar manner, we can obtain with (5.16b) the consistent \mathbb{C} can be obtained with (5.17c). In here $\lambda_0 = \frac{\nu E}{(1+\nu)(1-2\nu)}$ and $\mu_0 = \frac{E}{2(1+\nu)}$ are the *Lamé* constants.

$$(5.17a) \quad \Psi(\mathbf{C}) = \frac{1}{2} \lambda_0 (\ln J)^2 - \mu_0 \ln J + \frac{1}{2} \mu_0 (\text{trace } \mathbf{C} - 3)$$

$$(5.17b) \quad \begin{aligned} \mathbf{S} &= \lambda_0 \ln J \mathbf{C}^{-1} + \mu_0 (\mathbf{I} - \mathbf{C}^{-1}) \\ \boldsymbol{\tau} &= \lambda_0 \ln J \mathbf{I} + \mu_0 (\mathbf{B} - \mathbf{I}) \end{aligned}$$

$$(5.17c) \quad \begin{aligned} C_{ijkl}^{SE} &= \lambda C_{ij}^{-1} C_{kl}^{-1} + \mu (C_{ik}^{-1} C_{jl}^{-1} + C_{il}^{-1} C_{kj}^{-1}) \\ C_{ijkl}^T &= \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{kj}) \end{aligned}$$

5.4.3 Kirchhoff material

The so-called *Saint Venant-Kirchhoff* material or just *Kirchhoff* material is probably the simplest **NL** elastic model. It is basically an extension of the *Hooke's* law. In cases where we have a problem in small strains and large rotations, this material provides a proper **NL** behaviour. In this case the potential w can be expressed as (5.18a). With this we can obtain the corresponding \mathbb{C}^{SE} (5.18b) and the \mathbf{S} (5.18c).

$$(5.18a) \quad w = \int S_{ij} dE_{ij} = \int C_{ijkl} E_{kl} dE_{ij} = \frac{1}{2} C_{ijkl} E_{ij} E_{kl} = \frac{1}{2} \mathbf{E} \cdot \mathbf{C} : \mathbf{E}$$

$$(5.18b) \quad \mathbb{C}^{SE} = \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbf{I}$$

$$(5.18c) \quad \mathbf{S} = \lambda \text{trace}(\mathbf{E}) \mathbf{I} + 2\mu \mathbf{E} = \mathbf{C} : \mathbf{E}$$

5.5 Implementation details

5.5.1 Introduction

The following section will include a series of numerical details related with the implementation of the constitutive models presented in this chapter. In a first subsection, 5.5.2. *Class structure in Kratos*, we will detail the modular structure followed in order to implement this numerical models inside *Kratos*. In this section only the *isotropic hardening* constitutive laws are detailed, but the very same principle has been considered for the *kinematic hardening* constitutive laws. For further detail about *Kratos* structure, we directly address to the corresponding appendix, B.2. *Kratos Multiphysics*.

The last subsection will introduce the numerical methods considered in order to obtain the tangent operators of the respective constitutive models, 5.5.3. *Numerical implementation tangent constitutive tensor*.

5.5.2 Class structure in Kratos

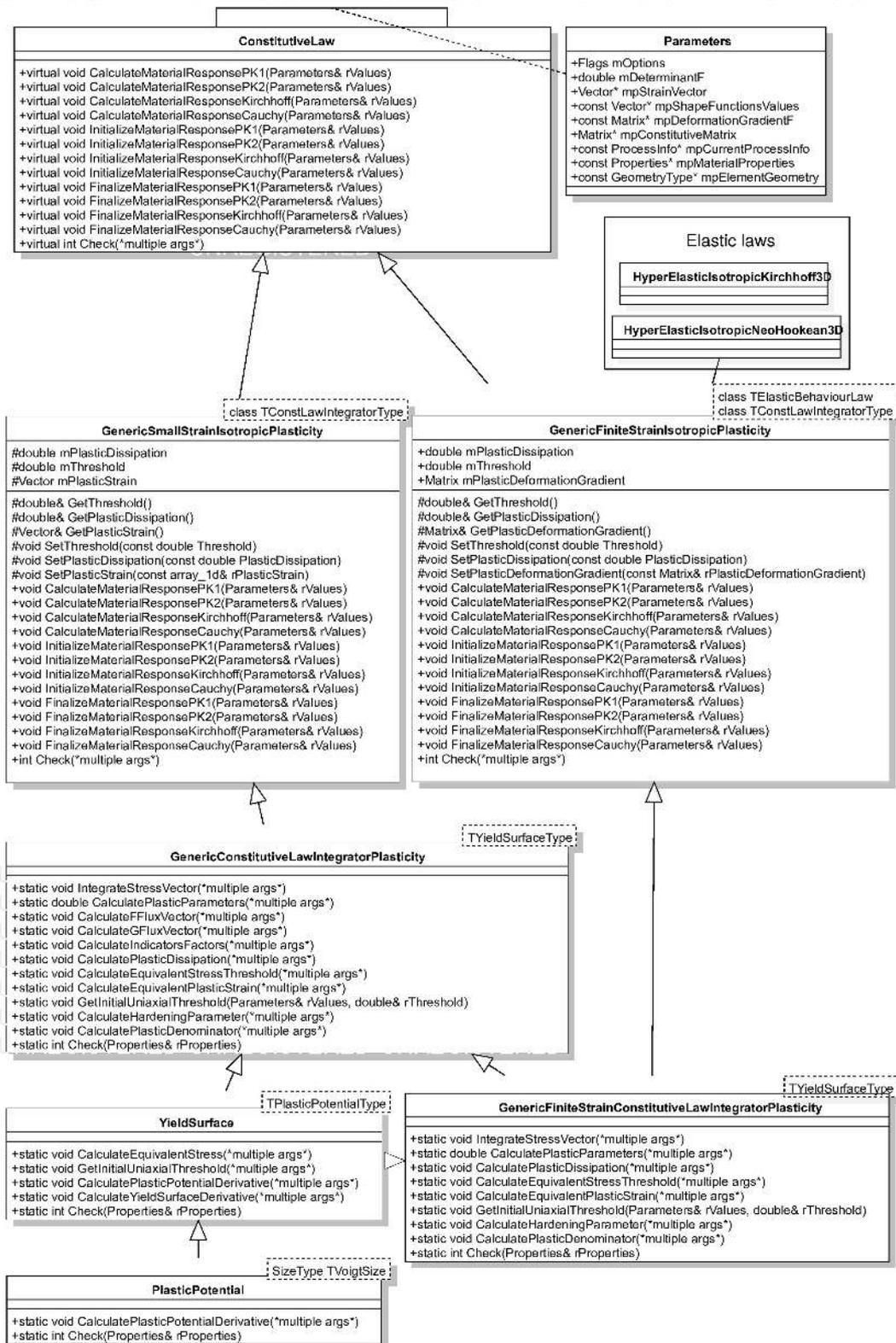


Figure 5.4: Elastic-plastic CL class structure

The class structure of the **CL** implemented in *Kratos* can be summarised as shown in Figure 5.4. This provides a modular interface similar to a *Matrioshka*, where the classes are recursively inside the others. In this class structure, we can appreciate like the templates are an essential part in the design. The main components of the **CL** are:

- **PlasticPotential**: Corresponds with Γ . It depends on the *Voigt* size (`TVoigtSize`). This class defines the plastic potential derivative.
- **YieldSurface**: Corresponds with Φ . This depends on the **PlasticPotential**. It is the class responsible of the equivalent stress and its derivative.
- **GenericConstitutiveLawIntegratorPlasticity**: With a derived class for **Finite Strain (FS)**. Depends on the **YieldSurface**. This is the class responsible of computing and integrate the stress vector.
- **GenericSmallStrainIsotropicPlasticity**: Depends on **GenericConstitutiveLawIntegratorPlasticity**. This is the **CL** which actually computes the response, for **Small Strain (SS)**.
- **GenericFiniteStrainIsotropicPlasticity**: The same as the previous case, but for **FS**. Additionally to the **GenericFiniteStrainConstitutiveLawIntegratorPlasticity** also depends on the **FS** elastic **CL** from 5.4. *Large deformation elastic models*.
- **ConstitutiveLaw**: Finally, everything it also derives from the base **CL**, which defines the base **API** for constitutive models.
- **Parameters**: The **CL** operates with the **Parameters** class which is extensively used as input class.

In conclusion, this is a generic interface for constructing elastic-plastic constitutive laws, which allows an extensive combination. The *drawback* to this is a significative increase at the compilation time for each combination.

5.5.3 Numerical implementation tangent constitutive tensor

It is not always possible to obtain the consistent tangent constitutive tensor in an analytical expression, or if possible this one is not trivial. The class structure shown in the previous section 5.5.2. *Class structure in Kratos* allows to obtain a large number of *elastic-plastic non-associated CL* combinations, and the corresponding \mathbb{C} which obtainment is not as simple as in the *associated* cases. In order to obtain the consistent \mathbb{C} we proceed with a numerical procedure based in the stress vector perturbation method [ArtMar+08; ArtMOB11]. This procedure is based in the concept from Equation (5.19) from Barbu [PhDBar16] and Cornejo [PhDCor], where the components of \mathbb{C} are associated to the increases in the stresses and the strain [ArtMOB11]. An alternative method to the previously stated would be considering the **AD**, as the method is applied with a *Forward Euler* scheme is denominated sometimes as **Forward Automatic Differentiation (FAD)** [ArtCOH14; ArtRH15].

$$(5.19) \quad \mathbb{C} = \mathbb{C}_{ijkl}^t = \frac{\partial \sigma_{ij}}{\partial \varepsilon_{kl}^e}$$

Here ε_{kl}^e corresponds with the elastic components of the strain tensor.

Particularly for the elasto-plastic tangent operator (\mathbb{C}^{ep}) is given by the expression of (5.20). This expression is not easy to compute in general, due to this, it is interesting to have a numerical alternative that provides us a general method for computing the tangent tensor.

$$(5.20) \quad \mathbb{C}^{ep} = \mathbb{C} - \frac{[\mathbb{C}^p : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}}] \otimes [\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathbb{C}]}{h_c \mathbf{h}_\kappa : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}} + (\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathbb{C}^p : \frac{\partial \Gamma}{\partial \boldsymbol{\sigma}})}$$

5.5.3.1 Forward FD

This is the approach followed by *Martinez*[ArtMar+08; ArtMOB11]. The approach takes only the first-order term of the *Taylor expansion* of (5.19). This leads to the forward **FD** from Equation (5.21a). Applying this **FD** to (5.19) we obtain the expression of Equation (5.21b). We can appreciate that this method implies to probe and calculate the stress 6 times in case of a 3D *Voigt* notation strain tensor is considered, therefore this is an expensive procedure, but can be acceptable in case of an implicit approach as ours.

The *Forward difference* can be defined as follows:

$$(5.21a) \quad \left(\frac{\partial u}{\partial x} \right)_i \approx \frac{u_{i+1} - u_i}{\Delta x}$$

Applying to our problem:

$$(5.21b) \quad \mathbb{C} = C_{ijkl}^t \approx \frac{\Delta \sigma_{ij}}{\Delta \varepsilon_{kl}^e} \approx \frac{\sigma_{ij}^{\varepsilon_{kl}^e + \Delta \varepsilon_{kl}^e} - \sigma_{ij}^{\varepsilon_{kl}^e}}{\Delta \varepsilon_{kl}^e}$$

Unfortunately, this approach is a mere first order $\mathcal{O}(h)$, Figure 5.5, so in order to ensure the quadratic convergence of the **NR** we recommend to consider the approach of the following section (5.5.3.2. *Centered FD*).

5.5.3.2 Centered FD

Applying a *Taylor expansion* to the criteria previously shown, and considering central difference approach (5.22a), we can obtain a second order approach $\mathcal{O}(h)^2$ [ArtPRH00]. The rate of convergence can be proved to be quadratic in this case, see Figure 5.5. In this case the drawback is that the number of perturbations required is double than the previous scheme, so the computational cost increase must be collated with the convergence boost obtained; therefore this will depend of the type of problem solved and the number of **DOF** of the system.

The definition of a central difference is the following:

$$(5.22a) \quad \left(\frac{\partial u}{\partial x} \right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}$$

We can apply this to the former (5.19) and then obtain:

$$(5.22b) \quad \mathbb{C} = C_{ijkl}^t \approx \frac{\Delta \sigma_{ij}}{\Delta \varepsilon_{kl}^e} \approx \frac{\sigma_{ij}^{\varepsilon_{kl}^e + \Delta \varepsilon_{kl}^e} - \sigma_{ij}^{\varepsilon_{kl}^e - \Delta \varepsilon_{kl}^e}}{2\Delta \varepsilon_{kl}^e}$$

This method is considered by default for constitutive relations in the software *MFront* developed by *Helfer*[ArtHPF15]. This software is specialised in the modelling of generic constitutive behaviours, showing up the generality and power of the method.

5.5.3.3 Numerical details

In *Kratos*, as any standard **FE** code, the constitutive equations of the material provide for a certain deformational state (ε or \mathbf{F}), the associated the stress tensor and its internal variables (α). From the resulting tensile-deformation state, a small disturbance is applied to ε or \mathbf{F} , and the stresses associated with this new deformation are computed. These stresses, together with the perturbation applied, will be used to calculate \mathbb{C} by means of the equations shown. This procedure will be repeated for each component of the corresponding strain tensor (ε) or the deformation gradient (\mathbf{F}).

From the mathematical point of view, the ideal perturbation should be $\Delta \varepsilon \rightarrow 0$ to obtain the exact \mathbb{C} . Of course from a numerical point of view this is not possible, and we must compute finite value such that the approximation is numerically close to the analytical value but free of numerical issues. For this, we will consider the *Martinez*[ArtMOB11] criteria, shown in Equation (5.23).

$$(5.23) \quad \begin{aligned} \text{if } \varepsilon_j \neq 0 &\rightarrow \Delta \varepsilon_j = \varepsilon_j \cdot 10^{-5} \\ \text{if } \varepsilon_j = 0 &\rightarrow \Delta \varepsilon_j = \min\{|\varepsilon_k|\} \cdot 10^{-5} \quad \forall k = 1, n \quad \varepsilon_k \neq 0 \end{aligned}$$

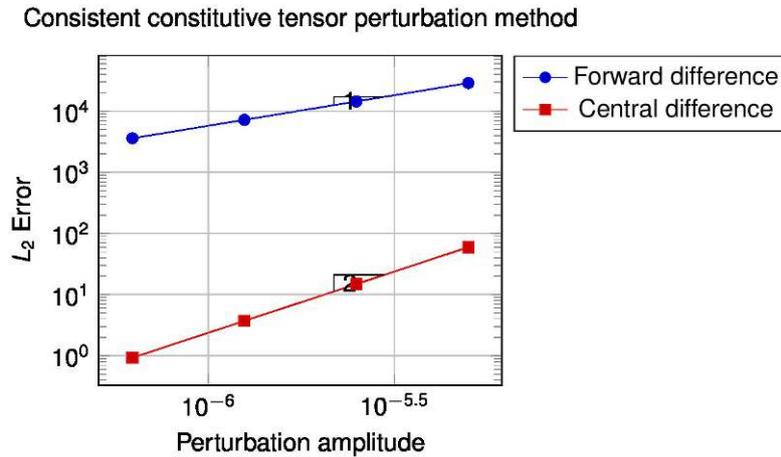


Figure 5.5: Consistent constitutive tensor perturbation method convergence. The code which tests the convergence rate can be found [here](#)

Applying this, we ensure that the increase will be always close to the initial configuration. The only drawback of the process described is that it can provide values of the perturbation near zero. *i.e.*, when one of the components of the ε is close to zero. In this situation, the Equations (5.21b) or (5.22b) may lead to an indeterminacy. In order to avoid this situation, we need to ensure that the value of the perturbation is large enough, for that we can impose the condition [ArtMOB11] from Equation (5.24).

$$(5.24) \quad \Delta\varepsilon_j > \max\{\varepsilon_k\} \cdot 10^{-10} \quad \forall k = 1, n$$

5.6 Numerical examples

5.6.1 Introduction

The following section introduces three minimal examples to show the results obtained with the implementation presented. First a minimal cube, one *hexahedron* element, is tested considering the proposed class structure approach and compared with the exact consistent *Von Mises* yield criterion. The second example is a tensile test, comparing the solution obtained with different types of elements. The third example is the application of the **CL** to a contact problem.

5.6.2 Cube minimal example

The problem here presented consists in a very simple mesh, consisting in only one *hexahedron* element, Figure 6.17. The size of the cube is $1 \times 1 \times 1$. In this example we will compare the solution obtained with an analytical consistent *elastic-plastic* law with an associative *Von Mises* yield criterion (J_2) with the **CL** derived of our class structure previously presented.

So, this example will compare the solution obtained for an *elastic-perfectly plastic* **CL** with the following parameters Table 5.1. In addition to the former, for our implementation, a large value for the fracture energy must be considered in order to properly model the perfect plasticity (*i.e.* 10^{16}).

The nodes from the base (1, 2, 3, 4 in Figure 6.17) will be considered fixed, the rest of the nodes (5, 6, 7, 8 in Figure 6.17) will be moved according to the following pattern (5.25), where ω is the frequency of the movement and will be considered to be $\omega = 5\text{Hz}$.

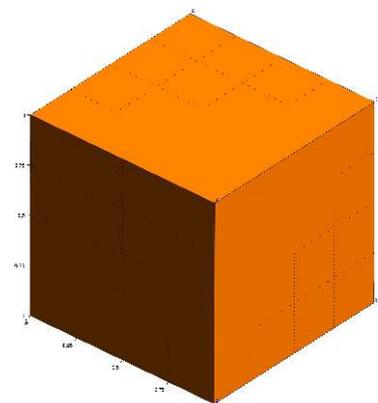


Figure 5.6: Simple cube geometry. Only one 8-node *hexahedron*

E	ν	γ_c
$2 \cdot 10^{11} Pa$	0.3	$9 Pa$

Table 5.1: Parameters considered for the cube minimal example

(5.25a)
$$\mathbf{u} = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2$$

Where:

(5.25b)
$$\mathbf{x}_1 = \sin(\omega t)\mathbf{X}, \mathbf{x}_2 = \sin(2\omega t)\mathbf{X}$$

(5.25c)
$$\mathbf{A} = \begin{bmatrix} 10^{-10} & -2 \cdot 10^{-10} & 0 \\ 5 \cdot 10^{-11} & 7 \cdot 10^{-11} & 10^{-11} \\ -2 \cdot 10^{-11} & 0 & -3 \cdot 10^{-10} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 7 \cdot 10^{-10} & 0 \\ 2.5 \cdot 10^{-10} & 1.7 \cdot 10^{-10} & 10^{-11} \\ 0 & 10^{-10} & -3 \cdot 10^{-10} \end{bmatrix}$$

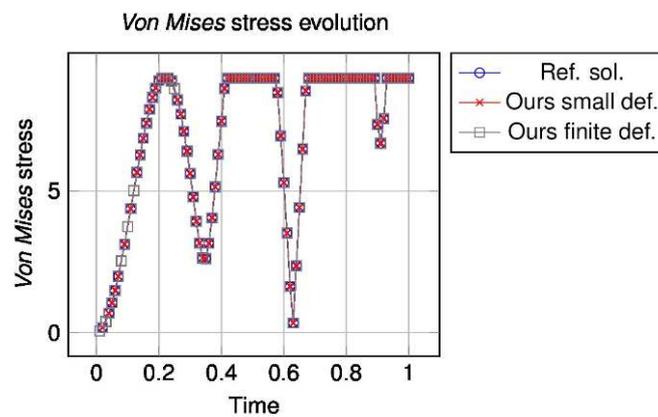


Figure 5.7: Von Mises stresses compared with the reference solution. The test can be found [here](#)

Finally, Figure 5.7 shows the solution comparison. We can see that the obtained solution is coincident with the reference solution. We can also appreciate that as expected in a *elastic-perfectly plastic* case the γ_c acts as the asymptotic value of the equivalent stress, Von Mises in this case.

5.6.3 Tensile test

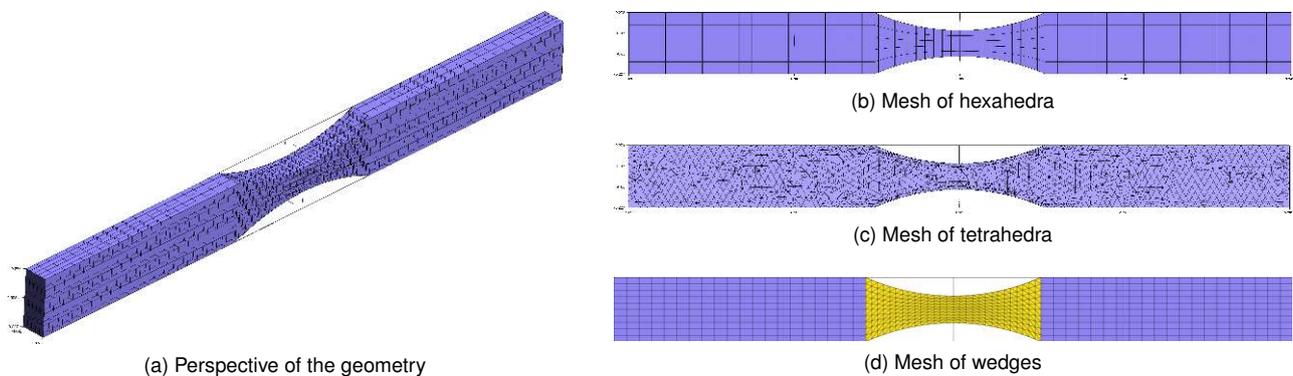


Figure 5.8: Mesh comparisons. It can be found [here](#)

The problem consists on a **tensile test**. Three different meshes have been created, see Figure 5.8, in order to validate the computation of the internal energy dissipation, which depends on the element length as previously shown. The test in general also shows the proper work of the implemented **CL**.

The geometry can be appreciated in Figure 5.8a. The meshes of hexahedra and wedges are structured (Figures 5.8b and 5.8d), on the other hand, the mesh of tetrahedra is unstructured (Figure 5.8c). We are in tensile bar with the following sizes $0.2 \times 0.5 \times 5.39$ and the material properties shown in Table 5.2. Additionally in all cases, an associative *Von Mises* model is considered, with an exponential softening as hardening curve.

E	ν	γ_c	G_f	Max. stress position	Maximum stress	Friction angle	Dilatancy angle
$2 \cdot 10^{11} Pa$	0.29	$2 \cdot 10^8 Pa$	$4 \cdot 10^5$	0.1	$210 \cdot 10^6 Pa$	32	32

Table 5.2: Parameters considered for the tensile test

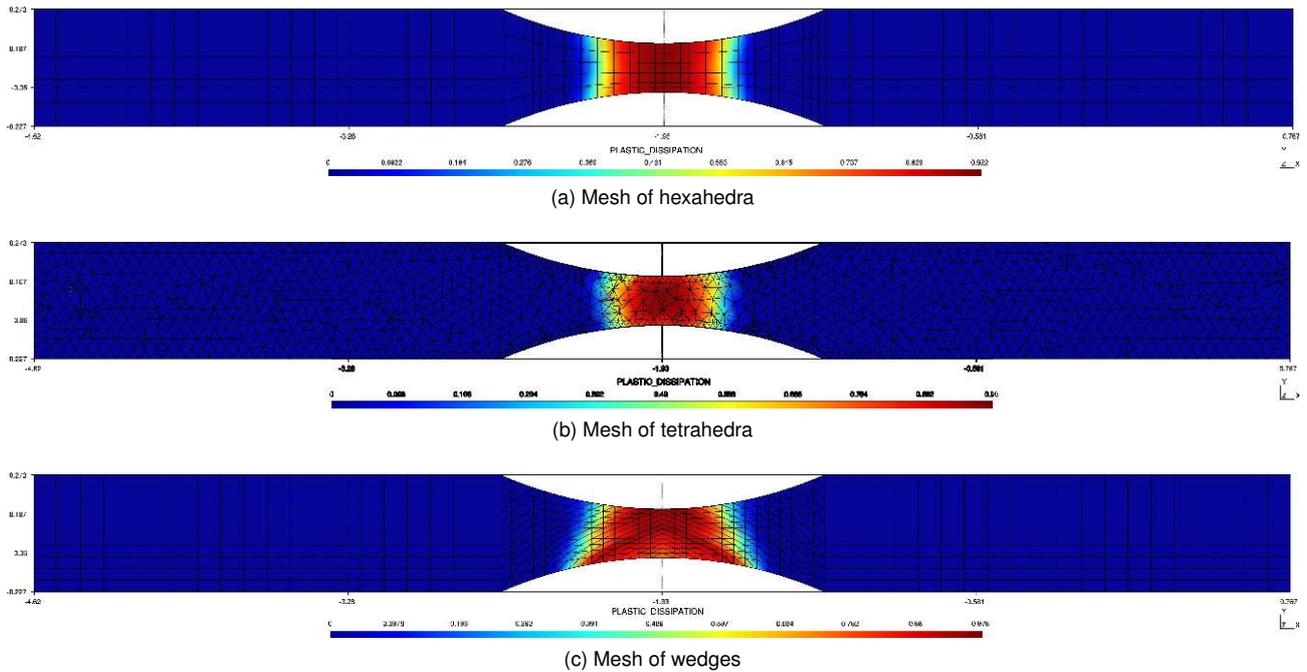


Figure 5.9: Resulting plastic dissipation

The left side is blocked and the right side of the tensile bar is moved with a pacing of $u_x = 0.003t$, being t the time variable which goes from 0.847 to 20. The resulting plastic dissipation at the end of the simulation (Figure 5.10) for each mesh corresponds with Figure 5.9.

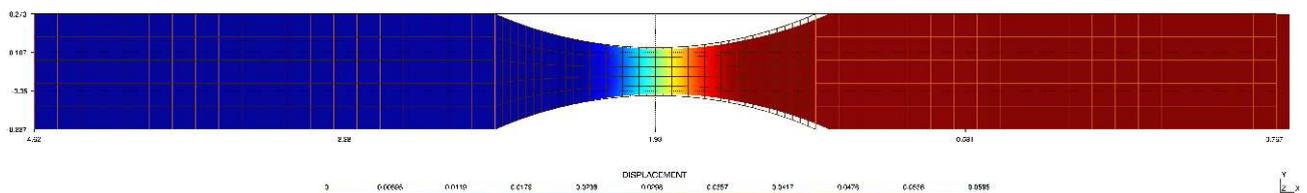


Figure 5.10: Resulting displacement for the mesh of hexahedra

From the Figure 5.9 we can appreciate the effect of the mesh element in the results obtained. The mesh of hexahedra from Figure 5.9a shows us a homogeneous distribution of the values. From the mesh of tetrahedra, we obtain a distribution with higher peaks and narrower than the previous case. Finally, in the case of the wedges, where the mesh is structured but non-symmetric, we can appreciate a very significant influence of the mesh. The distribution obtained depends on the mesh orientation.

With this we can conclude that the **mesh has a significant role on the results obtained** from the point of view of the **CL** considered. Because of this we must **pay attention** to the values considered and the mesh and element type considered.

5.6.4 Application in CCM. Gears example

The following is an example of a contact problem considering a plasticity law. The problem consists in two gears, one is fixed, the other has a rotational movement imposed. The rotational movement imposed in the smaller gear corresponds with Equation (5.26). The movement is such as the two gears come into contact.

$$(5.26) \quad \begin{cases} u_x = x_0 - R \cos \left(\text{atan} \left(\frac{y_0}{x_0} \right) - \omega t \right) \\ u_y = y_0 - R \sin \left(\text{atan} \left(\frac{y_0}{x_0} \right) - \omega t \right) \end{cases}$$

The contact area is a small region compared with the rest of the geometry of the problem as seen in Figure 5.11. Because of this, the contacting gear does not experiences any significant movement or increase of stresses in a overall view, see Figure 5.13. Is on the details of Figure 5.12 where we can appreciate that the larger gear is the one which experiences highest stresses. Because of this, it is the larger gear which exceeds γ_c and then yields, as seen in Figure 5.12c.

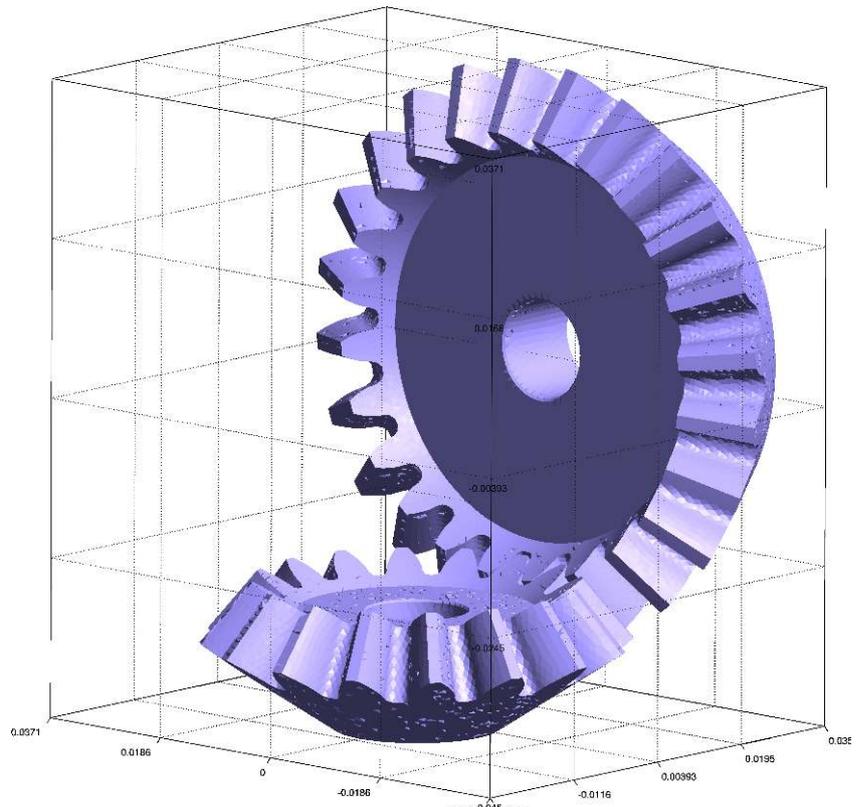


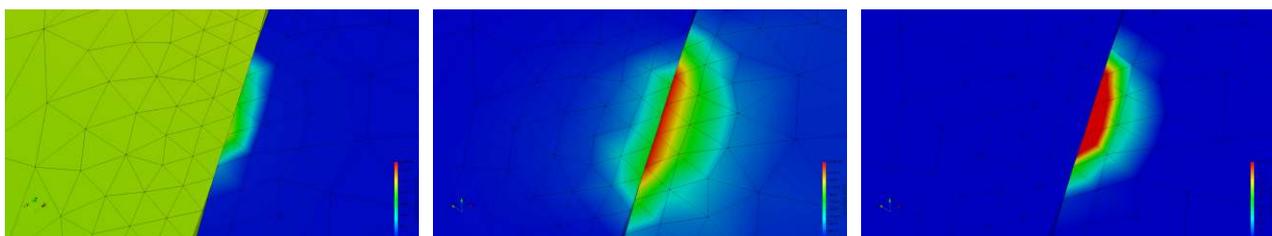
Figure 5.11: Gears example mesh

This example can be found in [Kratos examples repository](#). Additionally to the current setup linear elastic case is also present.

E	ν	γ_c	G_f
$2 \cdot 10^{11} Pa$	0.29	$52510^6 Pa$	$1 \cdot 10^8 Pa$

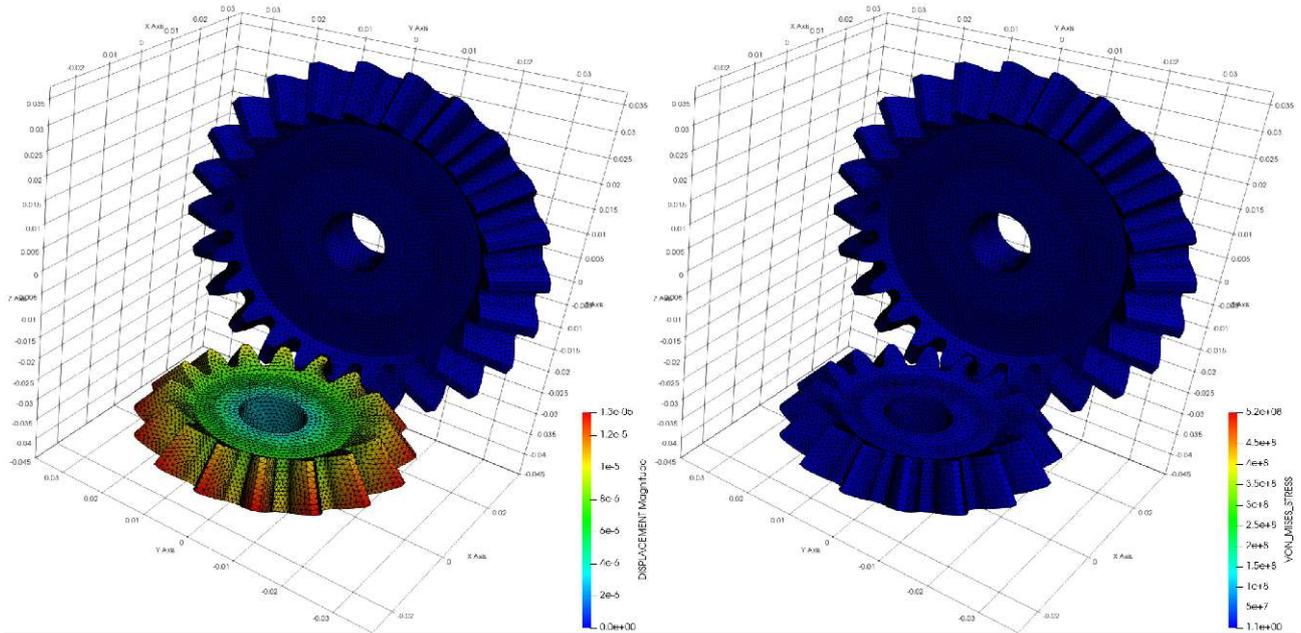
Table 5.3: Parameters considered for gear example

The parameters are the ones considered in Table 5.3. As an *elastic-perfectly plastic* in an associative **VM** yield criterion (J_2) the only plasticity parameters we need to define are γ_c and G_f . In order to compute the contact in considering this **NL** behaviour, the only precaution needed is to adjust the initial contact pair detection to avoid initial overpressures due to contact pairs in tension that are not supposed to come into contact.



(a) Displacement solution (b) VM stress (c) Plastic dissipation

Figure 5.12: Detail solution for the displacement, **VM** stress and plastic dissipation



(a) Displacement solution

(b) VM stress

Figure 5.13: Overall solution for the displacement and the VM stress

Bibliography

Books

- [BookBan10] *Sheet Metal Forming Processes: Constitutive Modelling and Numerical Simulation*. Dorel Banabic. Springer-Verlag Berlin Heidelberg. 1st ed. 2010.
- [BookBel+14] *Nonlinear Finite Elements for Continua and Structures*. Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. Wiley. 2nd ed. 2014.
- [BookBHS07] *Encyclopedia of Computational Mechanics (3 Volume Set)*. René de Borst, T.J.R. Hughes, and Erwin Stein. Wiley. 1st ed. 2007.
- [BookHil98] *The mathematical theory of plasticity*. R. Hill. Clarendon Press; Oxford University Press. 1998.
- [BookHS98] *Computational Inelasticity*. T.J.R. Hughes and J.C. Simo. Springer-Verlag New York. 1st ed. 1998.
- [BookLub08] *Plasticity theory*. Jacob Lubliner. Dover Publications. web draft. 2008.
- [BookNPO09] *Computational Methods for Plasticity Theory and Applications*. E. A. de Souza Neto, D. Perić, and D.R.J. Owen. Wiley. 2009.
- [BookOll14] *Nonlinear Dynamics of Structures*. Sergio Oller. Springer International Publishing. 2014.
- [BookSim93] *Topics in the numerical analysis and simulation of classical plasticity*. JC Simo. Elsevier Amsterdam. 1993.
- [BookTre65] *Mémoire sur l'écoulement des corps solides, par MH Tresca*. Henri Tresca. Noblet et Baudry. 1865.
- [BookWei] *Plasticity: Mathematical Theory and Numerical Analysis*. B. Daya Reddy (auth.) Weimin Han. Springer-Verlag New York. 2nd ed.
- [BookZTF14] *The Finite Element Method for Solid and Structural Mechanics*. O. C. Zienkiewicz, Robert L. Taylor, and David Fox. Butterworth-Heinemann. 7th ed. 2014.
- [BookZyc81] *Combined loadings in the theory of plasticity*. Michal Zyczkowski. Springer Science & Business Media. 1981.

Articles

- [ArtAA95] “Hydrogen-enhanced localization of plasticity in an austenitic stainless steel”. Daniel P Abraham and Carl J Altstetter. In: *Metallurgical and Materials transactions A*. No. 11, Vol. 26, 1995, pp. 2859–2871. Springer. DOI: [10.1007/bf02669644](https://doi.org/10.1007/bf02669644).
- [ArtBau86] “Über die Veränderung der Elastizitätsgrenze und der Festigkeit des Eisens und Stahls durch Strecken und Quetschen, durch Erwärmen und Abkühlen und durch oftmals wiederholte Beanspruchung”. Johann Bauschinger. In: *Mitteilungen des mechanisch-technischen Laboratoriums der Königlich Technischen Hochschule München*. No. 1, Vol. 13, 1886,
- [ArtCOH14] “Automatic differentiation for numerically exact computation of tangent operators in small-and large-deformation computational inelasticity”. Qiushi Chen, Jakob T Ostien, and Glen Hansen. In: . 2014, pp. 289–296. DOI: [10.1002/9781118889879.ch38](https://doi.org/10.1002/9781118889879.ch38).

- [ArtCou76] “An attempt to apply the rules of maxima and minima to several problems of stability related to architecture”. CA Coulomb. In: *Mémoires de l'Académie Royale des Sciences*. Vol. 7, 1776, pp. 343–382.
- [ArtDL76] “Inequalities in Mechanics and Physics Springer Verlag Berlin”. G Duvaut and JL Lions. In: . 1976, Heidelberg New York.
- [ArtDP52] “Soil mechanics and plastic analysis or limit design”. Daniel Charles Drucker and William Prager. In: *Quarterly of applied mathematics*. No. 2, Vol. 10, 1952, pp. 157–165. DOI: [10.1090/qam/48291](https://doi.org/10.1090/qam/48291).
- [ArtGer12] “8 - Modeling hydrogen induced damage mechanisms in metals”. W. Gerberich. In: . Vol. 1, 2012, pp. 209–246. Woodhead Publishing. DOI: [10.1533/9780857095374.2.209](https://doi.org/10.1533/9780857095374.2.209).
- [ArtHPF15] “Implantation de lois de comportement mécanique à l'aide de MFront: simplicité, efficacité, robustesse et portabilité”. Thomas Helfer, Jean-Michel Proix, and Olivier Fandeur. In: . 2015, Giens, France.
- [ArtJoh76] “Existence theorems for plasticity problems”. Claes Johnson. In: *Journal de Mathématiques Pures et Appliquées*. No. 4, Vol. 55, 1976, pp. 431–444. Gauthier-villars 120 Blvd Saint-germain, 75280 Paris, France.
- [ArtKoi53] “Stress-strain relations, uniqueness and variational theorems for elastic-plastic materials with a singular yield surface”. Warner Tjardus Koiter. In: *Quarterly of applied mathematics*. No. 3, Vol. 11, 1953, pp. 350–354. DOI: [10.1090/qam/59769](https://doi.org/10.1090/qam/59769).
- [ArtKoi60] “General theorems for elastic plastic solids”. Warner Tjardus Koiter. In: *Progress of Solid Mechanics*. 1960, pp. 167–221. North Holland Press.
- [ArtLL67] “Finite-strain elastic—plastic theory with application to plane-wave analysis”. EH Lee and DT Liu. In: *Journal of applied physics*. No. 1, Vol. 38, 1967, pp. 19–27. Aip. DOI: [10.1063/1.1708953](https://doi.org/10.1063/1.1708953).
- [ArtMar+08] “A numerical procedure simulating RC structures reinforced with FRP using the serial/parallel mixing theory”. Xavier Martinez, Sergio Oller, Fernando Rastellini, and Alex H Barbat. In: *Computers & Structures*. No. 15-16, Vol. 86, 2008, pp. 1604–1618. Elsevier. DOI: [10.1016/j.compstruc.2008.01.007](https://doi.org/10.1016/j.compstruc.2008.01.007).
- [ArtMis13] “Mechanik der festen Körper im plastisch-deformablen Zustand”. R v Mises. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*. Vol. 1913, 1913, pp. 582–592.
- [ArtMis28] “Mechanik der plastischen Formänderung von Kristallen”. R von Mises. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*. No. 3, Vol. 8, 1928, pp. 161–185. Wiley Online Library.
- [ArtMOB11] “Caracterización de la delaminación en materiales compuestos mediante la teora de mezclas serie/paralelo”. Xavier Martnez, Sergio Oller, and E Barbero. In: *Revista internacional de métodos numéricos para cálculo y diseño en ingeniera*. No. 3, Vol. 27, 2011, pp. 189–199. Elsevier.
- [ArtMT94] “A return mapping algorithm for isotropic elastoplasticity”. Anton Matzenmiller and Robert L Taylor. In: *International journal for numerical methods in engineering*. No. 5, Vol. 37, 1994, pp. 813–826. Wiley Online Library. DOI: [10.1002/nme.1620370507](https://doi.org/10.1002/nme.1620370507).
- [ArtPra24] “Spannungsverteilung in plastischen Körpern”. Ludwig Prandtl. In: . 1924, pp. 43–54.
- [ArtPra49] “Recent developments in the mathematical theory of plasticity”. William Prager. In: *Journal of applied physics*. No. 3, Vol. 20, 1949, pp. 235–241. Aip. DOI: [10.1063/1.1698348](https://doi.org/10.1063/1.1698348).
- [ArtPRH00] “Numerical differentiation for local and global tangent operators in computational plasticity”. Agustín Pérez-Foguet, Antonio Rodríguez-Ferran, and Antonio Huerta. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1, Vol. 189, 2000, pp. 277–296. Elsevier. DOI: [10.1016/s0045-7825\(99\)00296-0](https://doi.org/10.1016/s0045-7825(99)00296-0).
- [ArtRH15] “Automatic differentiation for stress and consistent tangent computation”. Steffen Rothe and Stefan Hartmann. In: *Archive of Applied Mechanics*. No. 8, Vol. 85, 2015, pp. 1103–1125. Springer. DOI: [10.1007/s00419-014-0939-6](https://doi.org/10.1007/s00419-014-0939-6).

- [ArtSai70] “Mémoire sur l'établissement des équations différentielles des mouvements intérieurs opérés dans les corps solides ductiles au delà des limites où l'élasticité pourrait les ramener à leur premier état”. Barre de Saint-Venant. In: *Compt. Rend.* Vol. 70, 1870, pp. 473–480.
- [ArtSim88a] “A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: Part I. Continuum formulation”. Juan C Simo. In: *Computer methods in applied mechanics and engineering*. No. 2, Vol. 66, 1988, pp. 199–219. Elsevier. DOI: [10.1016/0045-7825\(88\)90076-x](https://doi.org/10.1016/0045-7825(88)90076-x).
- [ArtSim88b] “A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. Part II: computational aspects”. Juan C Simo. In: *Computer methods in applied mechanics and engineering*. No. 1, Vol. 68, 1988, pp. 1–31. Elsevier. DOI: [10.1016/0045-7825\(88\)90104-1](https://doi.org/10.1016/0045-7825(88)90104-1).
- [ArtST85] “Consistent tangent operators for rate-independent elastoplasticity”. Juan C Simo and Robert Leroy Taylor. In: *Computer methods in applied mechanics and engineering*. No. 1, Vol. 48, 1985, pp. 101–118. Elsevier. DOI: [10.1016/0045-7825\(85\)90070-2](https://doi.org/10.1016/0045-7825(85)90070-2).
- [ArtZha16] “Technical problem identification for the failures of the liberty ships”. Wei Zhang. In: *Challenges*. No. 2, Vol. 7, 2016, p. 20. Multidisciplinary Digital Publishing Institute. DOI: [10.3390/challe7020020](https://doi.org/10.3390/challe7020020).

Ph.D.'s thesis

- [PhDBar16] “Numerical simulation of fatigue processes: application to steel and composite structures”. Lucia Gratiela Barbu. Universitat Politècnica de Catalunya. 2016.
- [PhDCor] “Coupled formulation between the finite element method and the discrete element method for studying multi-fracture processes in solids and structures”. Alejandro Cornejo.
- [PhDOll88] “Un modelo de daño continuo para materiales-friccionales”. Sergio Oller. 1988.

Chapter 6

Adaptative remeshing

“The measure of intelligence is the ability to change.”

Albert Einstein
(1879 - 1955 AD, German-born theoretical physicist)

6.1 Introduction

Adaptive remeshing of **FEM** solutions refers to improving the quality of the solutions by enriching the approximation in some manner so as to achieve the best solution for a given computational effort. The concept of adapting the mesh to improve the quality of the solution is summarised in Figure 6.1 and the conceptual steps on the Algorithm 5.

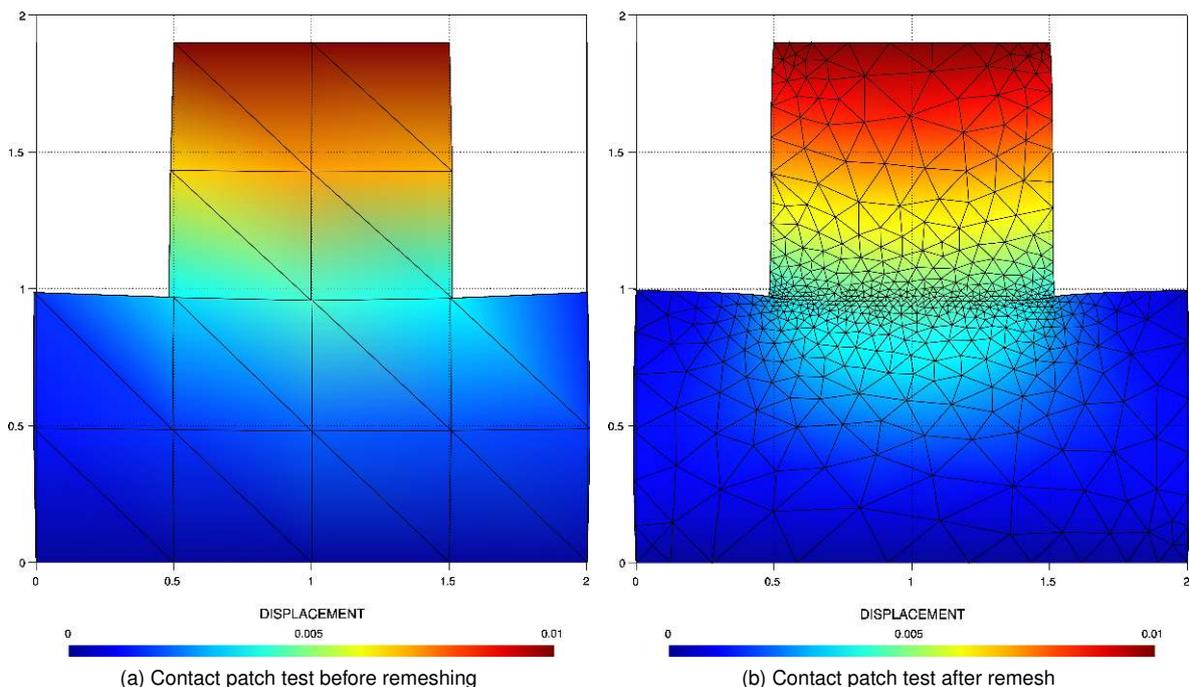


Figure 6.1: Adaptive mesh techniques applied on a contact patch test

Thus the objective of adaptive remeshing is to obtain a mesh which is optimal, in the sense that the computational costs involved are minimal under the constraint that the error in the **FE** solution is below a certain limit and, if possible, distributed over the entire mesh. In order to do that we need to estimate the magnitude of the error of the **FE** solutions. This is not trivial, as we do not know *a priori* the solution of the problem. Which means we need to develop techniques which can estimate the error of the finite element solution from the known data[BookWri08], this means, on the geometry, material data and our current approximate solution.

Algorithm 5 Adaptive refinement process

```

1: procedure ADAPTIVE REFINEMENT PROCESS
2:   Select an initial mesh, which approximates accurately the geometry
3:   while  $error > tolerance$  do
4:     Solve the discrete problem
5:     Compute error estimators or error indicators
6:     if  $error > tolerance$  then
7:       A new mesh has to be constructed
8:       The already computed deformations and internal variables have to be projected onto the new mesh

```

This chapter introduces the adaptive remeshing techniques developed in order to improve the quality of the solution. First the state-of-the-art of the respective techniques will be introduced on 6.2.State of the Art in mesh refinement, later the different approaches followed to define the mesh metrics, each one with its respective section: *Hessian* metric (6.3.Hessian based remeshing technique), *level set* metric (6.4.Level set based remeshing technique), next the **SPR** method is presented (6.5.SPR based remeshing technique), later a short section dedicated to the internal values interpolation techniques (6.6.Internal values interpolation) and finally an adaptation of the former methods applied on **CCM** (6.8.Adaptive remeshing methods applied on CCM). Part of this work has been published by Cornejo and Mataix[ArtCor+19].

6.2 State of the Art in mesh refinement

6.2.1 Mesh generation

During the last years, many techniques have emerged for this purpose. The early works from *O.C. Zienkiewicz*[ArtZP71] on 1970s, which was only based on the geometries boundaries of the domain size and the required distribution of element size¹. Since then many different technologies have arise, including mapping techniques, semi-automatic remeshing methods, where the domain had to be subdivided manually in an initial stage into simpler subdomains[Art-TWM82]. This methodology then deals with *know-how* dependency from the user, and that is generally limited for *structured meshes* (quadrilaterals and hexahedra).

Alternatively to the *structured meshes* algorithms can find the family of *unstructured meshes* techniques. In contrast with the previous ones, these methods deal with the called *simplex* geometries, corresponding to triangles in two dimensions and tetrahedra in three dimensions. These techniques are based mainly in four families of algorithms[BookZZT13; BookWri08]:

- *Delaunay* triangulation methods[ArtBow81].
- The advancing front method[ArtLo85].
- Tree methods, *quadtrees* for two dimensions and *octrees* for the three-dimensional cases[ArtYS84].
- The algorithm of *recursive region splitting*[ArtRRS95] places nodes on the boundary of the region to be meshed by considering a certain density distribution.

These techniques can be used additionally for the remesh of unstructured quadrilateral meshes with the combination of subsequent triangles, and the use of octree meshes for the automatic generation of hexahedric meshes[BookZZT13; ArtPJR17; ArtPVR18].

¹Isoparametric mapping method

6.2.2 Adaptive finite element refinement techniques

Mesh adaptation is now widely used in numerical simulations to improve the accuracy of the solutions as well as to capture the behaviour of physical phenomena[ArtFA05]. This technique allows to reduce considerably the computational cost, associated with the corresponding **DOF** reduction, but obtaining an accurate solution nevertheless[ArtAF03].

This allows to compute complex problems with good results in three dimensions without dealing with the problematic of the initial remeshing during pre-process, which can be a time consuming and *error-prone*[BookZZT13] task. Additionally, allowing to adapt the computation during the simulation evolution, which alternative implies the creation during the first stage a mesh that fits all the problem evolution, which can be *a priori* not known.

The process is called adaptation due to the existing dependence nature, the process depends on the previous results[BookZZT13]; these methodologies were introduced originally by *Babuka and Rheinboldt*[ArtBR78][ArtBR79] on the late 70s. The reasons for doing these adaptive procedures can be summarised, for solid mechanics problems, displacements and stress errors should be contained in a certain tolerance. A very common approach of this is the computation of the error energy norm and prescribing a threshold in order to not exceed a certain percentage of the total energy norm.

In order to doing so several techniques exist, like the **Recovery by Equilibrium of Patches (REP)** techniques. One of the most extended techniques based on **REP** is the **SPR** by *Zienkiewicz and Zhu* [ArtZZ92]. Of course the methodologies based on **REP** are not the only ones available to measure the error, also from *Zienkiewicz and Zhu* we can find techniques as the *recovery methods*[ArtZZ87]. From *Babuka and Rheinboldt*[ArtBR78] we can find the *residual-based methods*, these methodologies as the name suggest estimate the error from the residual obtained on the finite element approximation. We can also find relevant contributions in the field of mesh adaptability based on the solution error in the work of *Oñate and Bugeda*[ArtOB93; ArtOña+06; ArtBO94]. To study on detail all the techniques previously mentioned we address to the book of *E.Stein et al.*[BookRam01], a complete compendium of works from different authors addressing a set of complex **NL** problems.

Recently techniques based on the *Hessian* have emerged[ArtWes+03], this requires that the variable to be used as error estimation to be twice continuously differentiable in the case we consider an analytical solution, or otherwise requires to us to be able to numerically estimate this second derivative. This methodology has additionally the advantage of giving a proper measure in order to create an **anisotropic** new mesh, **anisotropic** meshes are those that employ extremely narrow mesh elements. This type of meshes have the advantage to reduce the number of new elements necessary on regions where the minimal size required would imply a high number of elements in case of considering an isotropic mesh. These last techniques are the ones consider on this study and the theoretical background will be expanded on the theory section (6.3.Hessian based remeshing technique).

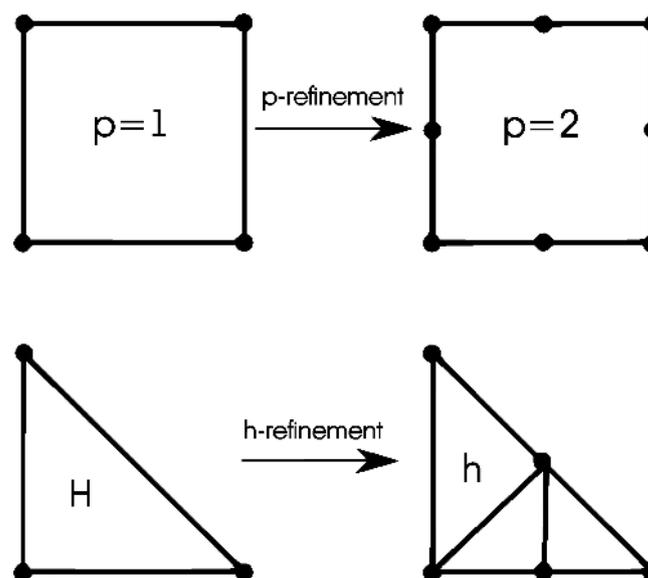


Figure 6.2: Concept of **h-refinement** and **p-refinement**

Finally, we can identify several procedures for the finite element mesh refinement[BookWri08], the most common

are the following four:

- **h-refinement:** The same type of elements is used but their size is changed. See Figure 6.2. This method can be divided at the same time on:
 - *Enrichment:* Also called element division. At the same time, we can distinguish between *hanging nodes* and *transition elements* methods.
 - *Remeshing:* A complete new mesh is generated.
- **p-refinement:** Where the same size of element is considered, but the polynomial order of the elements is altered hierarchically. See Figure 6.2. Here the size of the local matrices of the elements is increased, with it corresponding computational cost in computing and assembling.
- **hp-refinement:** This last technique consists in a combination of the two previous methods.
- **r-refinement:** Replaces finite element nodes such that the element sizes are optimised within a **FE** mesh in order to reduce the overall error of the **FE** solution. The convergence to the exact solution for this method is not guaranteed[BookZZT13].

The methodology considered on this work corresponds with the **h-refinement**, in particular we will consider the remeshing technique, which means as already introduced, that we will create a whole new mesh. In order to do so we will consider the *Mmg* library, an **anisotropic** metric based remeshing library. For more details we will address the corresponding section in the appendixes [B.3.Mmg library](#).

6.3 Hessian based remeshing technique

In this section we analyse on detail the techniques considered for remeshing. In order to understand the concepts considered on the contact applied remeshing (6.8.2.Hessian metric), we need to introduce first the concepts of error applied to a general **PDE** (6.3.1.Error estimation), later the concept of metrics is introduced (6.3.2.Metric based remeshing) and general *Hessian* based metric measure (6.3.3.Hessian based metric measure). The section will end with the introduction of the transfer operators used for the damage.

6.3.1 Error estimation

6.3.1.1 Upper bound on the interpolation error

We consider a general pde defined in a bounded domain Ω of \mathbb{R}^3 where \mathbf{u} denotes the exact solution of the problem. We will call to \mathbf{u}_h the **FE** solution obtained in a certain mesh \mathcal{T}_h , and therefore we can define the solution error \mathbf{e}_u as the difference between these two values, (6.1). Using this error, we desire to generate a new mesh \mathcal{T}_h on which the error \mathbf{e}_u computed on this mesh new mesh, is bounded by a given tolerance value.

$$(6.1) \quad \mathbf{e}_u = \mathbf{u} - \mathbf{u}_h$$

Usually, the approximation error is usually significantly difficult to quantify, and depends intrinsically of the nature of the **PDE** which define the problem. Because of this is interesting to define an indirect approach[ArtFA05] which allows us to estimate \mathbf{e}_u . If we define $\Pi_h \mathbf{u}$ as the interpolation of the solution \mathbf{u} in the mesh \mathcal{T}_h , therefore the interpolation error ($\tilde{\mathbf{e}}_u$) will be defined as the difference between the exact solution and this interpolated one, (6.2).

$$(6.2) \quad \tilde{\mathbf{e}}_u = \mathbf{u} - \Pi_h \mathbf{u}$$

The *Cea's lemma*[ArtCea64] it is an important tool for proving error estimates for the **FEM** applied to **elliptic PDE**, although it is possible to prove that this approach also works well for **hyperbolic PDE**.

$$(6.3) \quad \|\mathbf{u} - \mathbf{u}_h\| \leq c \|\mathbf{u} - \Pi_h \mathbf{u}\|$$

Where c is a constant independent of the current mesh \mathcal{T}_h . With this it is possible to control the approximation error e_u by controlling the interpolation error \tilde{e}_u . The solution \mathbf{u}_h converges towards \mathbf{u} as h is reduced. In any case the final objective is to reduce the number of **DOF** while preserving the desired level of accuracy for the **FE** solution. Therefore, the objective of the following approach concern[ArtFA05]:

- The construction of a suitable \tilde{e}_u estimate
- The definition of an **anisotropic** metric map
- The adaptation of a mesh to the sizing and stretching constraints of the metric map

Defining K as one element in the given mesh \mathcal{T}_h , with its corresponding vertices of the given geometry. For example, for the triangle $K = [a, b, c]$ and for a tetrahedron $K = [a, b, c, d]$. In here, the interpolated solution $\Pi_h \mathbf{u}$ coincides with \mathbf{u} in K . This interpolation is performed considering the standard **FE** shape functions². In the following the deduction will assume that we work with linear tetrahedra. In order to bound $\tilde{e}_u = \mathbf{u} - \Pi_h \mathbf{u}$ on K , we consider a **Taylor expansion** with integral rest of the function e at a vertex of K (in here we consider the vertex a as an example) with respect to any interior point x in K in (6.4).

$$(6.4) \quad (\mathbf{u} - \Pi_h \mathbf{u})(a) = (\mathbf{u} - \Pi_h \mathbf{u})(x) + \langle \vec{x}\vec{a}, \nabla(\mathbf{u} - \Pi_h \mathbf{u})(x) \rangle + \int_0^1 (1-t) \langle \vec{a}\vec{x}, \mathbf{H}(x + t\vec{x}\vec{a})\vec{a}\vec{x} \rangle dt$$

Where $\nabla \mathbf{u}(x)$ denotes the gradient of the variable \mathbf{u} , and $\mathbf{H}(x)$ denotes the *Hessian*, at the point x . Assuming that the maximal error is reached at the point x (closer to a than to the rest of the vertex K , a, b, c for tetrahedra), see (6.5a), and this is equivalent to (6.5b).

$$(6.5a) \quad \nabla(\mathbf{u} - \Pi_h \mathbf{u})(x) = 0$$

$$(6.5b) \quad \langle \vec{v}, \nabla(\mathbf{u} - \Pi_h \mathbf{u})(x) \rangle = 0, \quad \forall \vec{v} \subset K$$

Defining $e(x)$ as the error, defined as (6.5c). Considering a' as the point corresponding to the intersection of the line ax with the face opposite to a . Considering the local coordinate ξ such that $\vec{a}\vec{x} = \xi \vec{a}\vec{a}'$. As a is closest to x than any other vertex of K , then $\xi \leq 3/4$ ³. Therefore (6.5c) can be rewritten as (6.5d).

$$(6.5c) \quad |e(x)| = \left| \int_0^1 (1-t) \langle \vec{a}\vec{x}, \mathbf{H}(x + t\vec{x}\vec{a})\vec{a}\vec{x} \rangle dt \right|$$

$$(6.5d) \quad \begin{aligned} |e(x)| &= \left| \int_0^1 (1-t) \lambda^2 \langle \vec{a}\vec{a}', \mathbf{H}(a + t\vec{x}\vec{a})\vec{a}' \rangle dt \right| \\ &\leq \frac{9}{16} \max_{y \in aa'} |\langle \vec{a}\vec{a}', \mathbf{H}(y)\vec{a}\vec{a}' \rangle| \left| \int_0^1 (1-t) dt \right| \\ &\leq \frac{9}{32} \max_{y \in K} |\langle \vec{a}\vec{a}', \mathbf{H}(y)\vec{a}' \rangle| \end{aligned}$$

We can now consider the infinity norm (L_∞) of the interpolation error \tilde{e}_u , we obtain (6.5e). From this, the bound (6.5f) can be deduced.

²In case of a triangle, these shape functions can be defined as $\Pi_h u = (1 - \xi - \eta)u(a) + \xi u(b) + \eta u(c)$, with $0 \leq \xi + \eta \leq 1$. For the tetrahedron the shape functions are defined as $\Pi_h u = (1 - \xi - \eta - \zeta)u(a) + \xi u(b) + \eta u(c) + \zeta u(d)$, with $0 \leq \xi + \eta + \zeta \leq 1$.

³Remark, we are considering a tetrahedron in this deduction.

$$(6.5e) \quad \|\mathbf{u} - \Pi_h \mathbf{u}\|_{\infty, K} \leq \frac{9}{32} \max_{y \in K} \left| \langle \vec{aa}', \mathbf{H}(y) \vec{aa}' \rangle \right|$$

$$(6.5f) \quad \|\mathbf{u} - \Pi_h \mathbf{u}\|_{\infty, K} \leq \frac{9}{32} \max_{y \in K} \langle \vec{aa}', |\mathbf{H}(y)| \vec{aa}' \rangle$$

As in (6.5f) the bound depends on the extremum x , which is not known a priori, we can reformulate it as (6.5g). The former equation defines a bound in the case where the maximum error is achieved inside the element K . In the case where the maximum error is obtained on the element face, then the corresponding expression coincides with (6.5h). In the case where the maximum value is obtained along an element edge, the expression changes to (6.5i).

$$(6.5g) \quad \|\mathbf{u} - \Pi_h \mathbf{u}\|_{\infty, K} \leq \frac{9}{32} \max_{y \in K} \max_{\vec{v} \subset K} \langle \vec{v}, \mathbf{H}(y) | \vec{v} \rangle$$

$$(6.5h) \quad \|\mathbf{u} - \Pi_h \mathbf{u}\|_{\infty, K} \leq \frac{2}{9} \max_{y \in [a, b, c]} \max_{\vec{v} \subset [a, b, c]} \langle \vec{v}, |\mathbf{H}(y)| \vec{v} \rangle$$

$$(6.5i) \quad \|\mathbf{u} - \Pi_h \mathbf{u}\|_{\infty, K} \leq \frac{1}{8} \max_{y \in ab} \langle \vec{ab}, |\mathbf{H}(y)| \vec{ab} \rangle$$

As a summary, that will be taken into consideration as a base for deductions in the next section, relation (6.5f) provides a proper bound for the interpolation error on an element K .

6.3.1.2 Numerical computation of the interpolation error

Taking (6.5f), where the upper bound on the interpolation error ($\tilde{\mathbf{e}}_u$), we can reformulate it as (6.6a), in here c_d is a constant related to the space dimension. The former expression has some drawbacks, as the two maximum values required are complex to compute.

$$(6.6a) \quad \|\mathbf{u} - \Pi_h \mathbf{u}\|_{\infty, K} \leq c_d \max_{x \in K} \max_{\vec{v} \subset K} \langle \vec{v}, \mathbf{H}(x) | \vec{v} \rangle$$

Because of that, we introduce the edges of K to replace the maximum value associated with all vectors included in K . As any vector \vec{v} of K can be written as linear combination of the edges of K , it yields (6.6b), where E_K is the set of edges of K .

$$(6.6b) \quad \forall \vec{v} \subset K, \quad \|\vec{v}\|_{|\mathbf{H}(x)|} \leq \max_{e \in E_K} \|\vec{e}\|_{|\mathbf{H}(x)|}$$

This allows us to rewrite the upper bound from (6.6a) as (6.6c). In this expression the RHS term is not trivial to numerically evaluate. So we must assume the existence of a metric tensor $\tilde{\mathcal{M}}(K)$ as defined in (6.6d). This metric $\tilde{\mathcal{M}}$ is such that the region defined by $\{\langle \vec{v}, \tilde{\mathcal{M}}(K) \vec{v} \rangle | \forall \vec{v} \subset K\}$ is minimal in volume.

$$(6.6c) \quad \|\mathbf{u} - \Pi_h \mathbf{u}\|_{\infty, K} \leq c_d \max_{x \in K} \max_{e \in E_K} \langle \vec{e}, \mathbf{H}(x) | \vec{e} \rangle$$

$$(6.6d) \quad \max_{x \in K} \langle \vec{e}, \mathbf{H}(x) | \vec{e} \rangle \leq \langle \vec{e}, \tilde{\mathcal{M}}(K) \vec{e} \rangle, \quad \forall e \in E_K$$

Finally, the interpolation error ε_K on an element K is given by the following relationship (6.6e).

$$(6.6e) \quad \varepsilon_K = c \max_{\mathbf{e} \in E_K} \langle \vec{\mathbf{e}}, \vec{\mathcal{M}}(K) \vec{\mathbf{e}} \rangle$$

The former relation relates the interpolation error ε_K to the square of the largest edge length in K with respect to the metric $\vec{\mathcal{M}}$. This implies that it is possible to control the interpolation error on the mesh \mathcal{T}_h by controlling the length of the mesh edges in K .

6.3.1.3 Mesh adaptation

From (6.6e), which defines ε_K , we can deduce that if the ε_K is fixed, the only remaining variable left to change is the edge length of K . The objective will be adapt the mesh edges in order that the interpolation error will be equidistributed over the adapted mesh, meaning that we aim to obtain the desired error level with a minimum number of **DOF** under fixed constraints.

Considering ε the maximum level of error tolerated on the mesh elements. The mesh edges must be such that (6.7a).

$$(6.7a) \quad \varepsilon = c \langle \vec{\mathbf{e}}, \vec{\mathcal{A}}(K) \vec{\mathbf{e}} \rangle, \quad \forall \mathbf{e} \in E_K$$

Considering (6.7b) the target metric tensor, this expression leads to (6.7c).

$$(6.7b) \quad \mathcal{M}(K) = \frac{c}{\varepsilon} \vec{\mathcal{M}}(K)$$

$$(6.7c) \quad \langle \vec{\mathbf{e}}, \mathcal{M}(K) \vec{\mathbf{e}} \rangle = 1, \quad \forall \mathbf{e} \in E_K$$

(6.7c) is equivalent to define the edge length with respect to the metric $\mathcal{M}(K)$. In order to evaluate the length, we need to consider the *Euclidean* norm of a vector \vec{u} (L_2) for a metric \mathcal{M} is defined as (6.7d), and the distance between two points given by (6.7e).

$$(6.7d) \quad \|\vec{u}\|_{\mathcal{M}} = \sqrt{\langle \vec{u}, \vec{u} \rangle_{\mathcal{M}}} = \sqrt{t \vec{u} \cdot \mathcal{M} \vec{u}}$$

$$(6.7e) \quad d_{\mathcal{M}}(A, B) = l_{\mathcal{M}}(\vec{AB}) = \|\vec{AB}\|_{\mathcal{M}} = \sqrt{t \vec{AB} \cdot \mathcal{M} \vec{AB}}$$

In consequence, the relation (6.7b) is equivalent to (6.7f), which prescribes edges of unit length for any edge of K in order to bound the interpolation error ε_K on K by a value ε .

$$(6.7f) \quad (l_{\mathcal{M}(K)}(\vec{\mathbf{e}}))^2 = 1$$

The equation (6.7f) introduces the notion of unit mesh as the target optimal mesh. An optimal mesh is the mesh on which the interpolation error ε_K is distributed uniformly and bounded by a tolerance value ε .

6.3.1.4 Computation of the relative error

Finally, the relation (6.6c) gives us an absolute bound on the interpolation error ε_K . [ArtFA05] considers a relative bound on this error and therefore defining an estimate of the relative error. Doing this will provide a metric independent of the nature of the physical problem, this allows to combine different variables together. Equation (6.8a) introduces

how to compute dimensionless variables and introduces a relative error. We will present our own normalisation procedures in the following sections.

$$(6.8a) \quad \left\| \frac{\mathbf{u} - \Pi_h \mathbf{u}}{|\mathbf{u}|_\epsilon} \right\|_{\infty, K} \leq c_d \max_{x \in K} \max_{\vec{e} \in E_K} \left\langle \vec{e}, \frac{|\mathbf{H}(x)|}{|\mathbf{u}(x)|_\epsilon} \vec{e} \right\rangle$$

In here $|\mathbf{u}|_\epsilon = \max(|\mathbf{u}|, \epsilon \|\mathbf{u}\|_{\infty, \Omega})$. Additionally we must take into account that the solutions vary from several orders of magnitude. It is often difficult to capture the weakest phenomena via mesh adaptation. [ArtFA05] proposes as a potential solution the consideration of a local error estimation in order to overcome this problem. Additionally [ArtCIA07; ArtCas+97], the error estimate can also be normalised using the local value of the gradient (∇) norm (L_2) of the variable \mathbf{u} , weak phenomena can be captured even in cases where the variables deal with sudden variations, like shocks or BC dependent variables. In [ArtFA05] the following error estimate (6.8b) is proposed. Where h is the element size on the mesh \mathcal{T}_h and $0 < \alpha < 1$.

$$(6.8b) \quad \left\| \frac{\mathbf{u} - \Pi_h \mathbf{u}}{\alpha |\mathbf{u}|_\epsilon + h \|\nabla \mathbf{u}\|_2} \right\|_{\infty, K} \leq c \max_{x \in K} \max_{\vec{e} \in E_K} \left\langle \vec{e}, \frac{|\mathbf{H}(x)|}{\alpha |\mathbf{u}(x)|_\epsilon + h \|\nabla \mathbf{u}(x)\|_2} \vec{e} \right\rangle$$

6.3.2 Metric based remeshing

With the concepts introduced in 6.3.1. [Error estimation](#) we are now ready to introduce the *Hessian* metric [ArtFA05]. In here, we will introduce in first place the concept of metric (6.3.2.1. [Concept of metric](#)), then we will show the intersection operations needed in case than more than one metric is taken into consideration (6.3.2.2. [Metric intersection](#)).

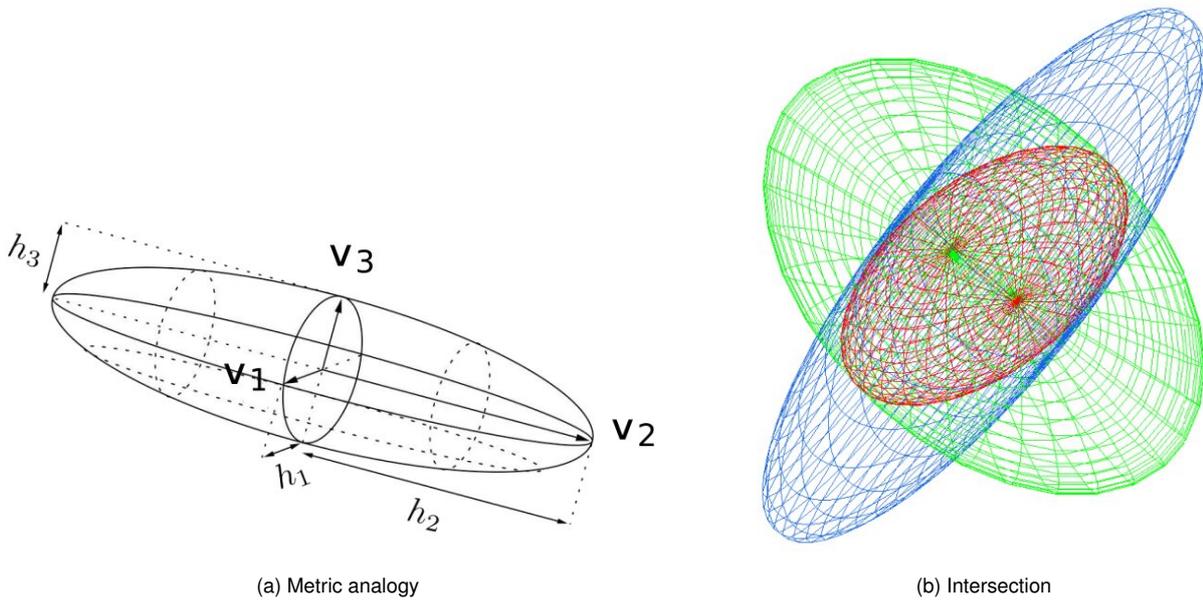


Figure 6.3: Metric analogies. Images from [ArtFA05]

6.3.2.1 Concept of metric

The metric represents the operator which is equivalent to a geometric interpolation law on h on the edge [ArtFA05]. Let \mathcal{M} be a discrete metric field defined at the vertices of a mesh \mathcal{T}_h of a domain Ω . Thanks to the interpolation operation, we have a continuous metric field in the whole domain, i.e. a *Riemannian* metric space $(\mathcal{M}(x))_{x \in \Omega}$. This representation of the metric field depends on \mathcal{T}_h as the interpolation law is applied at the element level.

In order to compute the length, we can consider two points in the domain, \mathbf{a} and \mathbf{b} . The length of the segment between this two points can be computed as *Gaussian* integration as seen in (6.9), where α_i corresponds with the

barycentric coefficients and ω_i the integration weight. In here where $\mathcal{M}(\mathbf{a} + \alpha_j \mathbf{ab})$ is the metric at the i^{th} Gauss point. With the former expression, the concept of metric can be understood.

$$(6.9) \quad \ell_{\mathcal{M}}(\mathbf{ab}) = \int_0^1 \sqrt{t_{\mathbf{ab}} \mathcal{M}(\mathbf{a} + t \mathbf{ab})} dt \approx \sum_{i=1}^k \omega_i \sqrt{t_{\mathbf{ab}} \mathcal{M}(\mathbf{a} + \alpha_i \mathbf{ab})}$$

The notion of length in a metric space is related to the notion of metric [ArtAF03] and therefore to an adequate definition of the scalar product in the vector space considered. Define a metric tensor at a point P , respect an element K , from a mesh \mathcal{T}_h it considering a matrix \mathcal{M} ($d \times d$) symmetric positive defined and not degenerated. In three dimensions the we can consider (6.10), which can be assimilated to the analogy of an ellipsoid (Figure 6.3a).

$$(6.10) \quad \mathcal{M} = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \text{ such that } a > 0, d > 0, f > 0$$

and $\det(\mathcal{M}) > 0$, considering $a, b, c, d, e \in \mathbb{R}$

The tensor \mathcal{M} can be diagonalised because it is symmetrical. Then, \mathcal{M} can be written $\mathcal{M} = \mathcal{R} \mathbf{\Lambda} \mathcal{R}^{-1}$, where \mathcal{R} is the matrix of the eigenvectors and $\mathbf{\Lambda}$ the matrix of the eigenvalues of \mathcal{M} . It should be noted that this decomposition is not unique, indeed one can take any eigenvector of the subspace associated with one of the eigenvalues or it is enough to invert columns of \mathcal{R} and $\mathbf{\Lambda}$.

To illustrate the effect of the metric on the actual mesh, we can observe the Figure 6.4, where the metrics present on the nodes sketch the tetrahedra accordingly.

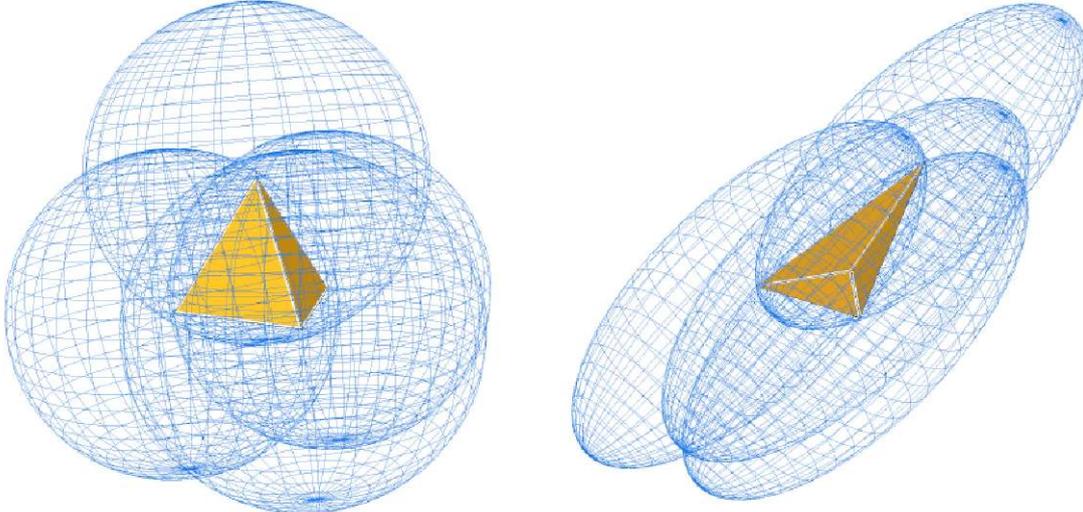


Figure 6.4: Effects of the metric on a tetrahedra

6.3.2.2 Metric intersection

Frequently, one is brought to work with several metrics. In the case where several metrics are specified at the same point of the mesh, we try to define a single metric that takes into account all these metrics, that is to say to define the metric given by the intersection of all these metrics.

To define the intersection of two metrics, we use the fact that a metric tensor is represented geometrically by a two-dimensional ellipse or a three-dimensional ellipsoid. The metric intersection consists then in keeping the most restrictive size constraint in all directions imposed by this set of metrics [ArtFA05] (see Figure 6.8b).

The simultaneous reduction enables to find a common basis (\mathbf{e}_1 , \mathbf{e}_2 and \mathbf{e}_3) such that \mathcal{M}_1 and \mathcal{M}_2 are congruent

to a diagonal matrix, in this basis then \mathcal{N} is introduced:

$$(6.11a) \quad \mathcal{N} = \mathcal{M}_1^{-1} \cdot \mathcal{M}_2$$

\mathcal{N} can be diagonalised in \mathcal{B} , because it is symmetrical in the metric \mathcal{M}_1 . The base in question is given by the normalised eigenvectors of \mathcal{N} that we note \mathbf{e}_1 , \mathbf{e}_2 and \mathbf{e}_3 (they form a base because \mathcal{N} is diagonalisable). The eigenvalues of \mathcal{M}_1 and \mathcal{M}_2 are found in this base using the *Rayleigh* quotient:

$$(6.11b) \quad \lambda_i = \mathbf{e}_i^t \cdot \mathcal{M}_1 \mathbf{e}_i \text{ and } \mu_i = \mathbf{e}_i^t \cdot \mathcal{M}_2 \mathbf{e}_i$$

Considering $\mathcal{P} = (\mathbf{e}_1 \mathbf{e}_2 \mathbf{e}_3)$ be the matrix the columns of which are the eigenvectors of \mathcal{N} , common basis of both metrics, so it can diagonalise at the same time \mathcal{M}_1 and \mathcal{M}_2 .

$$(6.11c) \quad \begin{aligned} \mathcal{M}_1 &= \mathcal{P}^{-t} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \mathcal{P}^{-1} \\ \mathcal{M}_2 &= \mathcal{P}^{-t} \begin{pmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{pmatrix} \mathcal{P}^{-1} \end{aligned}$$

Computing the metric intersection as:

$$(6.11d) \quad \mathcal{M}_{1 \cap 2} = \mathcal{M}_1 \cap \mathcal{M}_2 = \mathcal{P}^{-t} \begin{pmatrix} \max(\lambda_1, \mu_1) & 0 & 0 \\ 0 & \max(\lambda_2, \mu_2) & 0 \\ 0 & 0 & \max(\lambda_3, \mu_3) \end{pmatrix} \mathcal{P}^{-1}$$

6.3.3 Hessian based metric measure

Taking as starting point the concepts introduced in 6.3.1. [Error estimation](#), and the concepts of metrics from previous sections, we are now ready to introduce the *Hessian* based metric (\mathcal{M}). We can summarise the following properties introduced by *Alauzet*[ArtAF03]:

- The analysis and results obtained are not asymptotic, which means that the size of the mesh h is not tending to zero. This solves some potential errors.
- It is based in the *Hessian* (\mathbf{H}) of the solution.
- The obtained metric is **anisotropic**. This leads to reduce the number of degrees of freedom to solve the problem as compared to the equivalent isotropic mesh, for the same level of accuracy[ArtFA05].
- It is independent of the nature of the operator, so it can be used with any type of equation.

6.3.3.1 Hessian theory

We can compute the *Hessian*[ArtWes+03] matrix (\mathbf{H}) of a scalar variable f as proposed in Equation (6.12).

$$(6.12) \quad \mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \text{ or just:}$$

$$\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

We remark the fact that the mathematical concept underlying in the consideration of the *Hessian* in mesh adaptation has been already introduced in 6.3.1.3. *Mesh adaptation*. Once the *Hessian* matrix (\mathbf{H}) has been computed, we can compute the corresponding anisotropic metric by the following (6.13)[ArtFA05]. We highlight that for **anisotropic** remeshing the resulting metric \mathcal{M} , no additional operations are needed as the *Hessian* (\mathbf{H}) already provides an **anisotropic** mesh.

$$(6.13a) \quad \mathcal{M} = \mathcal{R}^t \tilde{\Lambda}^t \mathcal{R} \text{ with}$$

$$(6.13b) \quad \tilde{\Lambda} = \text{diag}(\tilde{\lambda}_i)$$

Being the eigenvalues from Equation (6.13) equivalent to the expression from Equation (6.13c). In here we will consider both the maximum (h_{max}) and minimum (h_{min}) desired mesh sizes.

$$(6.13c) \quad \tilde{\lambda}_i = \min \left(\max \left(\frac{c_d |\lambda_i|}{\epsilon}, \frac{1}{h_{max}^2} \right), \frac{1}{h_{min}^2} \right)$$

Being ϵ the error threshold and c_d a constant ratio of a mesh constant and the interpolation ratio⁴. For an isotropic mesh the metric (\mathcal{M}_{iso}) will be a diagonal matrix, where each element of the diagonal corresponds with the maximum eigenvalue as seen in Equation (6.13d), where \mathcal{R} corresponds with the eigenvectors matrix of \mathcal{M} .

$$(6.13d) \quad \mathcal{M}_{iso} = \text{diag}(\max(\tilde{\lambda}_i)) = \begin{pmatrix} \max(\tilde{\lambda}_i) & 0 & 0 \\ 0 & \max(\tilde{\lambda}_i) & 0 \\ 0 & 0 & \max(\tilde{\lambda}_i) \end{pmatrix}$$

On the other hand, we can enforce a higher level of anisotropy using the anisotropic ratio (ρ) on the mesh in consideration of a relative anisotropic radius ($R_{\lambda rel}$) as seen in the metric (\mathcal{M}_{aniso}) in Equation (6.13e).

$$(6.13e) \quad \mathcal{M}_{aniso} = \mathcal{R}^t \tilde{\Lambda}_{aniso} \mathcal{R}$$

Being the matrix $\tilde{\Lambda}_{aniso}$ equivalent to Equation (6.13f). In here we consider the minimum between the maximal eigenvalue and the corresponding eigenvalue. This value is at the same time compared with the relative radius, where the maximum between the two values is taken into consideration.

$$(6.13f) \quad \tilde{\Lambda}_{aniso} = \begin{pmatrix} \max(\min(\tilde{\lambda}_1, \tilde{\lambda}_{max}), R_{\lambda rel}) & 0 & 0 \\ 0 & \max(\min(\tilde{\lambda}_2, \tilde{\lambda}_{max}), R_{\lambda rel}) & 0 \\ 0 & 0 & \max(\min(\tilde{\lambda}_3, \tilde{\lambda}_{max}), R_{\lambda rel}) \end{pmatrix}$$

The relative radius then can be calculated using Equation (6.13g).

$$(6.13g) \quad R_{\lambda rel} = |\tilde{\lambda}_{max} - R_\lambda| \text{ where:} \\ R_\lambda = (1 - \rho) |\tilde{\lambda}_{max} - \tilde{\lambda}_{min}|$$

⁴ ϵ can be taken as 10^{-6} as reference value. In the other hand c_d can be considered as $\frac{2}{9}$ for 2D cases and $\frac{9}{32}$ for 3D cases.

6.3.3.2 Numerical example

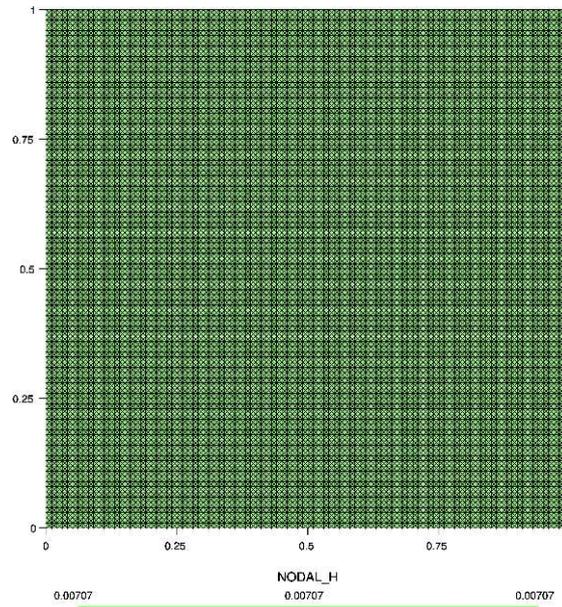


Figure 6.5: Initial mesh

The objective is to remesh the structured 1 mesh with the scalar function from Figure 6.5 and (6.14). The original mesh has a high number of elements structured, our objective is to obtain an unstructured mesh where the smaller elements will be around our objective function. The χ shaped function (6.14):

$$(6.14) \quad f(x, y) = \tanh(-100(y - 0.5 - 0.25 \sin(2\pi x))) + \tanh(100(y - x))$$

The results obtained can be seen on Figure 6.6b, here the smaller elements are around the χ shape that can be seen in the Figure 6.6a showing the scalar function representation.

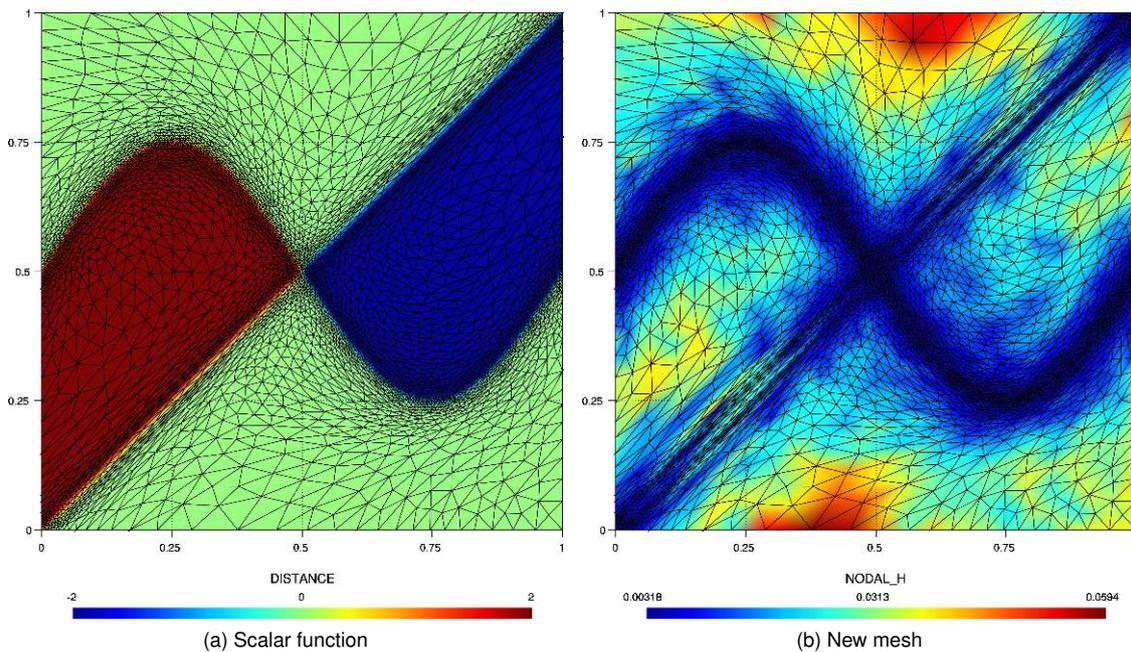


Figure 6.6: Solution for the example

6.4 *Level set* based remeshing technique

6.4.1 Theory

On the following section, the fundamentals of the *level set* technique for building a mesh metric are presented. The concepts of the metric already presented in the previous section (6.3.2. *Metric based remeshing*) are taken into account. We compute the gradient (6.15) of a scalar variable f in order to compute an anisotropic metric to remesh, using the procedure from (6.16). On Figure 6.7 the gradient of a distance function is presented as an example. The former example is shown in detail on the section 6.4.2. *Numerical example*.

$$(6.15) \quad \nabla f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k}$$

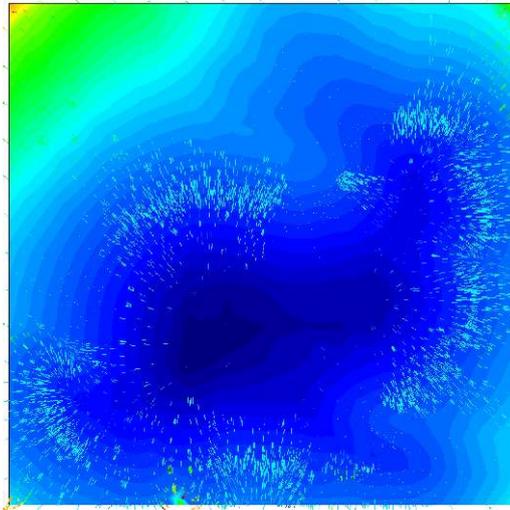


Figure 6.7: Scalar and its gradient

We compute the following auxiliary coefficients (6.16), with them we can calculate the metric for the 2D cases (6.16c) and for 3D cases (6.16d). Calling h the element size, ρ the anisotropic ratio, the metric \mathcal{M} , f the scalar value and ∇f the gradient from that scalar.

$$(6.16a) \quad \begin{cases} c_0 = \frac{1.0}{f^2} & \text{Isotropic metric} \\ c_1 = \frac{c_0}{\rho^2} & \text{Applying anisotropic ratio} \end{cases}$$

$$(6.16b) \quad \mathcal{M} = c_0 n \otimes n + c_1 (\mathbf{I} - n \otimes n), \text{ with } n = \frac{\nabla f}{\|\nabla f\|}$$

Then for **2D**:

$$(6.16c) \quad \mathcal{M} = \begin{pmatrix} c_0(1 - \nabla f_x^2) + c_1 \nabla f_x^2 & (c_1 - c_0) \nabla f_x \nabla f_y \\ (c_1 - c_0) \nabla f_x \nabla f_y & c_0(1 - \nabla f_y^2) + c_1 \nabla f_y^2 \end{pmatrix}$$

And for **3D**:

$$(6.16d) \quad \mathcal{M} = \begin{pmatrix} c_0(1 - \nabla f_x^2) + c_1 \nabla f_x^2 & (c_1 - c_0) \nabla f_x \nabla f_y & (c_1 - c_0) \nabla f_x \nabla f_z \\ (c_1 - c_0) \nabla f_x \nabla f_y & c_0(1 - \nabla f_y^2) + c_1 \nabla f_y^2 & (c_1 - c_0) \nabla f_y \nabla f_z \\ (c_1 - c_0) \nabla f_x \nabla f_z & (c_1 - c_0) \nabla f_y \nabla f_z & c_0(1 - \nabla f_z^2) + c_1 \nabla f_z^2 \end{pmatrix}$$

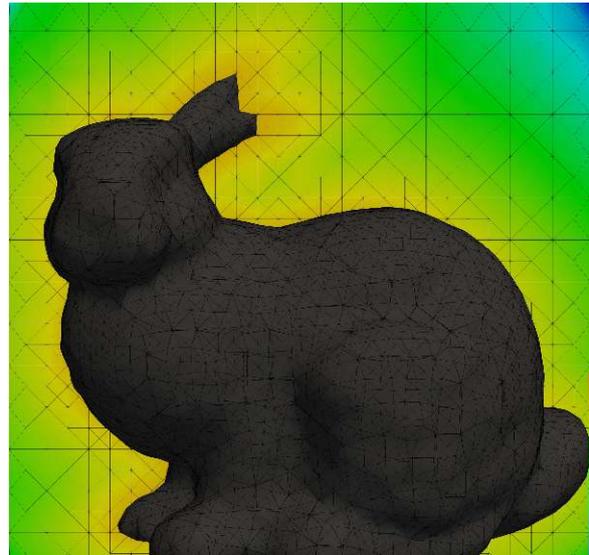
6.4.2 Numerical example

The challenge consists in meshing anisotropically the geometry using as metric measure the gradient of the distance, measured previously with an octree mesher *GiD*[OnlMel+16]. The geometry chosen is taken from the [Stanford 3D Scanning Repository](#), and it denominated *Stanford bunny*[BookRH12], see Figure 6.8a.

The mesh corresponding before remeshing corresponds with Figure 6.8b, the surrounding mesh to the isosurface is relatively coarse and does not fit the isosurface geometry.



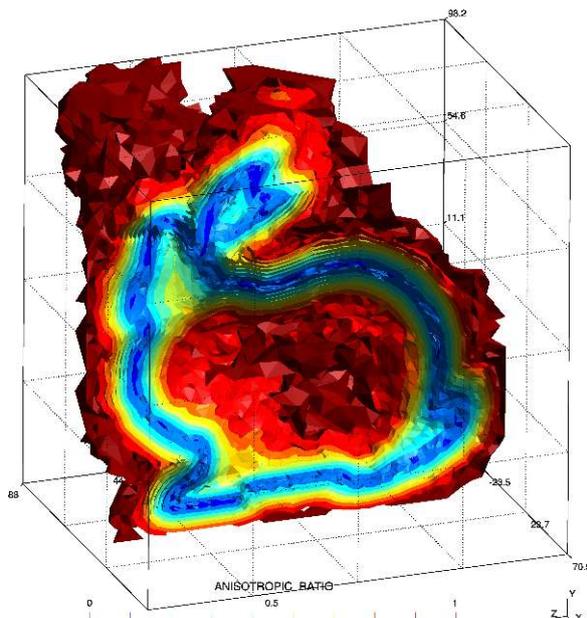
(a) Stanford's bunny



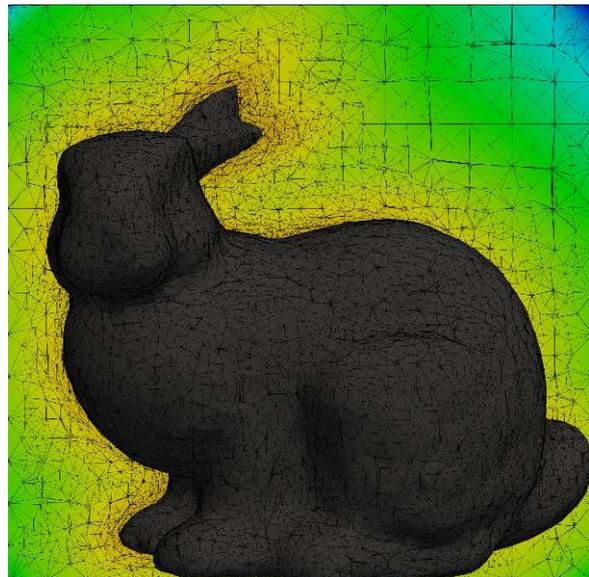
(b) Original octree mesh

Figure 6.8: Stanford's Bunny example

On Figure 6.9 the solution obtained is presented. In Figure 6.9a the anisotropy distribution (ρ) is represented in isosurfaces around the bunny. The resulting mesh, which fits the zero distance isosurface of the bunny, can be seen at Figure 6.9b.



(a) Anisotropic ratio



(b) Mesh after remeshing

Figure 6.9: Resulting mesh after remeshing

6.5 SPR based remeshing technique

6.5.1 Introduction

As we already mentioned previously, the **SPR** method was proposed by *Zienkiewicz* and *Zhu* in 1992[ArtZZ92], and continued with later contributions[ArtZZ95]. In order to present the method, the underlying concepts related with error measure and error estimation will be introduced. We remark that the basic concept of error estimation has been already introduced in 6.3.1. *Error estimation*, here we will extend those definitions.

6.5.2 Theory

6.5.2.1 Error measures

The most basic error measure (\mathbf{e}_{var}) would be to simply subtract the obtained solution in the **FEA** (\mathbf{u}_h) to the exact solution (\mathbf{u}), this can be extended to any value we want to use as reference like stresses or strains (6.17). Nonetheless these local error measures are in general not applicable for a refinement strategy as many singularities, like zero-size elements, may occur. Because of that various error norms ($\|\mathbf{e}\|_E^2$) representing an integral quantity are chosen to measure the error[BookZZT13], Equation (6.18) presents the energy norms as an example. It is relevant to mention that according to this concept, it is not always necessary for the mesh in the entire model to be refined. This can be seen once we consider the *Saint-Venants Principle*, which enforces that the local stresses in one region do not affect the stresses elsewhere.

$$(6.17) \quad \begin{aligned} \mathbf{e}_u &= \mathbf{u} - \mathbf{u}_h \\ \mathbf{e}_\varepsilon &= \varepsilon - \varepsilon_h \\ \mathbf{e}_\sigma &= \sigma - \sigma_h \end{aligned}$$

The global energy norm can be defined as (6.18a).

$$(6.18a) \quad \|\mathbf{e}\|_E^2 = \int_{\Omega} (\sigma - \sigma_h)^T \mathbb{C}^{-1} (\sigma - \sigma_h) d\Omega$$

The sum of the local energy norm of each element (K) contributions (6.18b).

$$(6.18b) \quad \|\mathbf{e}\|_E^2 = \sum_K \left[\int_{\Omega_K} (\sigma - \sigma_h)^T \mathbb{C}^{-1} (\sigma - \sigma_h) d\Omega \right] = \sum_K \|\mathbf{e}\|_{E,K}^2$$

6.5.2.2 Error estimation

Unfortunately, these errors measure is not possible to obtain in practice, as we do not know the exact solution most of the time, so we need to calculate an estimation of the error ($\hat{\mathbf{e}}_{var}$), in such a way that $\|\hat{\mathbf{e}}_{var}\| \approx \|\mathbf{e}_{var}\|$. Usually an *a posteriori* error estimators are used. These types of error estimators are calculated based on a reference solution of the problem which is usually obtained by a computation on a coarse mesh. On the other hand, an *a priori* error estimators estimate the error before a solution is calculated. Due to the nature of the present work, an *a posteriori* error estimator will be considered.

The *a posteriori* error estimators can be divided on[BookAO11]:

- **Residual based method:** These methods can be divided on:
 - **Explicit residual methods:** Explicit schemes involve a direct computation using available data. The explicit residual method employs the residuals in the current approximation.
 - **Implicit residual methods:** Implicit schemes involve the solution of an algebraic system of equations. The error is estimated via the element residual method indirectly, it generally involves the solution of a small linear algebraic system.

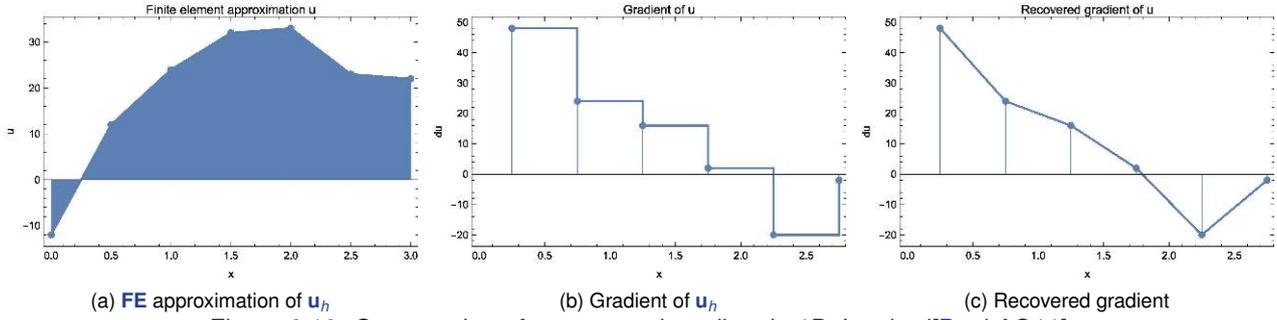


Figure 6.10: Construction of a recovered gradient in 1D. Inspired[BookAO11]

- **Recovery based methods:** In linear **FE** the stresses and strains are constant over the element, meaning that the stresses field will be discontinuous between elements. Taking these lower order approximations, recovery methods generate a higher-order solution (see Figure 6.11). **SPR** method falls inside this category.

Each of the presented method has its own strength and weaknesses[BookAO11]. Explicit residual methods require low computational effort, but requires the estimation of an unknown constant. Implicit residual methods require the resolution of an auxiliary problem, therefore higher computational cost, and additionally the *Neumann BC* are complex to take into account. On the other hand, recovery methods require low to very low computational costs, and the only weakness is that no upper and lower error bounds exist.

6.5.2.3 Recovery based methods

These methods were originally based on postprocessing procedures. We want to compute the nodal recovered stress values ($\tilde{\sigma}^*$) from the overall recovered stresses (σ^*) using the standard shape functions (**N**) (6.19).

$$(6.19) \quad \sigma^* = \mathbf{N} \tilde{\sigma}^*$$

It is assumed that the exact solution is approximated by the recovered solution sufficiently. In consequence we can evaluate the energy norm error from 6.18 replacing the exact stress with σ^* . The resulting expression (6.20) gives us an approximation of the energy norm.

$$(6.20a) \quad \|\mathbf{e}\|_E^2 = \int_{\Omega} (\sigma^* - \sigma_h)^T \mathbb{C}^{-1} (\sigma^* - \sigma_h) d\Omega$$

$$(6.20b) \quad \|\mathbf{e}\|_E^2 = \sum_K \left[\int_{\Omega_K} (\sigma^* - \sigma_h)^T \mathbb{C}^{-1} (\sigma^* - \sigma_h) d\Omega \right] = \sum_K \|\mathbf{e}\|_{E,K}^2$$

6.5.2.4 Superconvergence concept

The idea to derive these error indicators is based on the fact that many **FE** meshes have superconvergence properties[BookRam01]. Superconvergent points are points of the solution field where the error of the computed values decreases faster than elsewhere and those errors have an order $\mathcal{O}(h^{p+1})$. This concept can be illustrated on Figure 6.11, where the **FE** solution of the primary variable \mathbf{u} is actually exact at the nodes while the **FE** solution of the first derivative $\frac{du}{dx}$ is exact at the centre of the element. In this 1D case, the order of convergence is at least one order higher (order $p+1$) at the nodes for the primary variable and at the **GP** for secondary variables (order p) like stresses or strains[ArtZZ92]. This property cannot be applied on general, but can be shown that for problems of the form $\mathbf{B}^T \mathbb{C} \mathbf{B} \mathbf{u} = \mathbf{f}$ with a differential deformation operator **B** and a symmetric constitutive tensor \mathbb{C} this property holds.

By using a projection method, an improved stress field can be obtained from the superconvergent points. With the superconvergent values, it is possible to smoothing of such values by a polynomial of order p within a *patch* of

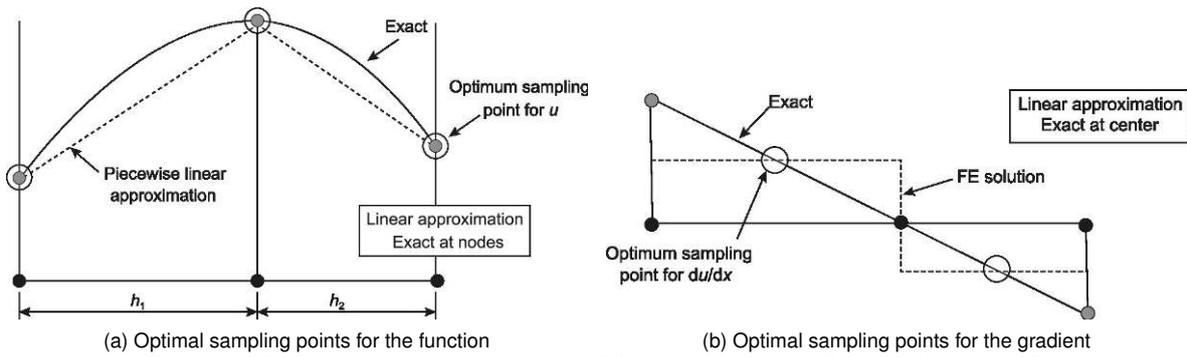


Figure 6.11: Optimal sampling points in 1D linear elements. Source[BookZZT13]

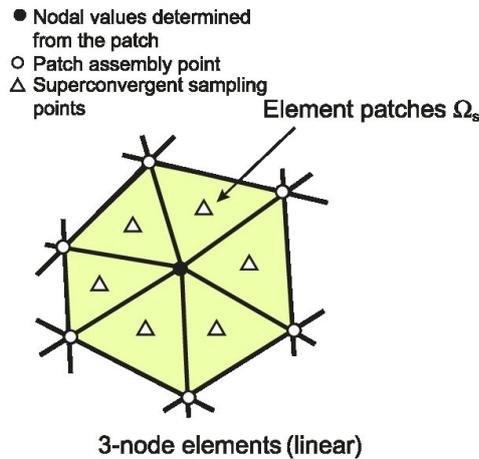


Figure 6.12: SPR example for linear triangles. Source[BookZZT13]

elements for which the number of sampling points can be taken as greater than the number of parameters in the polynomial. For *simplex* linear elements⁵, which are used in this work, it has been shown that the average value of the derivatives at mid sides of adjacent elements are superconvergent[ArtChu80; ArtLev82] (see Figure 6.12). Additionally in order to avoid the singularity on the LHS of the system we need to solve for the stress recovery, some additional precautions must be taken into account on the corner elements, with an incomplete patch. One possible way to deal with this issue is to calculate the recovered solution from a valid neighbour patch to the original patch. It is relevant to highlight that the optimal sampling points are considered the centroidal points for the construction of a recovered stress field[ArtZZ92], for further detail see[BookZZT13].

6.5.2.5 SPR calculation

Based on the procedure described in the previous section, we want to construct a solution field with superconvergent properties in the whole domain from the superconvergent sampling points. In order to do so, we will apply a smoothing of the stress values within an element patch considering a least-square polynomial fit. In the following, the concepts and procedures important for SPR-based error estimation are presented in (6.21).

Defining the polynomial of order p that will be considered on the recover. Being \mathbf{a}_i is a set of unknown parameters and \mathbf{p} contains appropriate polynomial terms. \bar{x} represents the coordinates respect the interior vertex node of the patch. Then we can deduce (6.21a).

$$(6.21a) \quad \begin{aligned} \bar{\sigma}_i^* &= \mathbf{p}\mathbf{a}_i \\ \mathbf{p} &= [1 \quad \bar{x} \quad \bar{y} \quad \bar{z} \quad \dots \quad \bar{z}^p] \\ \mathbf{a}_i &= [a_1 \quad a_2 \dots a_m]^T \end{aligned}$$

⁵Linear 2D triangular elements, and the linear tetrahedra in extension for 3D.

Our objective is to minimise the following (6.21b).

$$(6.21b) \quad \min_{\mathbf{a}_i} \Pi = \min_{\mathbf{a}_i} \frac{1}{2} \sum_{k=1}^n [\sigma_{h,i} - \bar{\sigma}_i^*]^2 = \min_{\mathbf{a}_i} \frac{1}{2} \sum_{k=1}^n [\sigma_{h,i} - \mathbf{p}\mathbf{a}_i]^2$$

This minimisation defines the following system of equations with **LHS** and **RHS**.

$$(6.21c) \quad \begin{aligned} \mathbf{LHS}\mathbf{a}_i &= \mathbf{RHS}_i \\ \mathbf{LHS} &= \sum_{k=1}^n \mathbf{p} \otimes \mathbf{p} \\ \mathbf{RHS}_i &= \sum_{k=1}^n \mathbf{p}\sigma_{h,i} \end{aligned}$$

The superconvergent nodal stresses may be computed by different patches adding and averaging its corresponding contribution to the shared nodes.

Finally, it is important to analyse the singularities on the patch, like the corner nodes, where the **LHS** from (6.21c) becomes singular. In order to solve that issue in [ArtZZ92] it is proposed to calculate the nodal superconvergent stress values for boundary nodes from an interior patch.

6.5.2.6 Local mesh size

6.5.2.6.1 Compute h :

With the previous error computed, we are now able to compute the element size h . In order to do so we apply the methodology from Zienkiewicz[BookZZT13], summarised in Equation (6.22).

The relative energy norm error (η) should be below a certain threshold value ($\eta_{threshold}$). We can consider the error estimated previously in all the following expressions. It is considered[BookZZT13] to be necessary to remesh in every entity where $\eta \leq 1$.

$$(6.22a) \quad \eta = \frac{\|\mathbf{e}\|_E}{\|\mathbf{u}\|_E} \approx \frac{\|\hat{\mathbf{e}}\|_E}{\|\mathbf{u}\|_E} \leq \eta_{threshold}$$

In the former expression, the total energy of deformation ($\|\mathbf{u}\|_E$) is expressed as (6.22b)⁶.

$$(6.22b) \quad \|\mathbf{u}\|_E^2 = \int_{\Omega} \boldsymbol{\sigma} \mathbb{C}^{-1} \boldsymbol{\sigma} d\Omega$$

With this we can define the threshold error ($\|\mathbf{e}\|_{threshold}$) as (6.22c).

$$(6.22c) \quad \|\mathbf{e}\|_{threshold} = \eta_{threshold} \|\mathbf{u}\|_E = \eta_{threshold} \sqrt{\|\mathbf{u}_h\|_E^2 + \|\mathbf{e}\|_E^2}$$

We are interested on distribute this error uniformly across all the n_K elements, then the threshold error for each element ($\|\mathbf{e}\|_{threshold,K}$) would be (6.22d).

$$(6.22d) \quad \|\mathbf{e}\|_{threshold,K} = \eta_{threshold} \sqrt{\frac{\|\mathbf{u}_h\|_E^2 + \|\mathbf{e}\|_E^2}{n_K}} \approx \eta_{threshold} \sqrt{\frac{\|\mathbf{u}_h\|_E^2 + \|\hat{\mathbf{e}}\|_E^2}{n_K}} := \bar{e}_K$$

With this we can define the refinement ratio for each element (ξ_K) as in (6.22e). We can assume that the error per element ($\|\mathbf{e}\|_K$) is proportional to the mesh size (h) to the polynomial power (6.22f). With the previous expression, and considering the refinement ratio for each element (ξ_K) the new element size (h_{new}) can be defined as (6.22g).

$$(6.22e) \quad \xi_K = \frac{\|\mathbf{e}_K\|}{\bar{e}_K}$$

$$(6.22f) \quad \|\mathbf{e}\|_K \propto h^p$$

$$(6.22g) \quad h_{new} = \frac{h}{\sqrt[p]{\xi_K}}$$

⁶ \mathbb{C}^{-1} implies that depending on the material considered, the method may not be applicable.

6.5.2.6.2 Compute respective metric :

The previously calculated element size must be transformed into the metric format, 6.3.2.Metric based remeshing, in order to be used with the *Mmg* library. In order to do so, as the size obtained corresponds with an isotropic mesh it will be as easy as a diagonal tensor, see (6.13d), such that we obtain the final expression from Equation (6.23).

$$(6.23) \quad \mathcal{M} = \delta_{ij} \frac{1}{h_{new}^2}$$

6.5.3 Example

We can illustrate the method with the following example, Figure 6.13. In here we have a simple cantilever beam with a distributed load. We start with the mesh from Figure 6.13a. Computing the energy error norm with the *SPR* the final mesh from Figure 6.13b is obtained. We can appreciate that the error is reduced significantly from Figures 6.13c to 6.13d. Additionally the displacement changes significantly too from Figures 6.13e to 6.13f with the finer mesh (the deformation factor is tweaked in order to fit better on these pages).

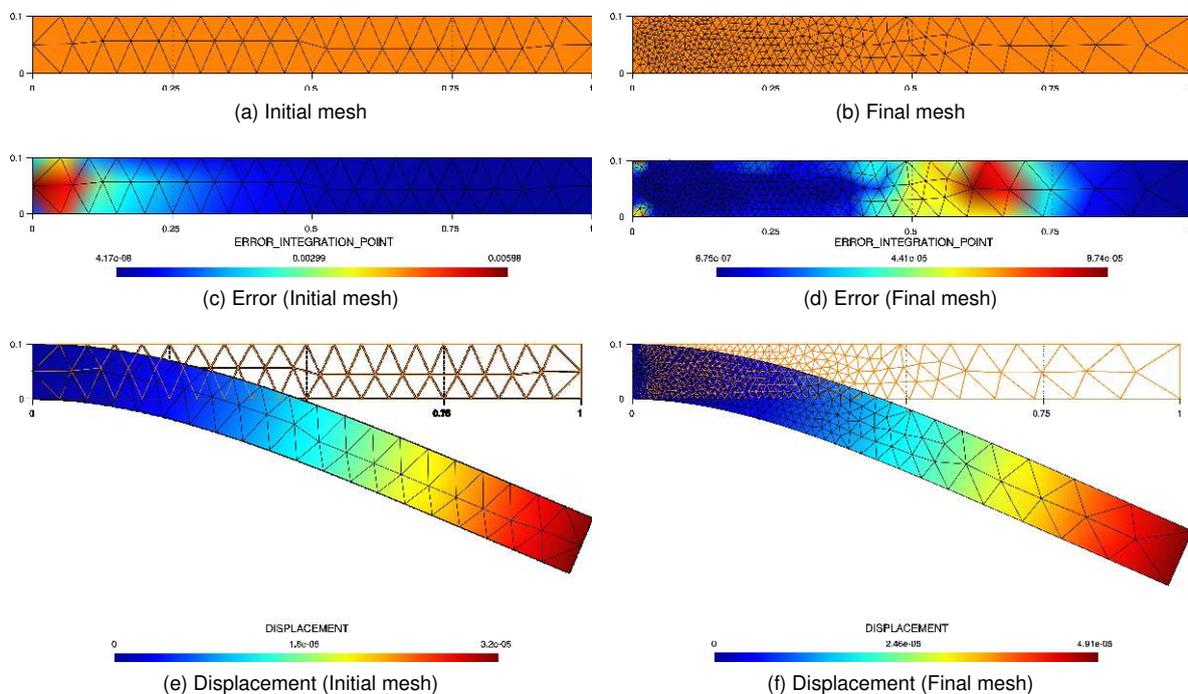


Figure 6.13: *SPR* example

6.6 Internal values interpolation

6.6.1 Theory

In order to recover the internal values information after remeshing, necessary in order to be able to work with constitutive models that depend on historical values, as the plasticity models used on this work. The Figure 6.14 shows graphically how each one of the *transfer operators* work [ArtGJ06], we can be listed as (all of them available on *Kratos*):

- **CPT** (Figure 6.14a): It just takes the value from the closest point. It provides acceptable results at low cost.
- **SFT** (Figure 6.14c): It interpolates the values using the standard **FEM** shape functions. Leads to an artificial diffusion of the value, but preserves the original shape of the value profile.

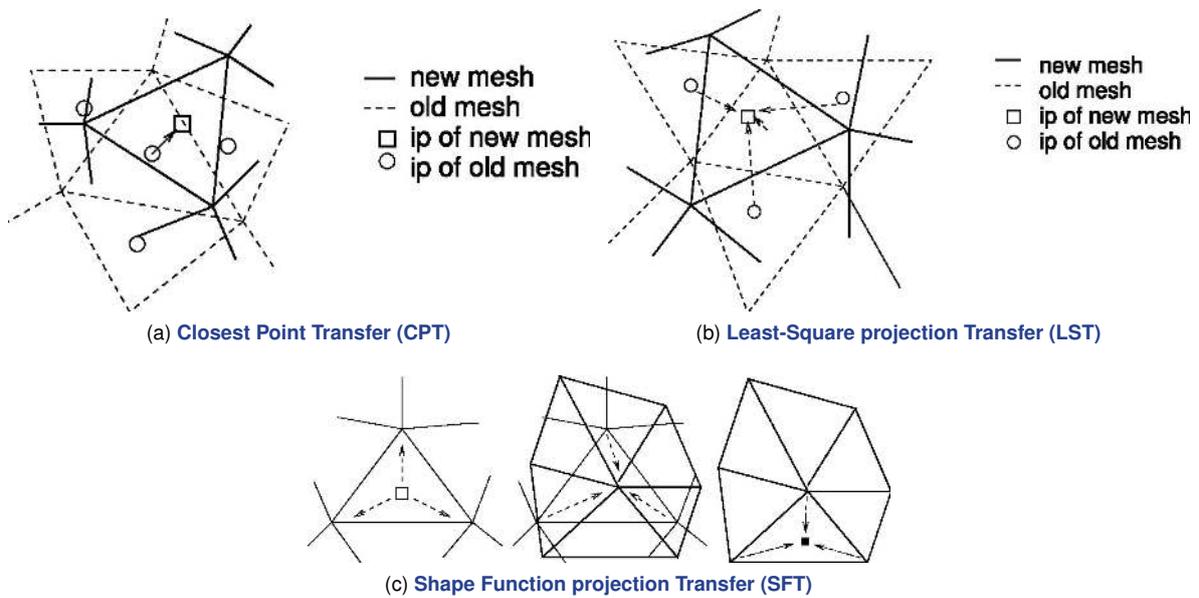


Figure 6.14: Transfer operators. Source[ArtGJ06]

- **LST** (Figure 6.14b): It considers a *least-square* transfer across the closest points. Probably the most accurate technique but computationally more expensive.

From the methodologies listed above the used in our simulations is the **CPT**, due to the fact that is the most commonly used on the literature[ArtChi+12]. There are more available techniques, we address the work of *Bussetta*[ArtBBP] for alternative mesh transfers operators.

6.6.2 Example

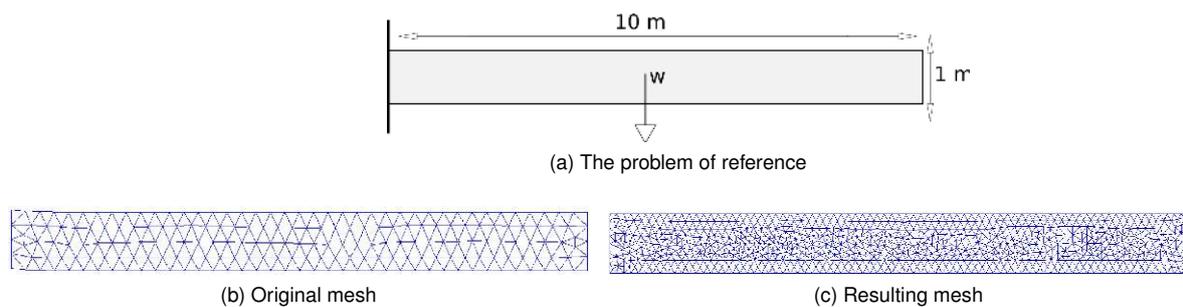
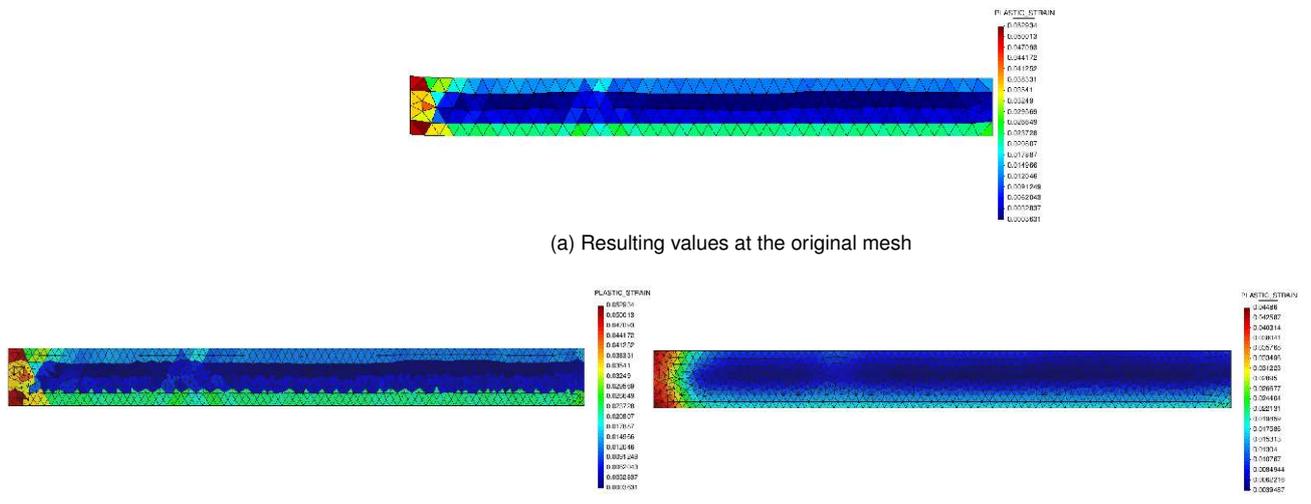


Figure 6.15: The beam problem of reference for the internal interpolation

The problem presented consists in a very simple beam of 10×1 as seen on Figure 6.15a. On this mesh we consider a simple plasticity law, and we wan to see if we are able to preserve the *plastic strain* between the initial mesh (Figure 6.15b) and the final mesh (Figure 6.15c).

The results we wan to interpolate are shown in Figure 6.16a. Considering the **CPT** method (Figure 6.16b) the resulting plastic strain profile is quite close to the original one. On the other hand, we have the results for the **LST** from Figure 6.16c, which solution is much smoother, but as previously stated this may lead to an artificial diffusion of the target value.

6.7 Integration points values extrapolation



(b) Interpolated results with CPT (c) Interpolated results with LST
 Figure 6.16: The beam problem of reference for the internal interpolation

6.7.1 Introduction

In order to be able to consider some GP as a candidate variable to apply the *level-set* (6.4. *Level set based remeshing technique*) or the *Hessian* (6.3. *Hessian based remeshing technique*) approach we need to transfer and extrapolate the values from the integration points to the nodes of the mesh. The technique required to do so is presented in the following.

6.7.2 Theory

Applying the same concepts considered for stress recovery [BookCA01], we can extrapolate the values from the integration points. Taking into account the concept of the integration values can be defined with the standard shape function definition and nodal values, Equation (6.24a). With this definition we can define a matrix operation which will allow to extrapolate locally the values from the GP, as seen in Equation (6.24b). In here n corresponds with the number of nodes in the element and m with the number of GP.

$$(6.24a) \quad value(\xi, \eta, \zeta) = [value_1 \quad \dots \quad value_n] \begin{bmatrix} N_1 \\ \vdots \\ N_n \end{bmatrix}$$

$$(6.24b) \quad \begin{bmatrix} value_1 \\ \vdots \\ value_n \end{bmatrix} = \begin{bmatrix} N_1^1 & \dots & N_1^m \\ \vdots & \ddots & \vdots \\ N_n^1 & \dots & N_n^m \end{bmatrix} \begin{bmatrix} value_1^{gp} \\ \vdots \\ value_m^{gp} \end{bmatrix}$$

The recovered values from the integration points locally calculated for each element, generally exhibit jumps between elements. In order to *smooth* these jumps, we can compute averaged nodal integration values. The two main methods in order to do so are:

1. **Unweighted averaging:** Where we assign the same weight to all elements that meet at a node
2. **Weighted averaging:** The weight assigned to element contributions depends on a certain value. For example, the integration weight of each GP, the element geometry or the element type

From these methods we will consider a *weighted averaging* based on the integration weights of each GP. In order to do so, the corresponding average value at each node will be the one shown in Algorithm 6.

Algorithm 6 Compute of nodal average value

```

1: procedure COMPUTE OF NODAL AVERAGE VALUE
2:   Reset the nodal average value on the mesh  $w_{node}$ 
3:   for all  $node \in Mesh_{nodes}$  do
4:      $w_{node} = 0$ 
5:   We add the contribution of each GP
6:   for all  $element \in Mesh_{elements}$  do
7:     for all  $GP \in element$  do
8:       Compute Jacobian  $J$ , integration weight  $w_{gp}$  and shape functions  $N$ 
9:       for all  $node \in element$  do
10:         $w_{node} += N_{node} J w_{gp}$ 

```

In order to compute the *smooth* result, we just need to sum the corresponding contribution of each GP multiplied by w_{node} , as seen in Equation (6.25).

$$(6.25) \quad value_{node} = \sum_{i=0}^{elem} \sum_{j=0}^{gp} \frac{N_{node} J w_{gp}}{w_{node}} value_{gp}$$

6.7.3 Numerical example

The following example consists in a very simple structural problem. I will consist in a variable height cantilever considering linear elastic plain strain constitutive model. The parameters considered are standard steel, see Table 6.1. Figure 6.18 shows the solution obtained. On Figure 6.18a we can see the VM stress before being recovered, and in Figure 6.18b the solution after consider the algorithm presented.

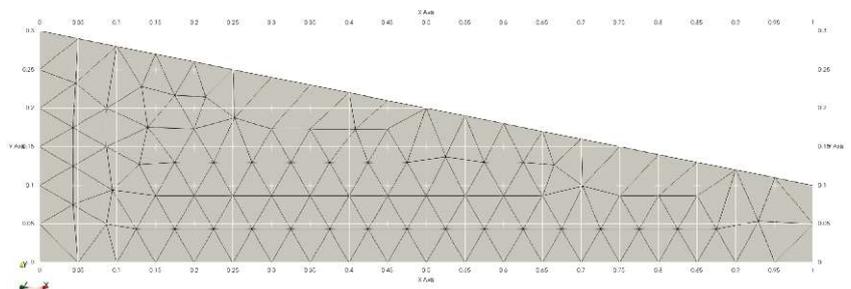


Figure 6.17: Cantilever mesh and geometry

E	ν	ρ
$2.069 \cdot 10^{11} Pa$	0.29	$7850 kg/m^3$

Table 6.1: Parameters considered for cantilever example

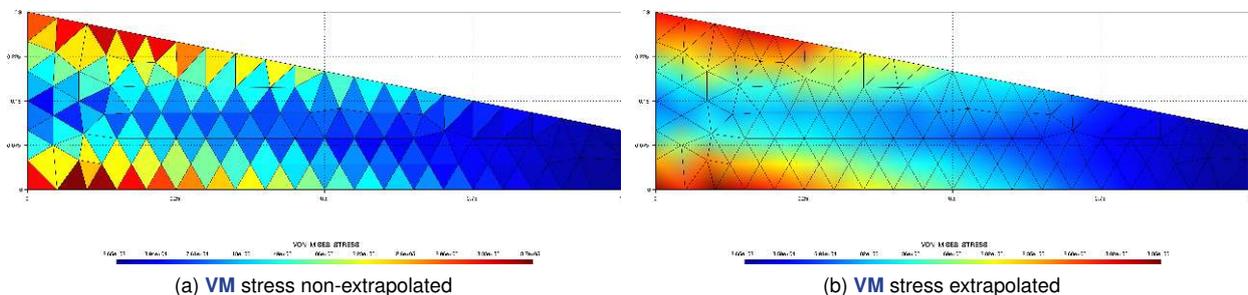


Figure 6.18: Resulting VM stress before and after recover

6.8 Adaptive remeshing methods applied on CCM

In this section we will introduce the particularities necessities in order to evaluate the remeshing techniques on the contact problems. The discontinuity and continuous evolution of the boundary, the nonlinearity of the problem itself,

introduce several challenges that must be taken into account in order to properly evaluate the problem.

Especially in the case of **CCM**, an adaptive refinement procedure is very beneficial: Particularly in the transition between the contact zone and the non-contact zone, the exact solution might not be covered appropriately by the **FE**-solution, for example due to a C^0 -continuity in the contact pressure. A fine mesh in this area gives the possibility to cover the contact behaviour appropriately while the computational costs are kept low due to a coarse mesh in areas where the solution behaves nicely and can be covered by fewer elements.

6.8.1 Level set metric

6.8.1.1 Theory

One first approach we can think in order to remesh the contact problem is the consideration of a *level set*, see 6.4. *Level set based remeshing technique*. This approach is adequate for fluid simulations were with this technique, it is possible to remesh in the boundary layer of the fluid . This is commonly used in **CFD**[ArtSAS13], particularly in embedded/immersed formulations, which is known to be crucial for accuracy. An equivalent to this last approach could be considering the equivalent to the distance gradient in **CCM**, which would be the gap gradient. In order to compute the consistent gap, we will consider the formulation proposed in 4.4.4. *Penetration definition*. In the following numerical example, we will study if this approach fits our needs.

6.8.1.2 Numerical example

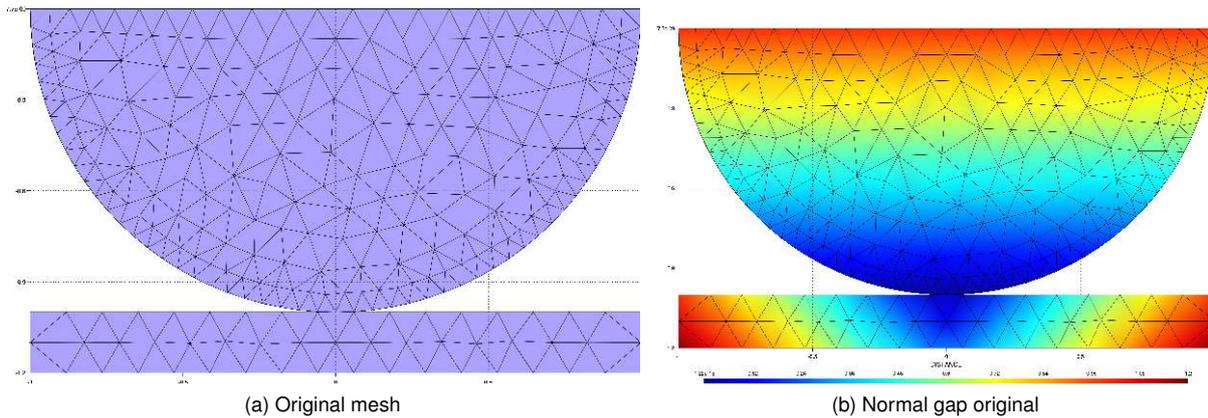


Figure 6.19: *Hertz* geometry considered. Example [here](#)

In order to study the suitability of the method, we can consider a relatively simple geometry so we can see if this solution fulfils our problematic. We will consider an *Hertz*-like geometry from Figure 6.19. This geometry is commonly considered as a benchmark in **CCM**, but additionally we can even compute analytically the gap (g_n). The initial mesh is on Figure 6.19a and the gap representation in the Figure 6.19b.

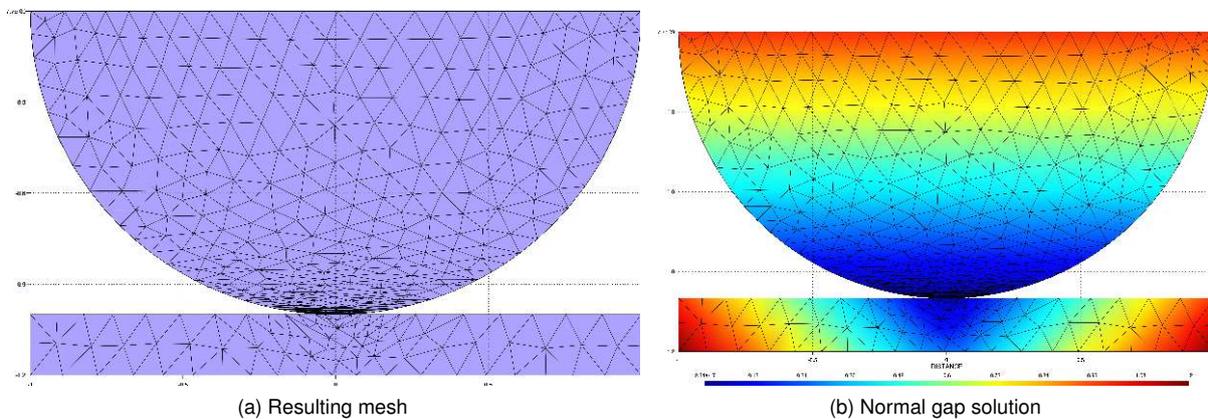


Figure 6.20: Solution obtained

Applying the level-set technique the resulting mesh, Figure 6.20, shows us a finer mesh around the contacting zone, but with an anisotropy that *a priori* does not benefit us. Therefore we can conclude that the **level-set approach does not provide a good mesh estimation** for contact problems.

6.8.2 Hessian metric

6.8.2.1 Theory

The next approach we want to consider in the CCM is the *Hessian* based remeshing procedure. This technique is described in detail in the previous section 6.3.Hessian based remeshing technique. We want to consider several variables which provide an adequate description of the contact problem so the mesh will be finer in the interest regions, especially the boundary region that deals with the contact problem.

The variables that can be taken into account can be many, one of them may be the displacement, but this choice does not provide a good information. In a generic CSD may be a good approach⁷, as the displacement may be a relevant variable which describes in a good manner the problem itself. But, for example, in a contact problem where the both domains are already in contact this variable will not be very helpful to describe our problem. A variable more associated to CCM is the contact pressure, but this one it is available only in the contacting interface, therefore not affecting the whole domain, and not providing a full information of the problem. Due to the intersection capabilities of the *metric-based* technique, see 6.3.2.2.Metric intersection, we can consider one or two additional variables which will provide a full information of the domain. We think in two different variables which are natural to consider on this problem, the VM equivalent stress and the *strain-energy*. They are natural in the sense that many CCM problems involve plasticity too, and in consequence are descriptive of the whole problem. In summary, the contact stress will provide us information about the contact boundary, and this information will be complemented with the VM equivalent stress and the *strain-energy*. The example shown in 6.8.2.2.2.Punch test explores the consideration or not consideration of the *strain-energy* in combination of the VM stress.

So we can consider the contact stress on both sides of the problem, not just in the *slave* domain, we must map the values from the *slave* domain to the *master* domain. In order to do so, and trying to be the most consistent possible with the formulation considered, we will apply the *Mortar* mapper developed in this work, see E.Mortar mapper. With this, the domain will be remeshed equally, or proportionally, in each one of the contact domains. Additionally, the VM equivalent stress must be extrapolated to the nodes in considering the method present in 6.7.Integration points values extrapolation. In order to normalise these values, so the metric obtained is not so restrictive, we will consider a normalisation factor which will depend on the *Young* modulus (E) and the *Poisson* ratio (ν). These values are taken into account in order to have a magnitude order to the stresses. The Equation (6.26) shows the normalisation factor (f_{norm}) considered in our formulations. This value has been obtained via numerical experimentation to get a proportion of factors providing a reasonable value.

$$(6.26) \quad f_{norm} = \frac{20}{\nu^2 E}$$

6.8.2.2 Numerical example

6.8.2.2.1 Simplest patch test :

The following is the simplest contact patch test possible, with only two blocks, and the displacements in the outer lines being fixed in the x axis. The geometry and the mesh of the problem considered can be seen in Figure 6.21, using triangular elements so we can remesh with *Mmg*. The continuous distribution of displacements in this problem can be seen in Figure 6.21b. A contact patch test is a problem where the distribution of the stresses is constant. The constant stresses can be observed on Figure 6.21c. The solution presented is interesting on this contest due to the computed *Hessian* will be zero, as there is no variation on the stresses, Figure 6.21d, and therefore in theory the obtained mesh with this method will be the original one, or the minimal sizes we impose on the resolution. Therefore this is just an illustrative case to show in which cases the method will preserve the original mesh.

⁷This is not the case for 6.10.2.Beam problem. *Hessian* of displacement, where the frequency corresponds with a highly oscillating beam.

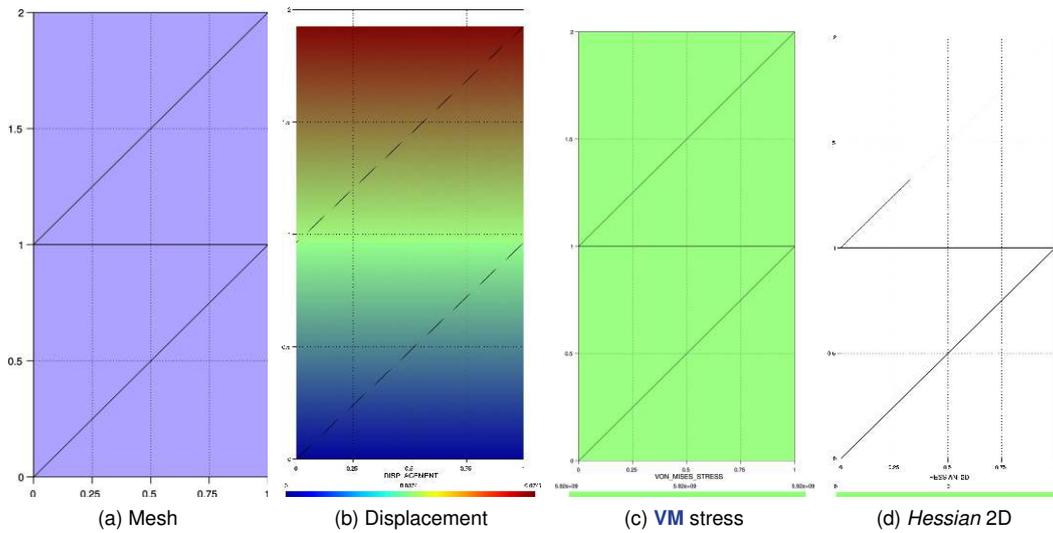


Figure 6.21: Simplest patch test

6.8.2.2.2 Punch test :

The numerical example will consist in a simple punch test in plane strain with a very coarse mesh. The problem is remeshed considering a *Hessian* metric. The metric intersects the contact pressure, **VM** stress and additionally the strain energy. We will evaluate 4 steps of $\Delta t = 0.5s$, with a vertical displacement in the top face of $u_y = -0.01t$. The remesh will be performed each 2 steps. The material considered is steel for both domains (Table 6.2).

Domain	E	ν
Upper domain	$2 \cdot 10^{11} Pa$	0.29
Lower domain	$2 \cdot 10^{11} Pa$	0.29

Table 6.2: Parameters considered for the *Hessian* remesh punch test

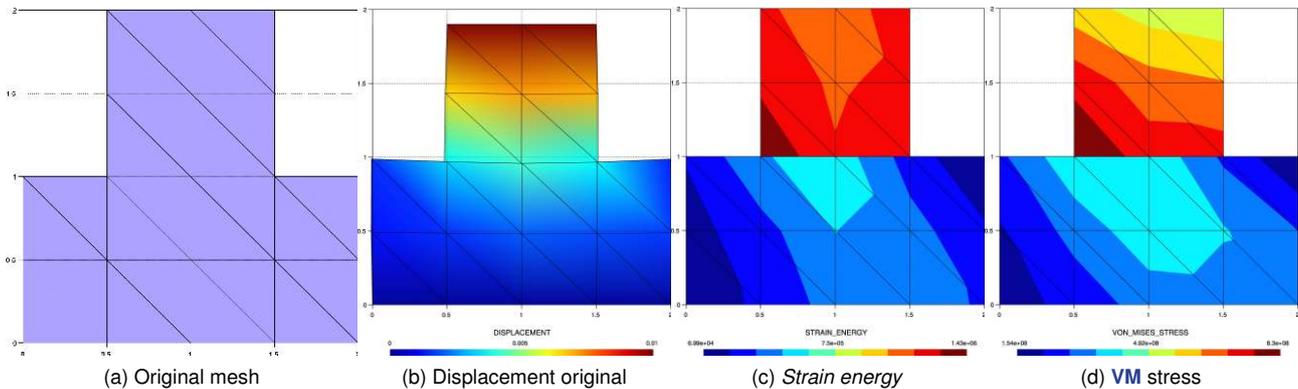


Figure 6.22: Punch test geometry considered. Solution $t = 0.5s$. Example [here](#)

We will compare the solution between taking into account the strain energy and without it. As the initial mesh is very coarse it does not provide much information, but in the later steps we can see as the mesh is refined around critical points. The original mesh, as well as the displacement, strain energy and **VM** stress can be found in 6.22.

The Figure 6.23 shows the solution at $t = 1.0s$ considering the *Hessian* metric for the contact stress and the **VM** stress. This solution can be compared with the solution provided with the metric that also considers the *strain energy* in the Figure 6.24. Both solutions are very similar, this is in part due to the fact that the first mesh is very coarse and the information which provides is limited. Anyhow, there are some small differences in the distributions obtained, and the ones which consider the *strain energy* provides a more detailed distribution in the solution achieved.

Solutions from Figures 6.25 and 6.26 show the results at $t = 2.0s$. In this case, as the previous mesh provided much more information than the original one, the final meshes are better defined.

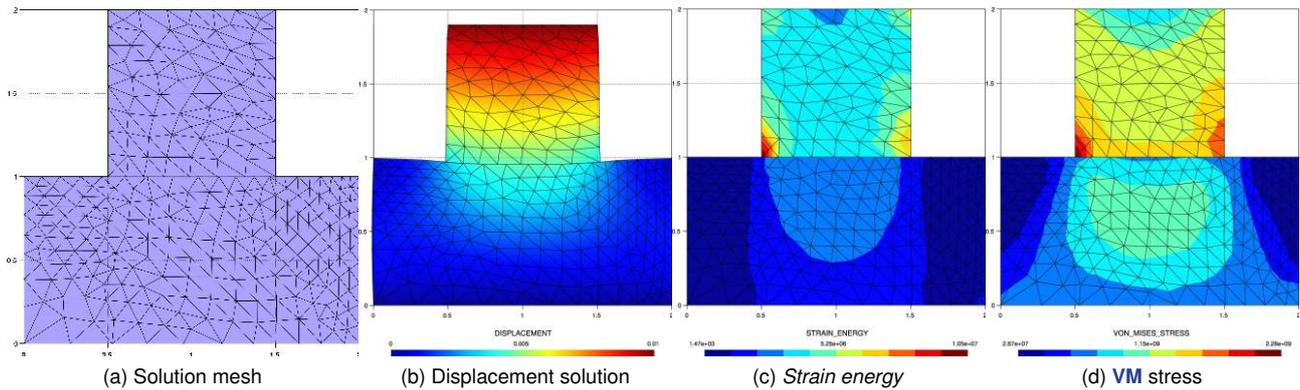


Figure 6.23: Punch test solution for **VM** stress and contact stress *Hessian* at $t = 1.0s$

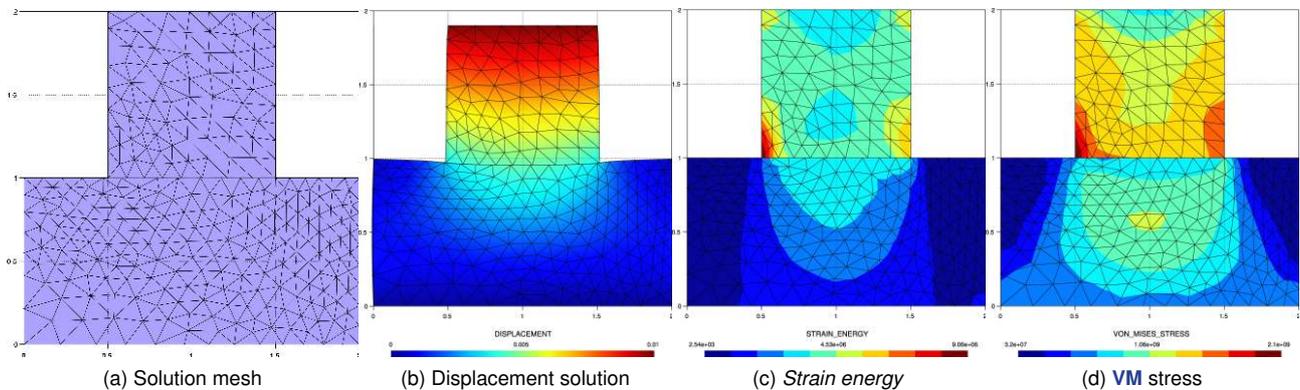


Figure 6.24: Punch test solution for *strain energy*, **VM** stress and contact stress *Hessian* at $t = 1.0s$

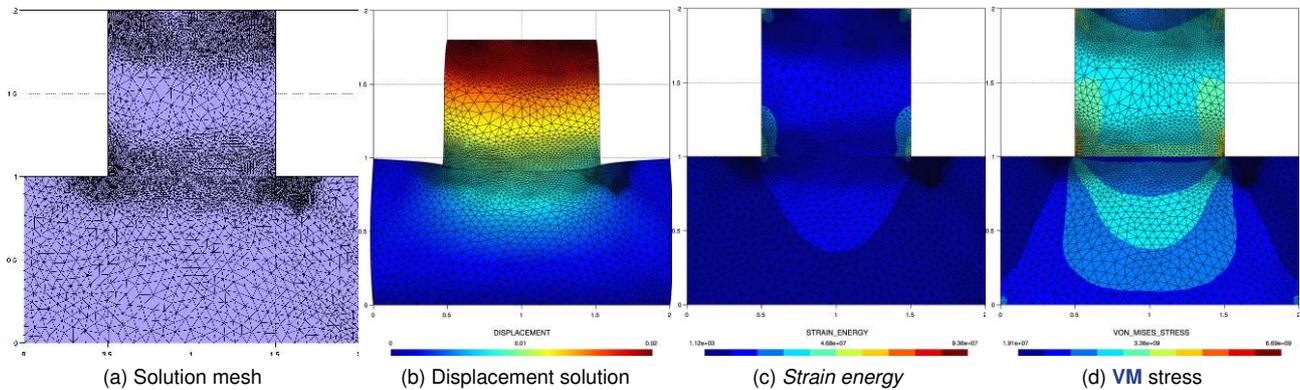


Figure 6.26: Punch test solution for *strain energy*, **VM** stress and contact stress *Hessian* at $t = 2.0s$

In both cases, Figures 6.25a and 6.26a show meshes which are finer in the contact boundary, as we must expect, as well as the points where the displacement is imposed, and therefore the reactions are concentrated in that region. Both meshes are quite similar, but the solutions obtained differ significantly, in the case of considering the *strain energy* too, the solution profile of the **VM** stress and the energy is more detailed and closer to the expected solution. In the other hand, the displacement solution, Figures 6.25b and 6.26b, converges quite fast in comparison with the other variables, as expected. This faster convergence was mentioned previously in the **SPR** section, in Figure 6.11 from 6.5.2.2. *Error estimation*.

Here, we remember that the intersection concepts have always provided the most restrictive of all the provides metrics, so the consideration of the additional variable always provides additional information without worsening the previous solution. Of course, this does not mean we can intersect all the possible variables, as our interest is to define

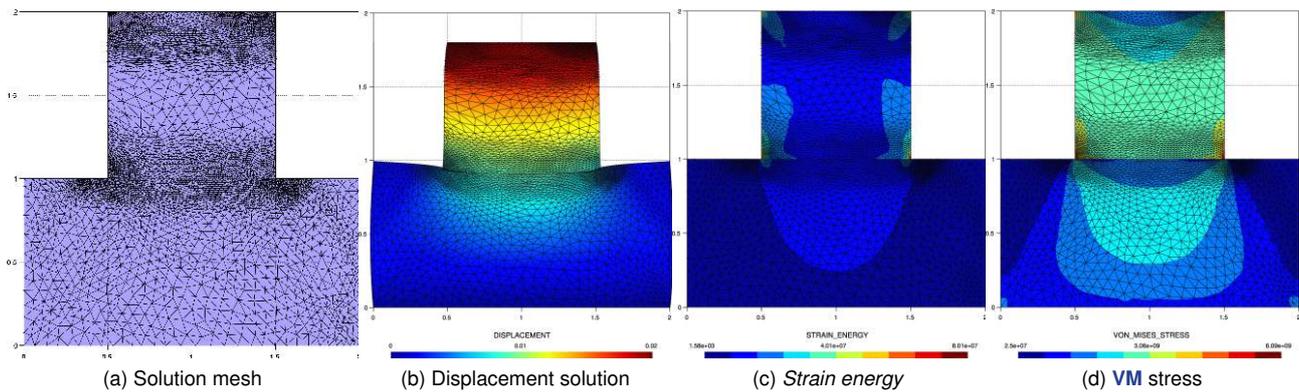


Figure 6.25: Punch test solution for **VM** stress and contact stress *Hessian* at $t = 2.0s$

a good mesh to solve our problem of interest, and so we should focus on those variables that are representative of the phenomenon of study, as we have shown in this example. We could for example, consider in addition the displacement, but as the displacement field is relatively homogeneous in this problem, and as the initial surfaces are already in contact, the information which is provided is not very relevant.

6.8.3 SPR metric

6.8.3.1 Theory

The adaptation of the method for **CCM** was first introduced by *Wriggers*[ArtWS98]. In order to do so, the stress recovery on contact boundaries needs some special consideration[BookRam01]. This approach states that each node on the slave contact boundary Γ_c^1 (standard patch) is associated with the closest node in the master contact boundary Γ_c^2 (extended patches), as seen in Figure 6.27. The continuity requirements inside the patch system are given by Equation (6.27). On these expressions $(\cdot)_e$ and $(\cdot)_s$ represent the external patch and the standard patch respectively. $(\cdot)^\Gamma$ represents that it is computed on the boundary.

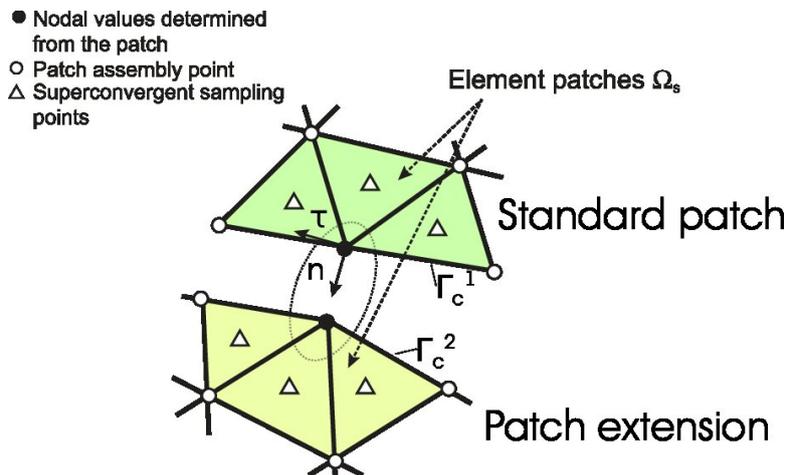


Figure 6.27: **SPR** contact interface example for linear triangles. Inspired[ArtWS98; BookZZT13]

The normal component:

$$(6.27a) \quad \mathbf{N} (\mathbf{P}_s^\Gamma \mathbf{a}_s - \mathbf{P}_e^\Gamma \mathbf{a}_e) = 0 \quad \text{on } \Gamma_c$$

The tangent component:

$$(6.27b) \quad \mathbf{T} (\mathbf{P}_s^\Gamma \mathbf{a}_s) = 0 \quad \text{and} \quad \mathbf{T} (\mathbf{P}_e^\Gamma \mathbf{a}_e) = 0 \quad \text{on } \Gamma_c$$

The normal and tangent vectors, are a the transposition in *Voigt* notation⁸ of the following (6.27c).

$$(6.27c) \quad \mathbf{N}_{expanded} = \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix} \text{ and } \mathbf{T}_{expanded} = \begin{bmatrix} \tau_x n_x & \tau_x n_y + n_x \tau_y & \tau_x n_z + n_x \tau_z \\ \tau_y n_x + n_y \tau_x & \tau_y n_y & \tau_y n_z + n_y \tau_z \\ \tau_z n_x + n_z \tau_x & \tau_z n_y + n_z \tau_y & \tau_z n_z \end{bmatrix}$$

As a new contribution adapting to our formulation *Wriggers*[ArtWS98] work, the contact constraint is enforced with the **ALM** method. Due to this the method **SPR** presented in *Wriggers*[ArtWS98] must be adapted in order to consider *Lagrange* multipliers instead of the penalty method. The main difference is that the contact pressure is *an unknown* of the system of equations instead of an estimated value from the current geometrical configuration. With this, the contact boundary conditions are enforced at each node l of the contact boundary for the normal direction (6.28), for the tangent direction is the same as in Equation (6.27b). In here $\bar{\lambda}_n$ is the augmented contact pressure obtained from the **ALM** formulation.

For the slave domain (standard patch):

$$(6.28a) \quad \mathbf{N}(\mathbf{x}_l) \mathbf{P}_s^\Gamma(\mathbf{x}_l) \mathbf{a}_s - \bar{\lambda}_n(\mathbf{x}_l) = 0 \quad \text{on } \Gamma_C^1$$

For the master domain (extended patch):

$$(6.28b) \quad \mathbf{N}(\mathbf{x}_l) \mathbf{P}_e^\Gamma(\mathbf{x}_l) \mathbf{a}_e - (\bar{\lambda}_n \circ \chi)(\mathbf{x}_l) = 0. \quad \text{on } \Gamma_C^2$$

On this expression χ represents the mapping operation. This is done considering the mortar mapper developed (see corresponding Appendix E.Mortar mapper) for a consistent computation of the normal gap (g_n) during the contact search (4.4.Contact detection. Search techniques).

We must now minimise the expressions from Equation (6.29). It should be noticed that the patches on the standard and extended side are not coupled. On these expressions, ε_n and ε_τ are the normal and tangent penalties, which do not necessarily coincide with the ones taken into consideration on the **ALM** formulation.

For the slave domain (standard patch):

$$(6.29a) \quad \min_{\mathbf{a}_i} \Pi = \min_{\mathbf{a}_i} \sum_{k=1}^n [\sigma_h(\mathbf{x}_k) - \mathbf{P}_s(\mathbf{x}_k) \mathbf{a}]^2 + \sum_{l=1}^m \left(\varepsilon_n [\mathbf{NP}_s^\Gamma(\mathbf{x}_l) \mathbf{a} - \bar{\lambda}_n(\mathbf{x}_l)]^2 + \varepsilon_\tau [\mathbf{TP}_s^\Gamma(\mathbf{x}_l) \mathbf{a}]^2 \right)$$

For the master domain (extended patch):

$$(6.29b) \quad \min_{\mathbf{a}_i} \Pi = \min_{\mathbf{a}_i} \sum_{k=1}^n [\sigma_h(\mathbf{x}_k) - \mathbf{P}_e(\mathbf{x}_k) \mathbf{a}]^2 + \sum_{l=1}^m \left(\varepsilon_n [\mathbf{NP}_e^\Gamma(\mathbf{x}_l) \mathbf{a} - (\bar{\lambda}_n \circ \chi)(\mathbf{x}_l)]^2 + \varepsilon_\tau [\mathbf{TP}_e^\Gamma(\mathbf{x}_l) \mathbf{a}]^2 \right)$$

The former, leads to the following equation system on Equation (6.30).

For the slave domain (standard patch):

$$(6.30a) \quad \begin{aligned} \text{LHS}^{slave} \mathbf{a} &= \text{RHS}^{slave} \\ \text{LHS}^{slave} &= \sum_{k=1}^{n_{slave}} \mathbf{P}_s^T(\mathbf{x}_k) \mathbf{P}_s(\mathbf{x}_k) + \sum_{l=1}^{m_{slave}} \left(\varepsilon_n \mathbf{P}_s^{\Gamma T}(\mathbf{x}_l) \mathbf{N}^T \mathbf{NP}_s^\Gamma(\mathbf{x}_l) + \varepsilon_\tau \mathbf{P}_s^{\Gamma T}(\mathbf{x}_l) \mathbf{T}^T \mathbf{TP}_s^\Gamma(\mathbf{x}_l) \right) \\ \text{RHS}^{slave} &= \sum_{k=1}^{n_{slave}} \mathbf{P}_s(\mathbf{x}_k) \sigma_h(\mathbf{x}_k) + \sum_{l=1}^{m_{slave}} [\varepsilon_n \bar{\lambda}_n(\mathbf{x}_l)] \end{aligned}$$

For the master domain (extended patch):

$$(6.30b) \quad \begin{aligned} \text{LHS}^{master} \mathbf{a} &= \text{RHS}^{master} \\ \text{LHS}^{master} &= \sum_{k=1}^{n_{master}} \mathbf{P}_e^T(\mathbf{x}_k) \mathbf{P}_e(\mathbf{x}_k) + \sum_{l=1}^{m_{master}} \left(\varepsilon_n \mathbf{P}_e^{\Gamma T}(\mathbf{x}_l) \mathbf{N}^T \mathbf{NP}_e^\Gamma(\mathbf{x}_l) + \varepsilon_\tau \mathbf{P}_e^{\Gamma T}(\mathbf{x}_l) \mathbf{T}^T \mathbf{TP}_e^\Gamma(\mathbf{x}_l) \right) \\ \text{RHS}^{master} &= \sum_{k=1}^{n_{master}} \mathbf{P}_e(\mathbf{x}_k) \sigma_h(\mathbf{x}_k) + \sum_{l=1}^{m_{master}} [\varepsilon_n (\bar{\lambda}_n \circ \chi)(\mathbf{x}_l)] \end{aligned}$$

⁸As the *Voigt* notation is not standard we present the whole matrix form.

6.8.3.2 Numerical examples

6.8.3.2.1 Simplest patch test :

We will consider the same example from 6.8.2.2.1. Simplest patch test in order to show that with the proposed method, where the resulting mesh corresponds with the original mesh. Following the same reasoning from that previous case, we can calculate the corresponding error in the integration points. We expect that it will be zero, Figure 6.28, as the stress is equal in all the points of the problem.

In conclusion, the mesh that will be obtained with this procedure will coincide again with the original one. One more time, the patch test allows us to show the relationship between stress distribution and the resulting metrics.

6.8.3.2.2 Punch test :

The following is a very simple punch test, the geometry is exactly the same shown in the Hessian example (6.8.2.2.2. Punch test). The problem consists in one 2×1 lower block and one 1×1 upper block, with a very coarse mesh, as seen in Figure 6.29a. We impose a small vertical displacement of 0.1 m in the top of the upper block. The Figure 6.29 shows the evolution of the mesh in each NL iteration, where the third and fourth are almost identical, when the error has converged. The second mesh (Figure 6.29b) is a very bad estimation, very fine a uniformly distributed, as the error from the first coarser mesh (Figure 6.31a) does not provide much information. In the following iterations (Figure 6.29c and 6.29d) we can appreciate as the mesh is finer across the points with higher demand, as expected.

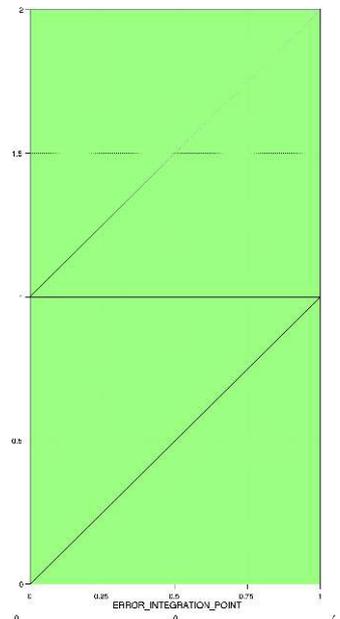


Figure 6.28: Error obtained

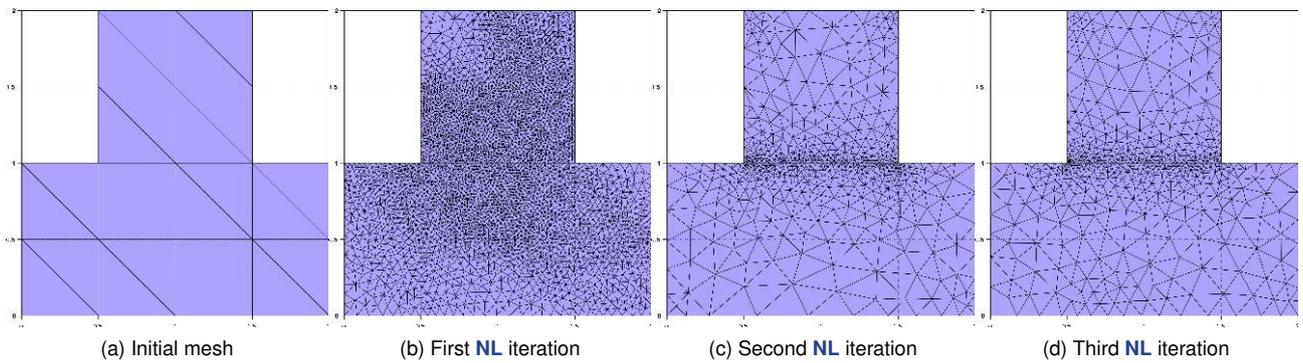


Figure 6.29: Mesh evolution

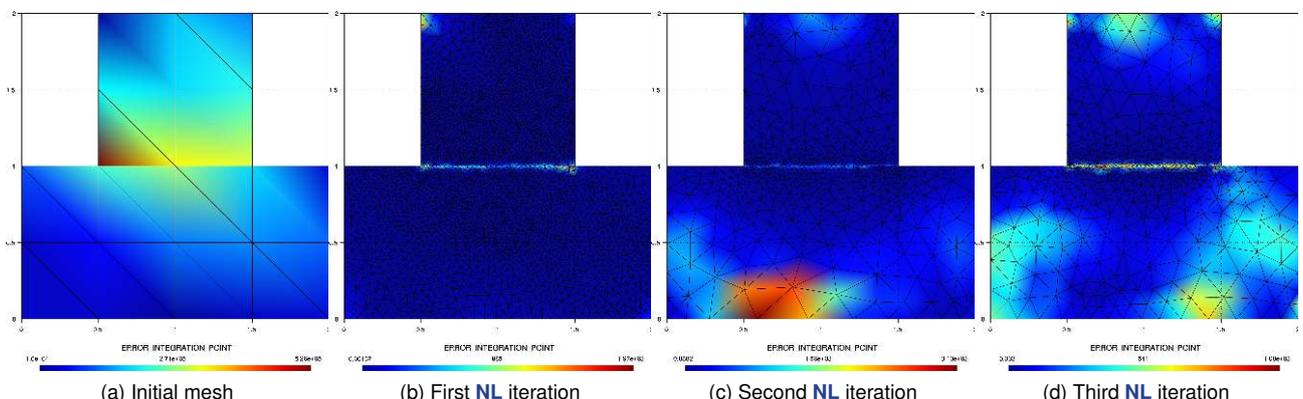


Figure 6.30: Error solution

Additionally the error, Figure 6.30, on these NL iterations is significantly smaller than in the previous ones. But more importantly, the mesh is more homogeneous, which is one of the objectives of the methodology to distribute the

error equally between the elements of the mesh.

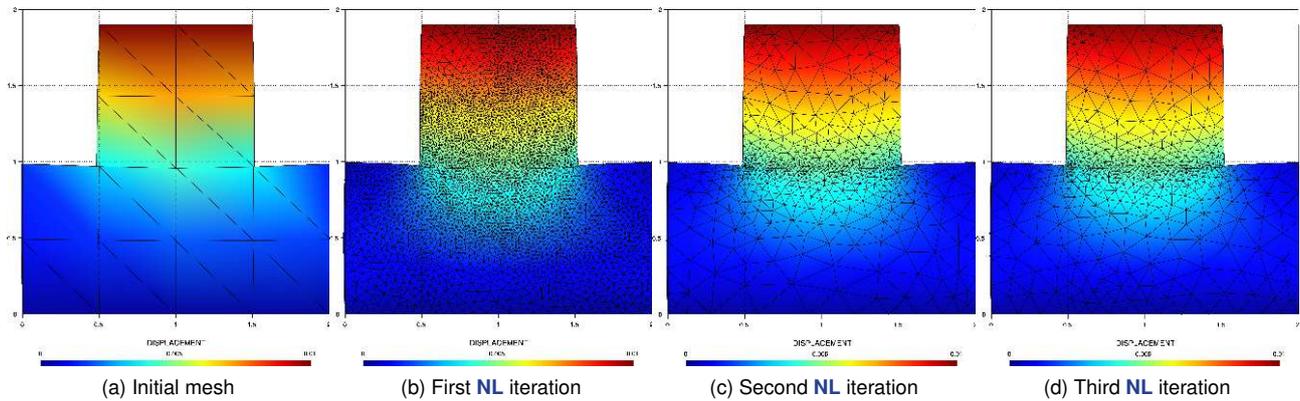


Figure 6.31: Displacement solution

Finally, as in the *Hessian* case from 6.8.2.2.2.Punch test, we see as the displacement converges faster than the previous values (Figure 6.31). As we have already stated, this is expected, as seen in 6.5.2.2.Error estimation.

6.9 Remeshing workflow

The following summarises the simulation workflow followed during the contact simulations. We will differentiate between the standard remeshing process considered during the *Level set* and the *Hessian* metrics, and the workflow followed for the *SPR* metric computation, which are slightly different.

6.9.1 Level set and Hessian remeshing

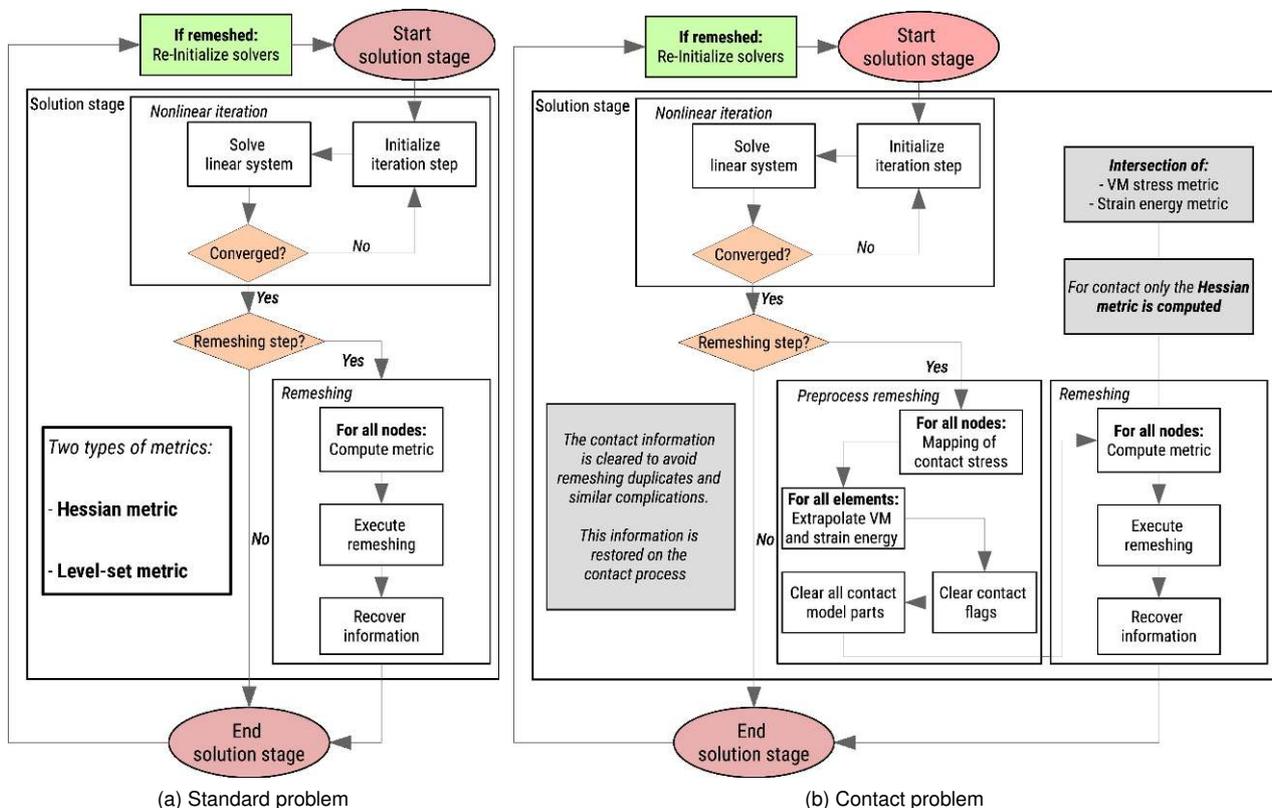


Figure 6.32: Workflow for *Level set/Hessian* metric remesh

We can differentiate between the standard problem and the contact problem.

6.9.1.1 Standard problem

For the standard case, we will compute the metric once the **NL** loop is converged, and the metric and the remesh will be computed independently of the previous solution, this means that we will not check any value is below a certain threshold. After remesh the information, the nodal and integration solutions, must be recovered. Once the solution stage is computed we will need to re-initialise the solvers and processes if the problem has been remeshed in order to restore the problem before execute the next time step.

6.9.1.2 Contact problem

The main difference between the standard problem and the contact one is that to avoid problematic and duplicated conditions the associated conditions are removed and cleared, this will be recovered after re-initialise the processes. In addition to that we will remap the contact stresses between domains and extrapolate the **VM** and the strain energy. After that some additional information must be cleared, as the contact flags and the contact model parts.

6.9.2 SPR remeshing

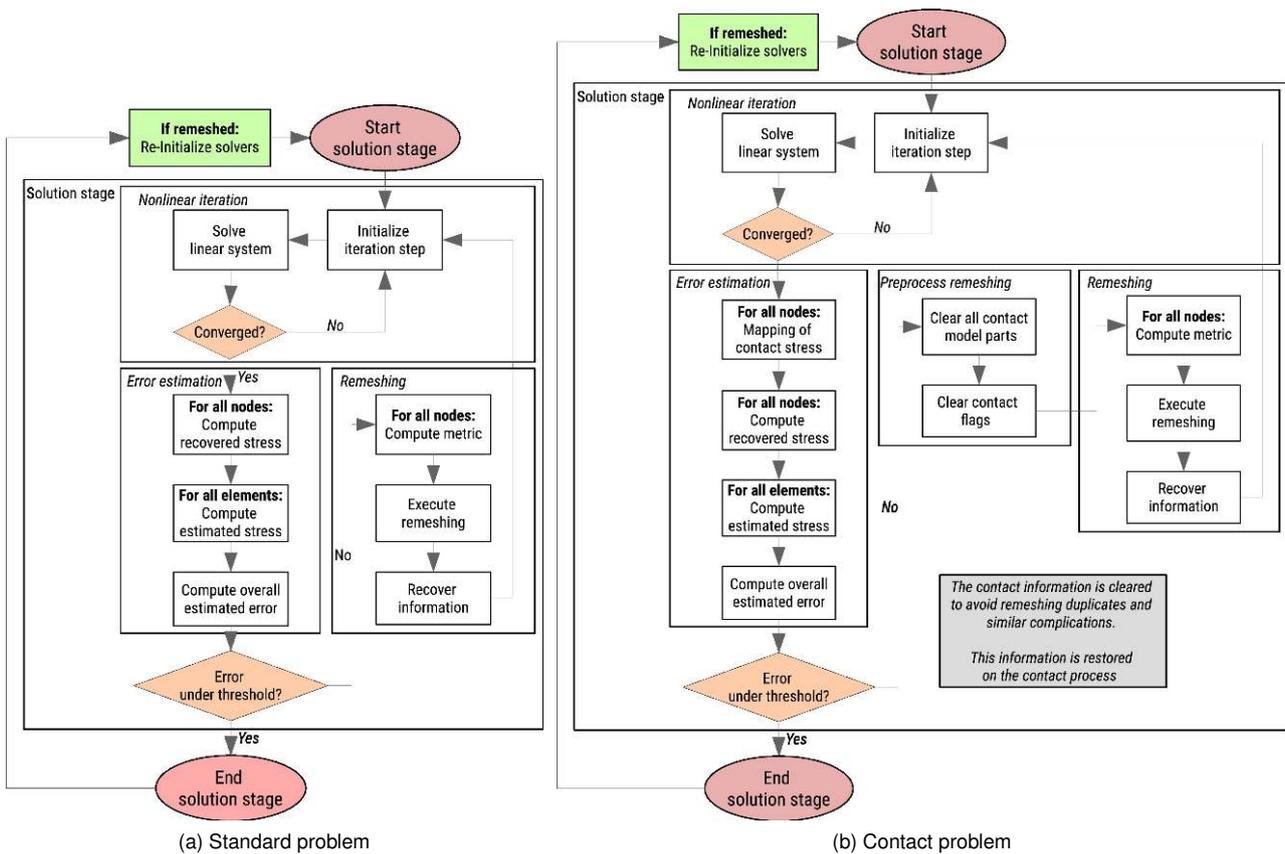


Figure 6.33: Workflow for **SPR** metric remesh

For the error estimation case the procedure is slightly different, not only because the error must be computed, but also because if the error is below a certain threshold the remeshing is not executed.

6.9.2.1 Standard problem

The main difference respect 6.9.1.1. Standard problem is that once the **NL** loop has converged, we will need to compute the recovered stress in the nodes, to later compute the estimated stress on the nodes. With this we will

compute the estimated error, if the error is below a certain threshold the time solution stage will be finished, if not with the previous values we will estimate the size of the elements, and with it, the corresponding metrics.

6.9.2.2 Contact problem

Additionally to the previous consideration for the **SPR** error metric, we will consider the modifications from 6.9.1.2. **Contact problem**, except that the extrapolation of integration values is not required and the mapping of the contact stresses is performed before computing the error on the mesh.

6.10 Numerical examples

This section will show more advanced cases than the previously presented. The ones formerly introduced acted as proof of concept to the concepts that were introduced in the respective sections. In this section we will focus in structural cases, but we will dedicate an additional section to **CFD** cases, 6.11. **CFD numerical examples**, in order to show up that this adaptive remeshing techniques are generic and can be considered in problems of different physical nature.

6.10.1 Coarse sphere. Level set

In this problem we remesh using the gradient of the distance function, which is the distance to the plane contained in the sphere centre. The function can be seen in the Figure 6.34, which corresponds with a linear distance and a constant gradient.

The solution obtained can be seen in Figure 6.35, where despite the original coarse configuration, the final one is smoothed due to the inner *Bezier* cubic surface considered by the *Mmg* library (see B.3.1. **What is Mmg and how does it work?**).

It can be seen like the elements in the centre, where the distance function is zero, are the smaller ones, and the ones more anisotropic.

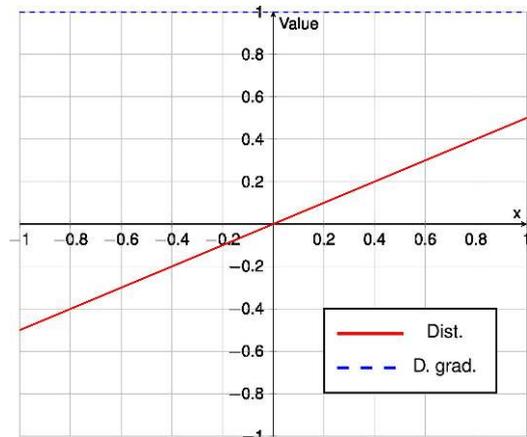
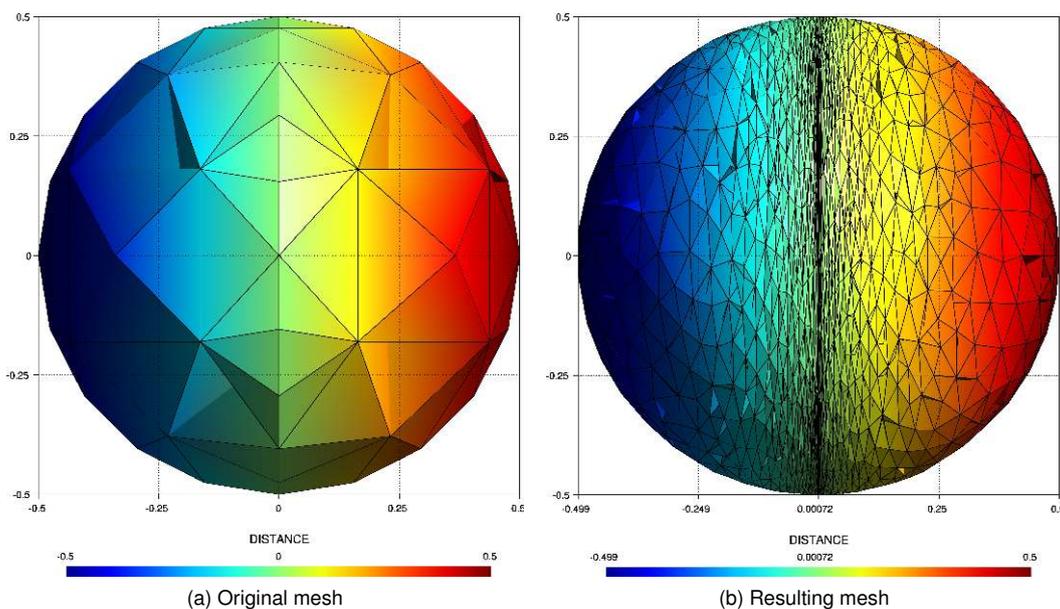


Figure 6.34: Distance function



(a) Original mesh (b) Resulting mesh
Figure 6.35: Mesh before and after remesh for coarse mesh. Example

6.10.2 Beam problem. *Hessian* of displacement

The problem corresponds with a steel cantilever 10×1 and considering as only load self-weight, see Figure 6.36. The initial mesh considered is the coarse mesh from Figure 6.36. We consider a simulation of 1s with $\Delta t = 0.01s$. The whole example can be found in [here](#).

The final configuration, displacement and mesh can be seen in Figure 6.37. Due to the high natural frequency of the problem, the displacement is not a good measure for the *Hessian* metric, as the high number of oscillations induce the resulting mesh in Figure 6.37. Where the distribution is not homogeneous, and does not follow any particular pattern.

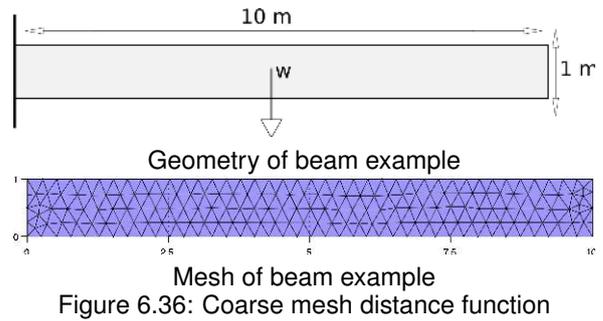


Figure 6.36: Coarse mesh distance function

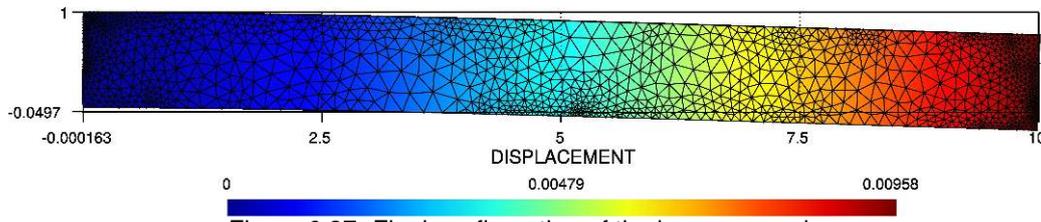


Figure 6.37: Final configuration of the beam example

6.10.3 Hertz problem. *Hessian* of contact and VM stress

The problem consists in a *Hertz* contact problem remeshing considering the *Hessian* of the **VM** stress and the contact pressure. We will consider a significant top pressure in order to maximise the contact area, which will help us to evaluate the quality of the solution achieved, reason why the solution may look so deformed. We will not take into consideration the *strain energy*, as the effect of this variable have been already taken into account in the punch test example from 6.8.2.2.2. **Punch test**. The materials of this problem can be found in Table 6.3, where the base can be considered rigid respect to the die.

Domain	E	ν
Upper domain	$2 \cdot 10^8 Pa$	0.35
Lower domain	$2 \cdot 10^{11} Pa$	0.29

Table 6.3: Parameters considered for the *Hessian* remesh *Hertz* example

There are 10 steps $\Delta t = 0.1s$, and the remesh is executed each 3 steps starting in the 4th. The load applied depends on time, and it is equal to $q = 10^7 t Pa$. The example can be located in [Kratos repository](#).

The first step, Figure 6.38, is a very coarse mesh, but dense enough to provide the information needed in order to estimate properly the required mesh. Looking at the **VM** stress from Figure 6.38c we can predict where the mesh will become denser, this corresponds mostly with the contact interface. Particularly taking into account that we consider the contact pressure in addition to the **VM** stress, this area will be especially enhanced. Anyhow, despite not being terrible, the mesh provides an improved information about the **VM** equivalent stress.

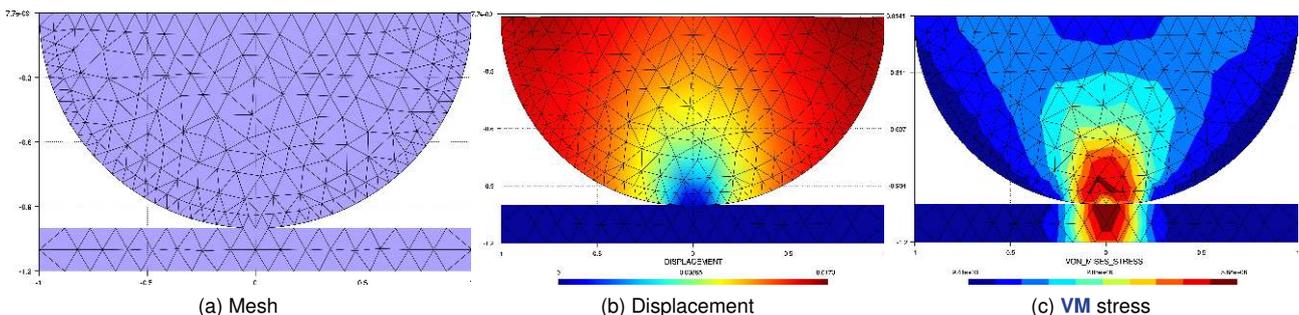


Figure 6.38: *Hertz* solution step 1

The solution in the second mesh, Figure 6.39, refines the mesh in the required regions predicted previously. In addition to the contact interface, the loaded face is also enhanced, this is expected as is the region of the problem where the loads are concentrated.

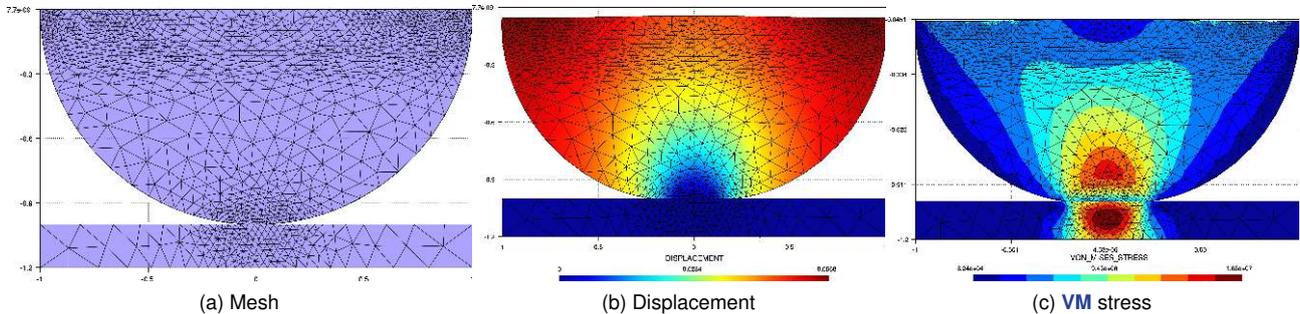


Figure 6.39: *Hertz* solution step 4

The third mesh, Figure 6.40, is more defined than the previous one, with less anisotropic elements near the load face. Additionally, as the contact zone is wider than in the previous case, the refined contact zone is extended.

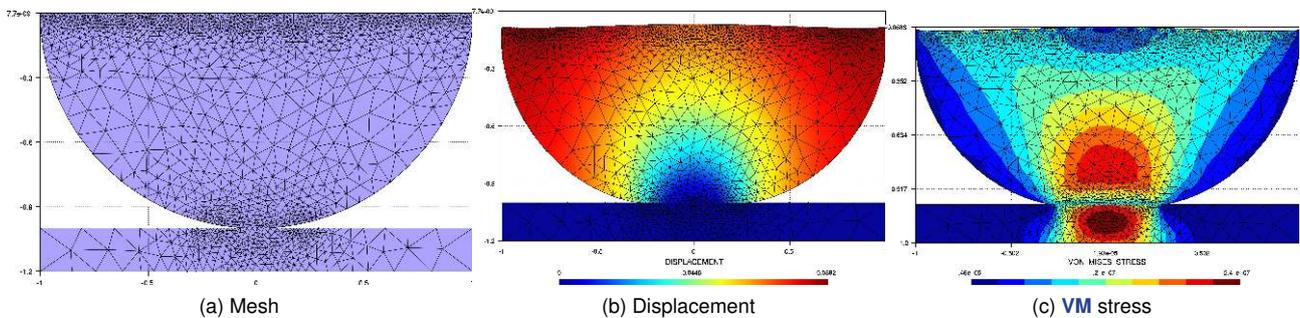


Figure 6.40: *Hertz* solution step 7

Finally, we can notice that the final mesh it almost identical to the previous one, Figure 6.41. This is an indicative that the current mesh already fits our requirements. In these two last meshes the **VM** stress is well represented after the refinement.

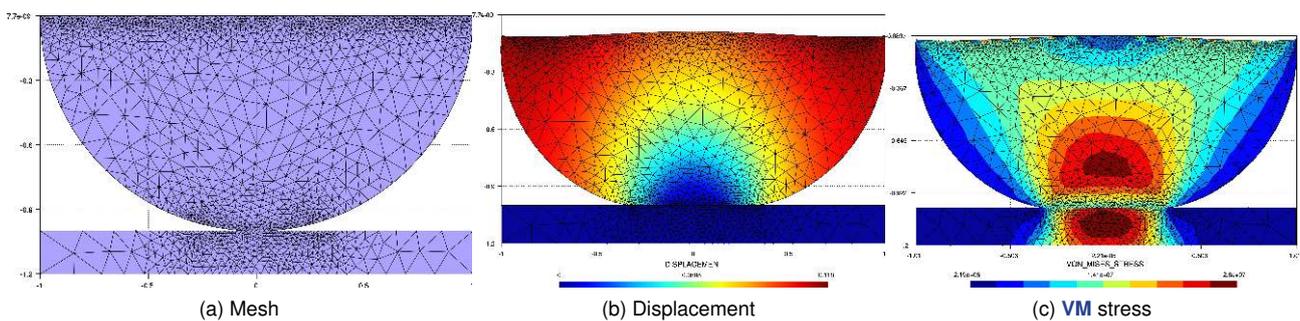


Figure 6.41: *Hertz* solution step 10

6.10.4 *Hertz* problem. SPR error computing

Now we take the same problem from 6.10.3. *Hertz* problem. *Hessian* of contact and **VM** stress, but we will consider the **SPR** metric instead of the *Hessian*. This will allow us to compare the solution obtained with both methods in addition to the previous punch examples. On the first step, we can observe as the error is mainly concentrated on the contact boundary, Figure 6.42c, this is what is expected like in the former *Hessian* case.

With the former error estimation, the obtained mesh is the one from Figure 6.43a. The elements are relatively small in the contact zone, as the error was concentrated due to the initial coarse mesh. In this new step, the error is

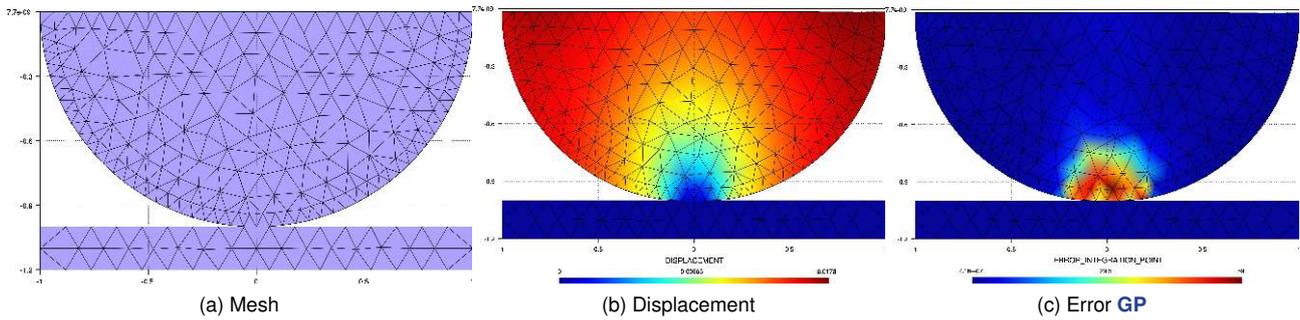


Figure 6.42: *Hertz* solution step 1

quite uniform on the sphere, Figure 6.43c. Therefore, we can expect that the elements will become more uniform, and become bigger, as the current mesh provides more information than the initial one.

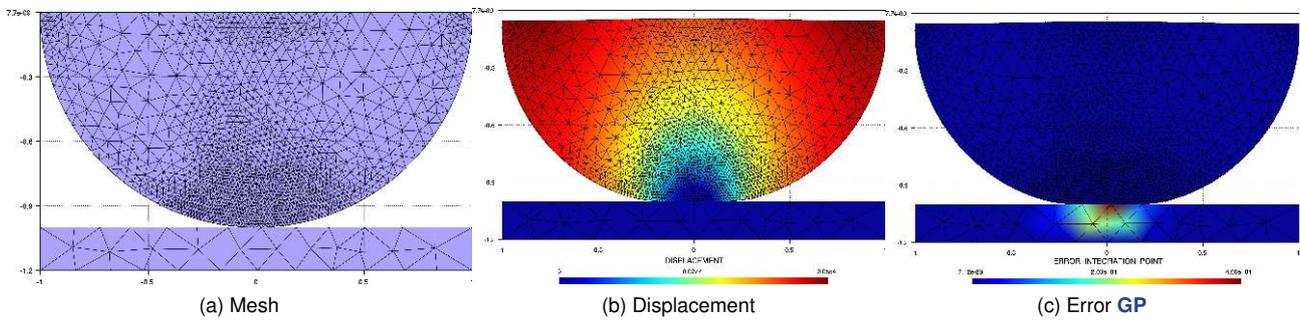


Figure 6.43: *Hertz* solution step 4

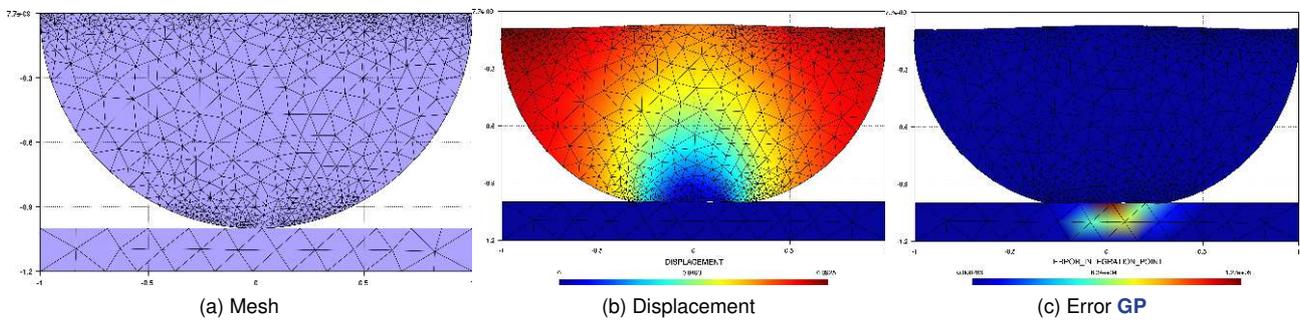


Figure 6.44: *Hertz* solution step 7

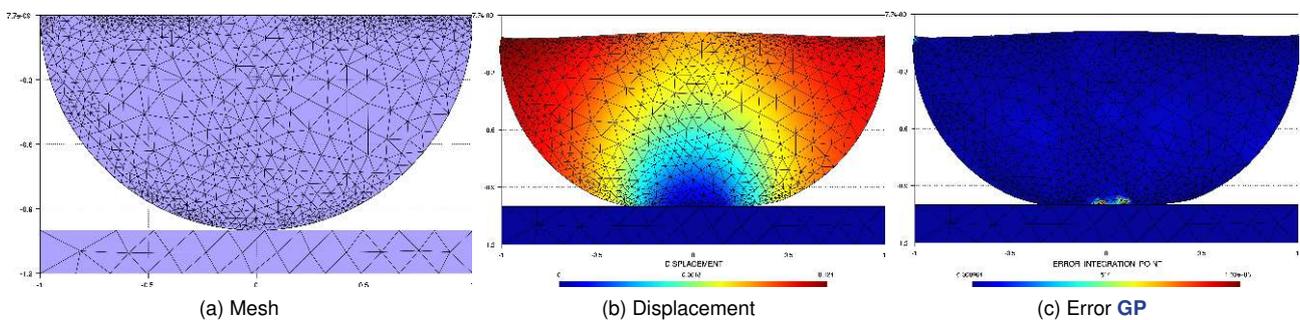


Figure 6.45: *Hertz* solution step 10

For the third step, the mesh becomes more uniform, Figure 6.44a, as previously mentioned. In any case it preserves the smaller elements in the contact interface and in the loaded face as it is supposed to be.

The last mesh, Figure 6.45, is quite similar to the previous step, as it is similar to the resulting mesh obtained with the *Hessian* metric, Figure 6.41. So, even if the approaches are quite different both methods provide reasonable results a relatively similar outcome.

One important difference with the *Hessian* solution is that the rigid base preserves the same mesh for all steps. This makes sense, as the stress distribution is constant for a rigid media. This does not happen in the *Hessian* case as we map the contact stress generating a variation on the contact boundary.

6.10.5 Contacting cylinders with adaptive remeshing

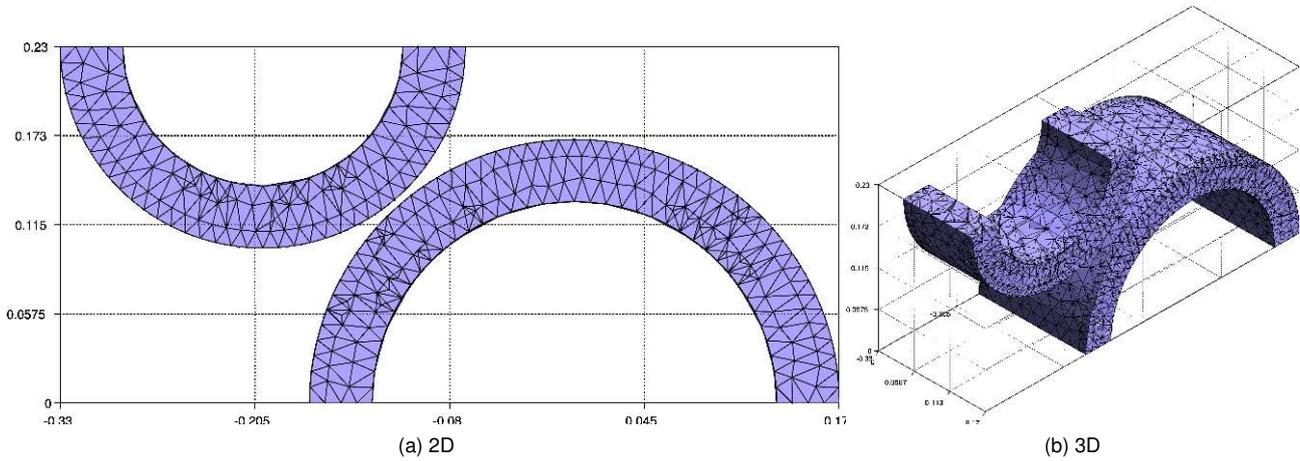


Figure 6.46: Initial mesh in contacting cylinders

The problem consists in a two cylinders with a very coarse mesh, Figure 6.46, where we impose a horizontal movement in the upper cylinder. This case is already presented in the contact chapter, 4.5.10. [Contacting cylinders](#), but preserving the same mesh. The cylinders are formulated as solids in **TL** framework with hyperelastic behaviour, with the properties from Table 6.4. We impose a horizontal movement $x = 0.2t$ at the upper cylinder, so the two cylinders contact between them. The problem has been remeshed each $2.5e - 2s$, and the total simulation time is 1.5s. In order to define the metric, the *Hessian* of the contact stress and the **VM** stress has been considered. This example is located at the [Kratos repository](#).

Domain	E	ν
Upper domain	$2 \cdot 10^{10} Pa$	0.35
Lower domain	$2 \cdot 10^{10} Pa$	0.35

Table 6.4: Parameters considered for the *Hessian* remesh contacting cylinders example

We present both the results from a 2D slice and a 3D perspective. The sliced section allows to better appreciate the element size evolution, as well as the value distribution inside the geometry. We complement with the 3D representation which allows us to picture the whole problem and the spatial variation in the values. In the following we represent the solution each $\Delta t = 0.35s$, so we have 4 steps fully represented. The values presented are the displacement of the configuration, the element size and the **VM** stress.

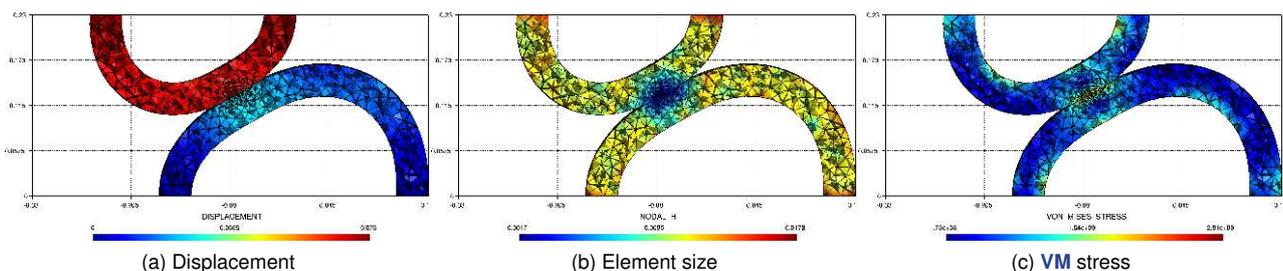


Figure 6.47: Contacting cylinders solution at $t = 0.35s$

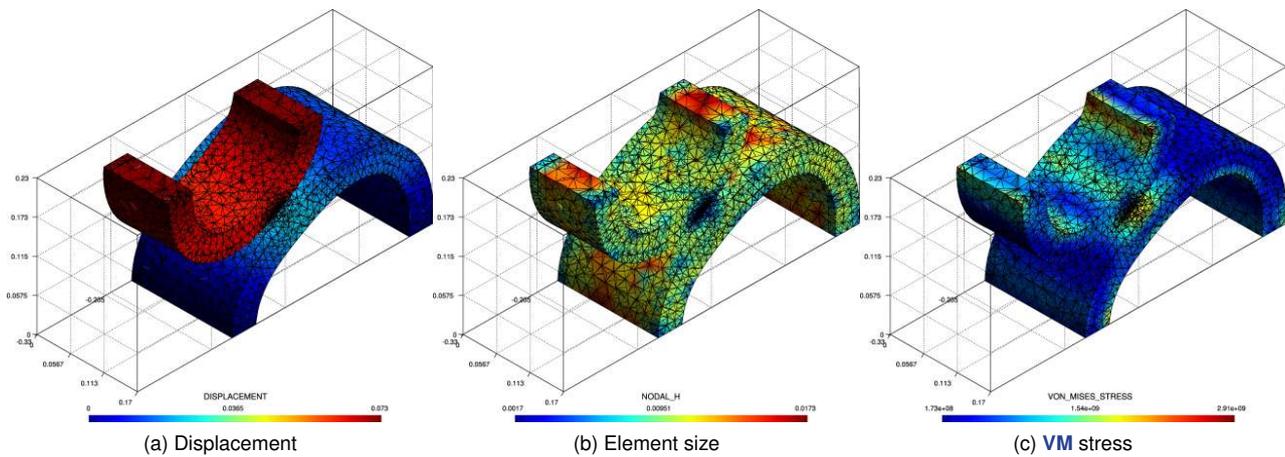


Figure 6.48: Contacting cylinders perspective solution at $t = 0.35s$

The solution obtained at $t = 0.35s$, Figures 6.47 and 6.48, shows like the region with the smaller elements is the expected one, the contact region, where the values of the contact pressure and the VM stresses are concentrated. This is even more notorious at $t = 0.7s$, Figures 6.49 and 6.50, where the contact zone is expanded and therefore there are more elements reduced in the remeshing process. The last steps, from Figures 6.51, 6.52, 6.53 and 6.54, are basically the opposite deformation from the previous ones, and the conclusions from these steps can be extended here.

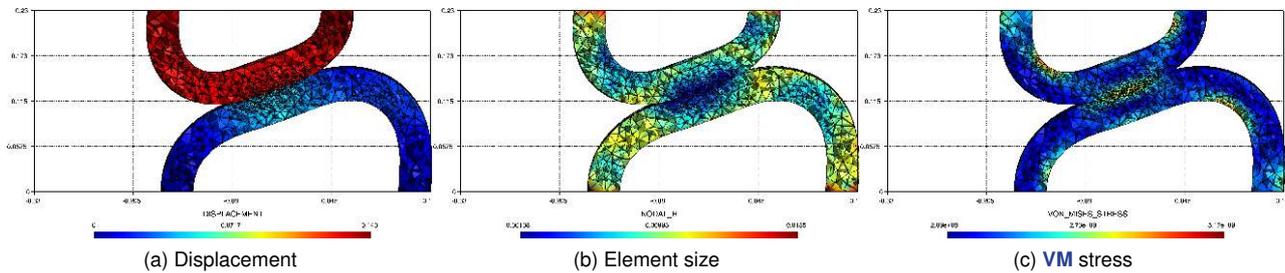


Figure 6.49: Contacting cylinders solution at $t = 0.70s$

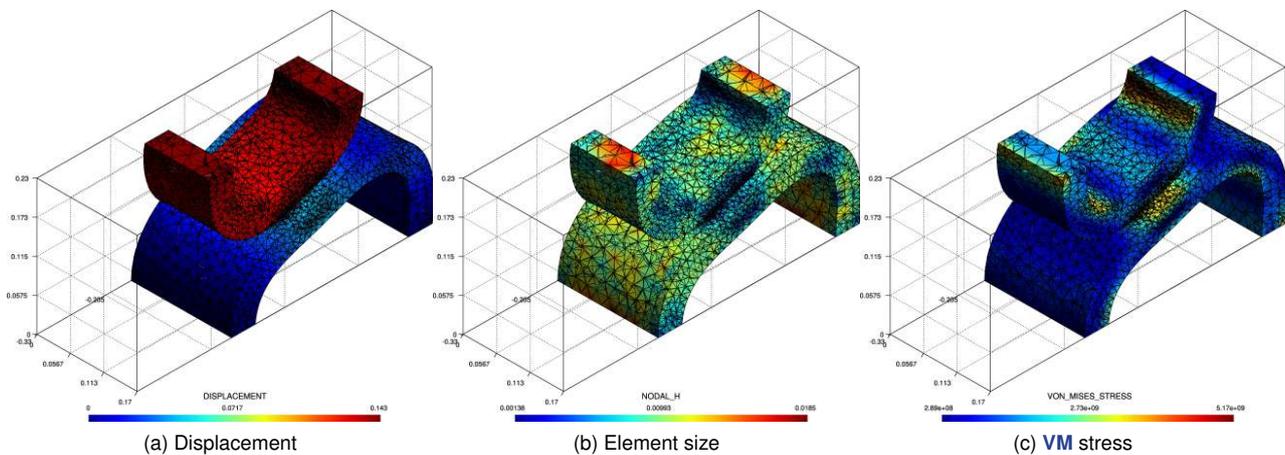


Figure 6.50: Contacting cylinders perspective solution at $t = 0.70s$

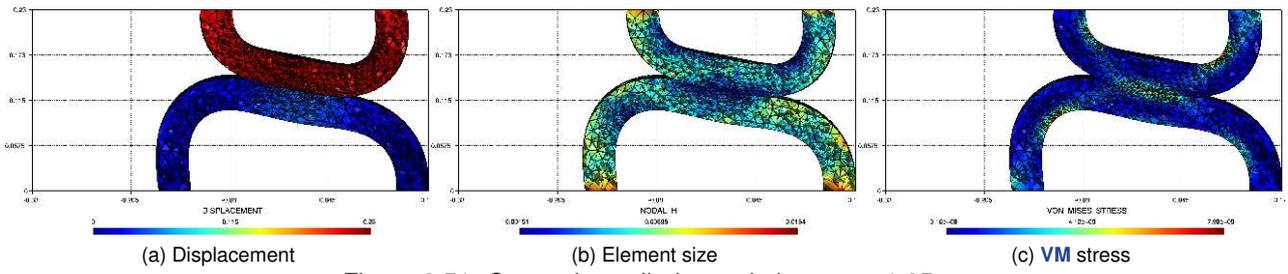


Figure 6.51: Contacting cylinders solution at $t = 1.05s$

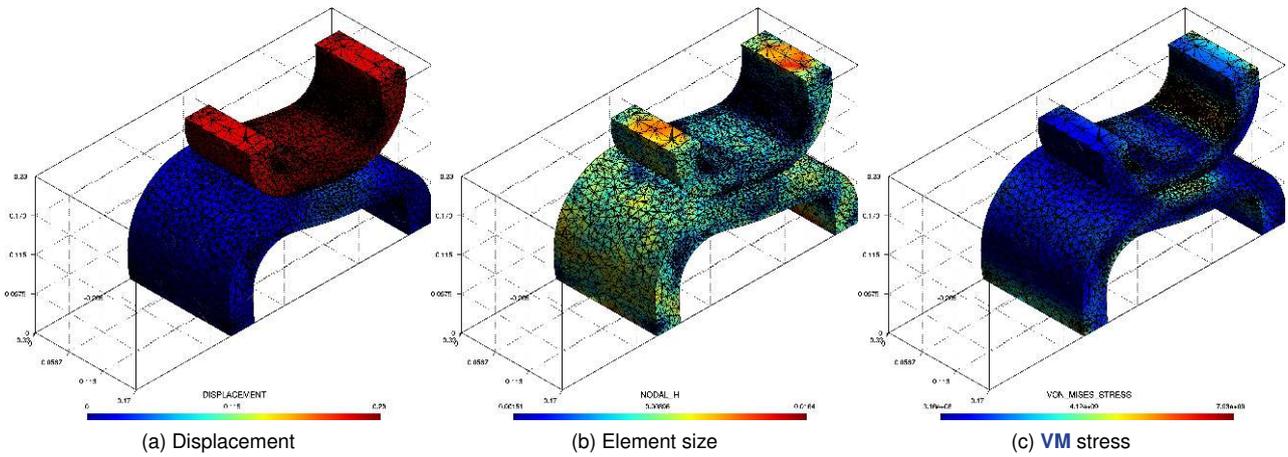


Figure 6.52: Contacting cylinders perspective solution at $t = 1.05s$

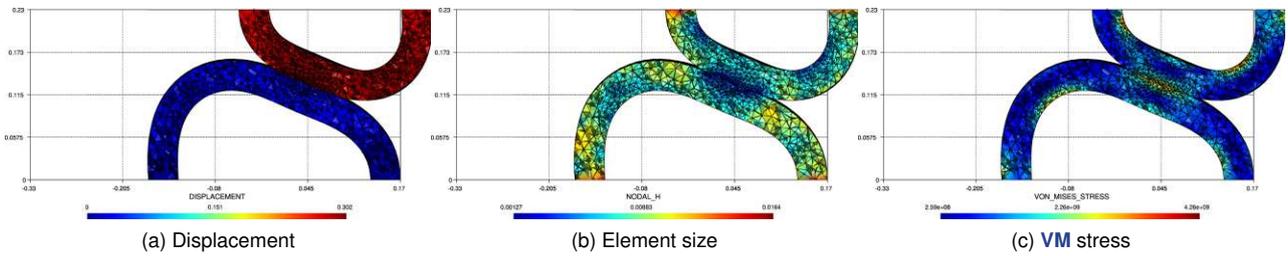


Figure 6.53: Contacting cylinders solution at $t = 1.4s$

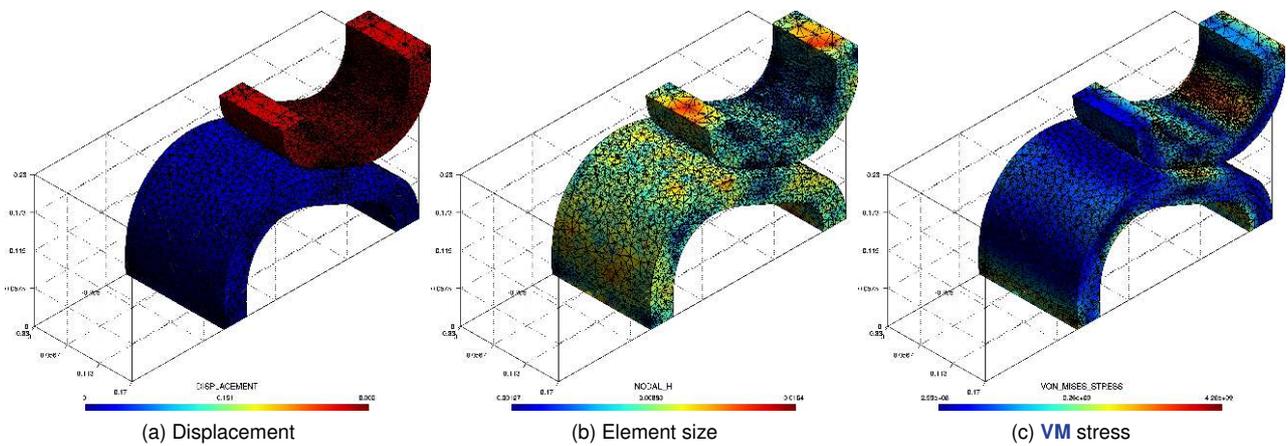


Figure 6.54: Contacting cylinders perspective solution at $t = 1.4s$

6.11 CFD numerical examples

6.11.1 Cavity. *Level set*

This is a very simple remeshing example, where we remesh a cavity 1×1 uses the distance to the centre as destination function. In Figure 6.55 it can be seen as the original mesh is homogeneous (Figure 6.55a) and the size of the mesh is equal for all elements (Figure 6.55b). The resulting mesh has the smaller elements around the centre of the square (Figure 6.55c) and the size is distributed depending on the closeness to the centre (Figure 6.55d). The example can be found [here](#).

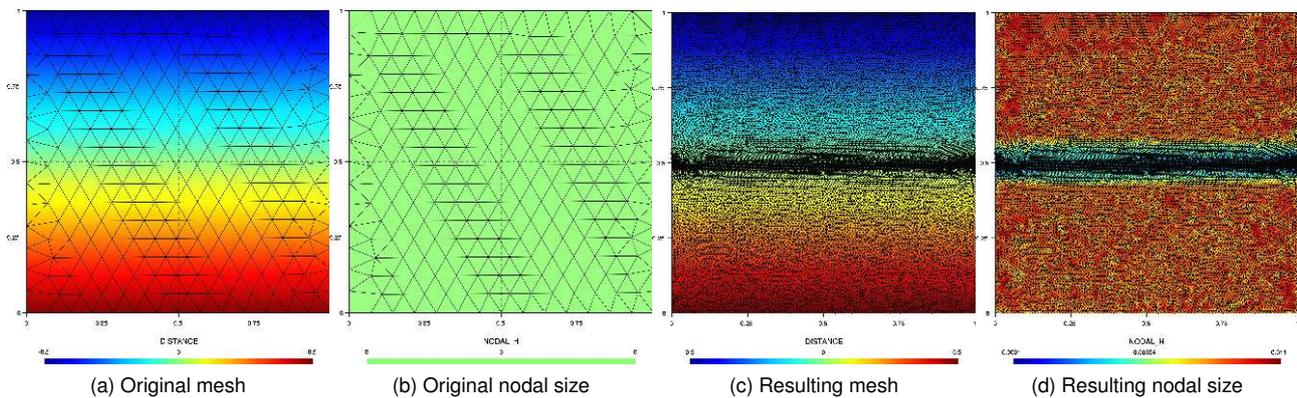


Figure 6.55: Mesh before and after remesh cavity case

6.11.2 Embedded cylinder. *Level set*

The problem consists in an anisotropic remeshing of a 2D fluid channel with a cylinder, using as level-set the distance function. It consists in a channel 5×1 , a cylinder of 0.3m diameter, Figure 6.56.

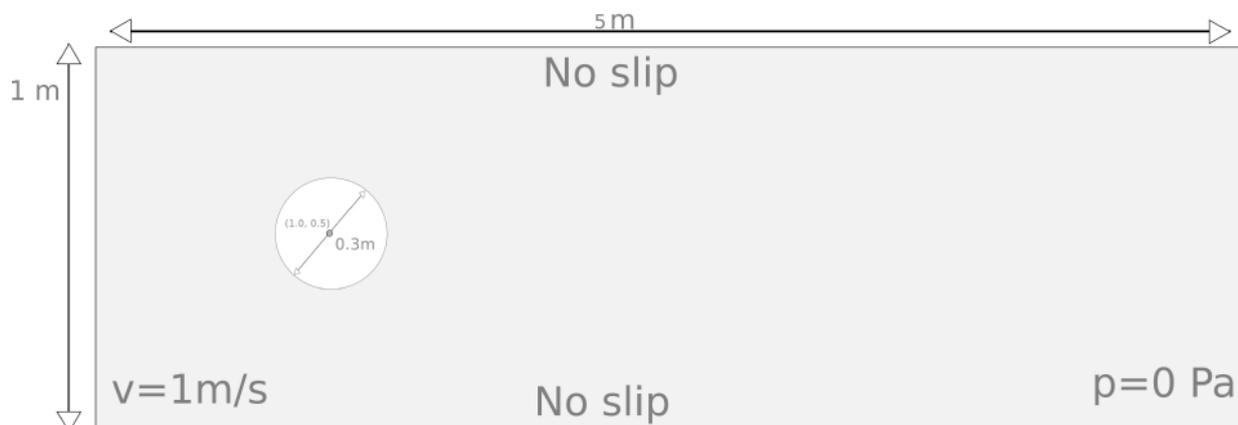


Figure 6.56: Embedded cylinder geometry

We start with a very refined mesh of 40000 nodes and it is simplified to a mesh of just 2500 nodes, centred around the cylinder of interest, Figure 6.57. The example can be found in the [Kratos Examples](#) repository.

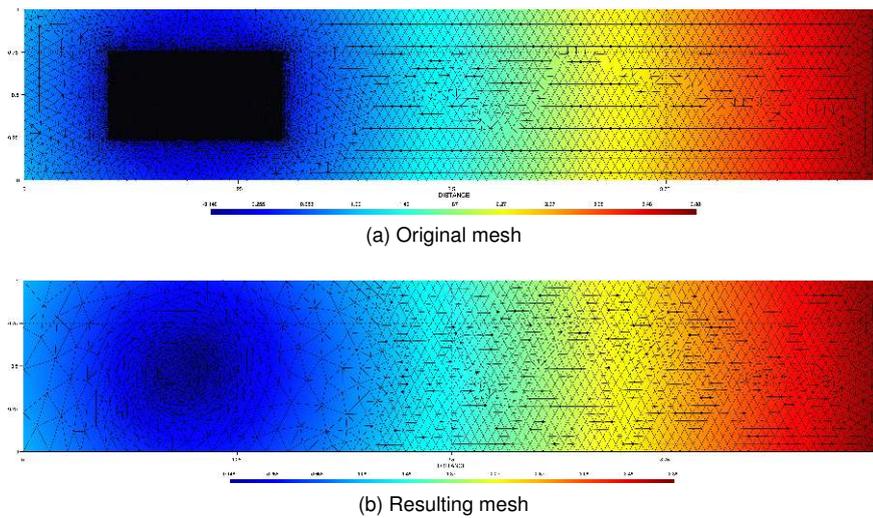


Figure 6.57: Mesh before and after remesh embedded case

6.11.3 Box isosurface example. *Level set*

In this case the challenge consists in meshing anisotropically the geometry using again the gradient of the distance. The initial mesh has a large number of elements (Figure 6.58a), we want to reduce them (Figure 6.58b). Additionally the remesher will consider the isosurface and we will remove the elements with a negative distance (see Figure 6.59). The example can be found in the following [link](#).

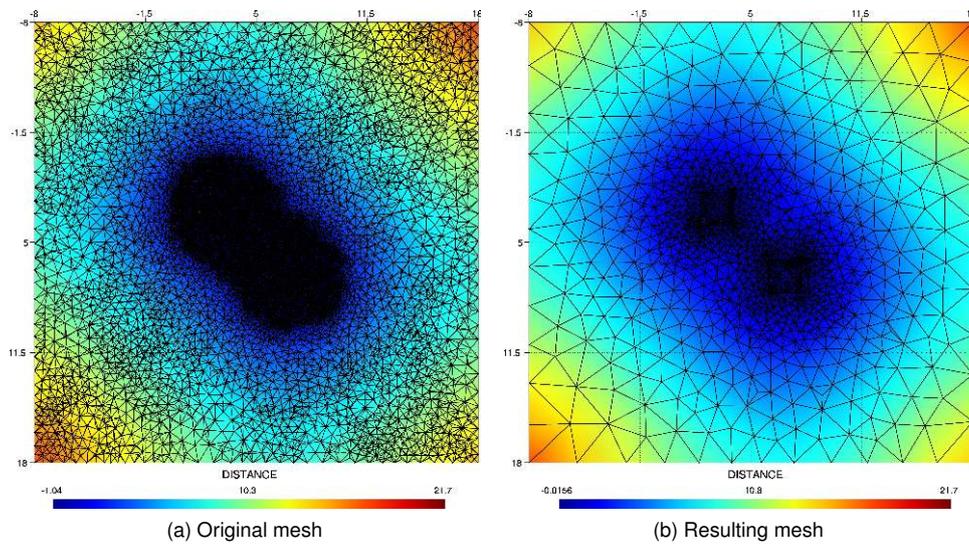


Figure 6.58: Base mesh before and after remesh boxes with isosurface

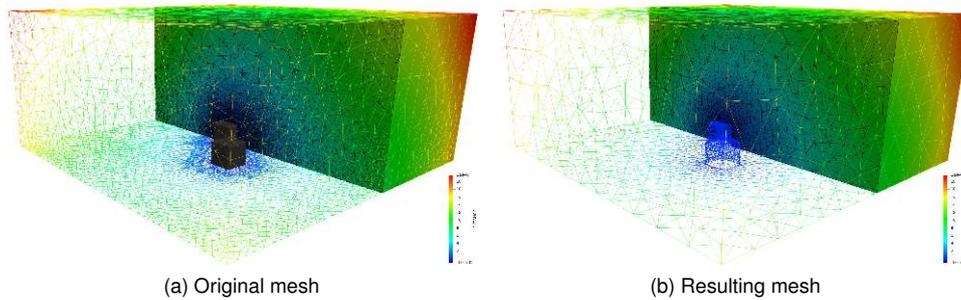
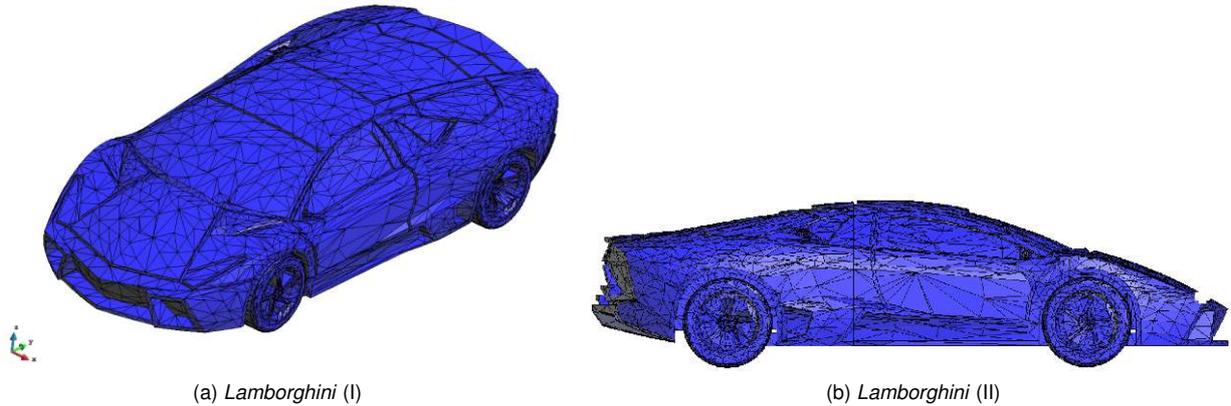


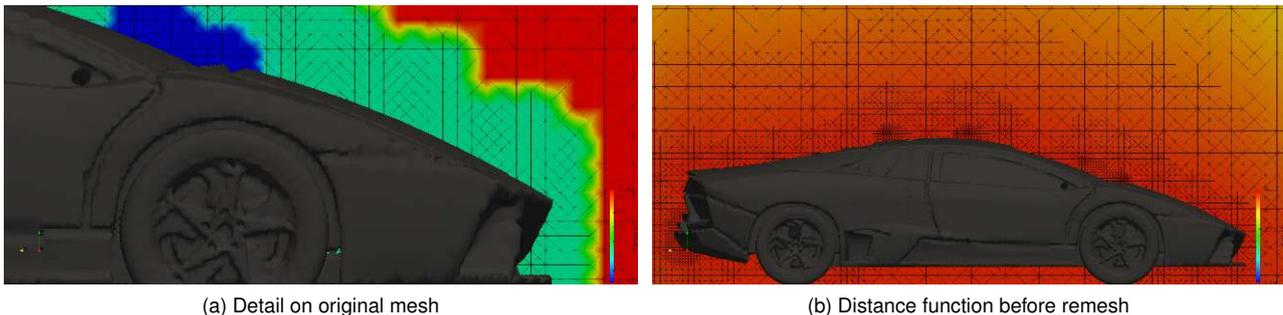
Figure 6.59: Mesh before and after remesh boxes with isosurface

6.11.4 Lamborghini. Level set



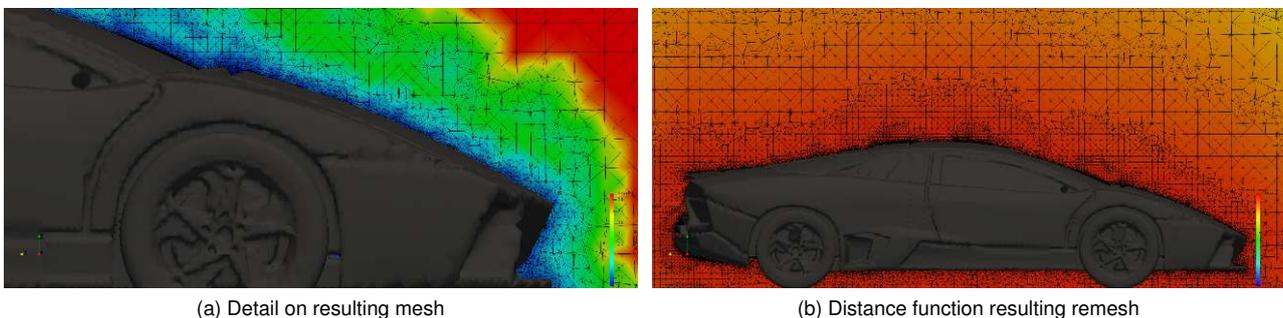
(a) *Lamborghini (I)* (b) *Lamborghini (II)*
Figure 6.60: **Standard Triangle Language (STL)** geometry for the *Lamborghini*

The problem corresponds with the Figure 6.60, the geometry is quite more complex than the previous cases considered. The challenge consists in meshing anisotropically the geometry using the gradient of the distance, previously measured with an octree mesher (*GiD*[OnlMel+16]). The idea behind this is to have a boundary layer around the car, so in case of a **CFD** simulation the details on the aerodynamics study can be captured. The **STL** file used and the rest of required files can be found [here](#). The mesh corresponding before remeshing corresponds with the one shown in Figure 6.61. In here we can appreciate like the mesh elements are isotropic, and does not fit the profile of the car.



(a) Detail on original mesh (b) Distance function before remesh
Figure 6.61: Original mesh for the *Lamborghini*

The resulting mesh can be found in the Figure 6.62, where we see that the mesh follows the profile of the car, and it does in an anisotropic way. Besides, it preserves the distance map from the original mesh.



(a) Detail on resulting mesh (b) Distance function resulting remesh
Figure 6.62: Resulting mesh for the *Lamborghini*

6.11.5 Channel CFD. Hessian of velocity

The following examples consist in two channels, one in 2D and the next one in 3D. In both cases it has been considered the *Hessian* of the velocity to define the metric of the problem.

6.11.5.1 2D cylinder channel

This case consists in an adaptive remeshing of a 2D fluid channel with cylinder using *Hessian* of velocity as metric measure. It consists in a channel 3×1 , a cylinder of $0.5m$ diameter, see Figure 6.63a. Additionally the **BC** are the following: An inlet velocity of $1m/s$, wall conditions in the upper and lower boundary and a zero pressure in the outlet. The total time of the simulation is $5s$ with a time step of $\Delta t = 0.01s$. The remeshing is performed each 20 steps. The mesh corresponding before remeshing corresponds with Figure 6.64a.

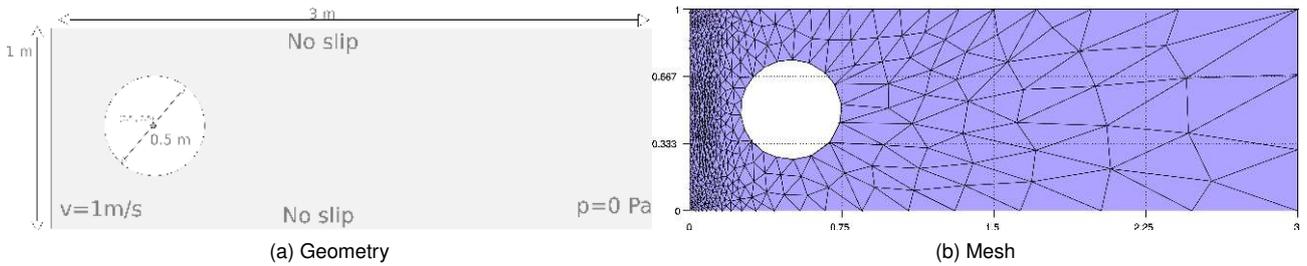


Figure 6.63: Geometry and initial mesh in 2D channel. The example can be found [here](#)

The results obtained correspond with the following, Figure 6.64. In these figures we can appreciate like the mesh fits the movement of the fluid, and particularly have refined the contour of the cylinder.

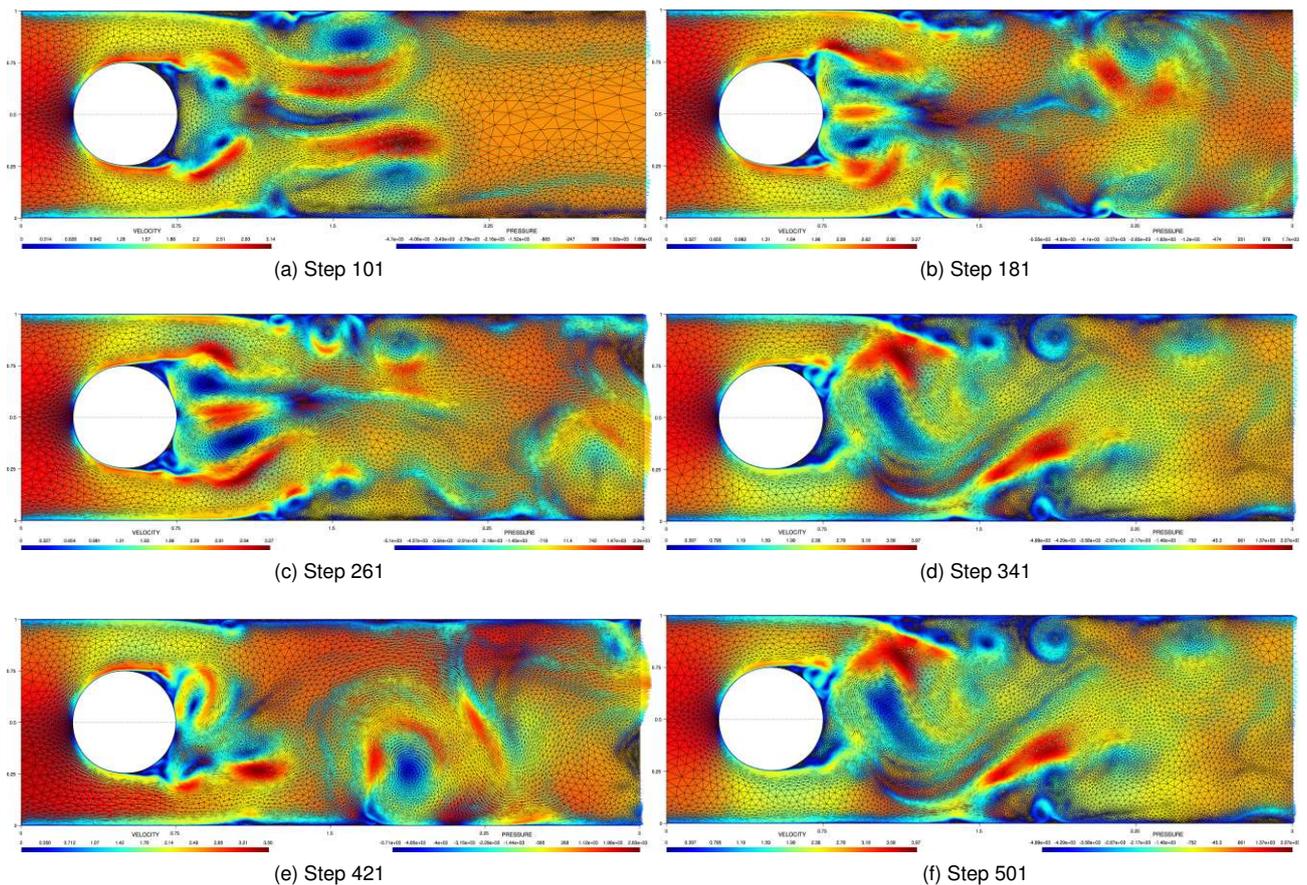


Figure 6.64: Solution in 2D channel

The mesh also captures the whirlpools of the flow and its adapt to them. In the other hand the inlet, which is a **BC** and therefore the velocity is imposed, preserves practically the same mesh in each of the steps.

6.11.5.2 3D sphere channel

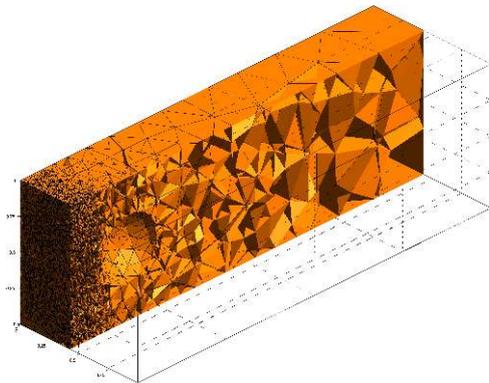


Figure 6.65: Initial mesh in 3D channel

This problem stands as in the previous case in an adaptive remeshing fluid channel with sphere using *Hessian* of velocity as metric measure, but on 3D this time. It consists in a channel $3 \times 1 \times 1$, a sphere of $0.5m$ diameter and with a velocity of $1m/s$ in the inlet an zero pressure in the outlet, the same as Figure 6.63a. The total time of the simulation is $1.2s$ with a time step of $\Delta t = 0.01s$. The remeshing is performed each 20 steps. The initial mesh considered is the one from Figure 6.65. The example can be found in the *Kratos* example repository.

The solution obtained is the one shown in Figure 6.66. On these figures we represent the size of the elements as well as the contour flow of the sphere. In the figures shown, we jump by 20 steps, except in the last three pictures where we increase by 10.

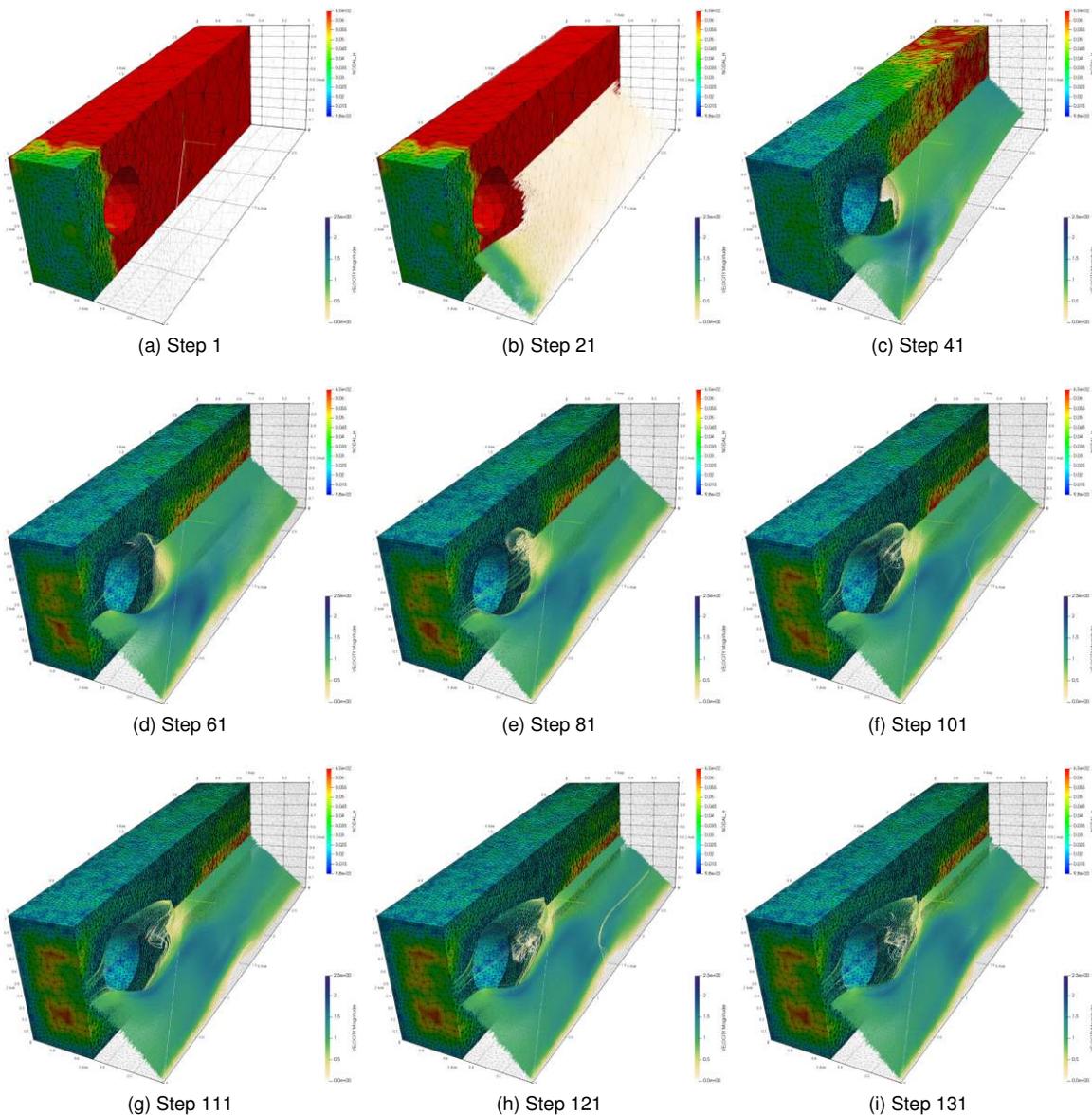


Figure 6.66: Solution in 3D channel

We can see that in the first steps, the size of the elements is similar and bigger than during the rest of the

simulation, especially in the downstream where the velocity where the fluid has not started to flow yet. Once the flow starts to evolve, the elements start to be reduced. But it is significative that the stall of the flow keeps a relative small size of the elements. The smaller elements are always the only surrounding the sphere. Finally, reached certain point the flow around the sphere is fully formed.

Bibliography

Books

- [BookAO11] *A posteriori error estimation in finite element analysis*. Mark Ainsworth and J Tinsley Oden. John Wiley & Sons. 2011.
- [BookCA01] *Introduction to finite element methods*. Felippa C.A. 2001.
- [BookRam01] *Error-controlled adaptive finite elements in solid mechanics*. Ekkehard Ramm E. Rank R. Rannacher K. Schweizerhof E. Stein W. Wendland G. Wittum Peter Wriggers Walter Wunderlich Erwin Stein. John Wiley & Sons. 2001.
- [BookRH12] *Virtual reality in medicine*. Robert Riener and Matthias Harders. Springer Science & Business Media. 2012.
- [BookWri08] *Nonlinear finite element methods*. Peter Wriggers. Springer. 1st ed. 2008.
- [BookZZT13] *The Finite Element Method: its Basis and Fundamentals*. O. C. Zienkiewicz, J.Z. Zhu, and Robert L. Taylor. Butterworth-Heinemann. 7th ed. 2013.

Articles

- [ArtAF03] “Estimateur d’erreur géométrique et métriques anisotropes pour l’adaptation de maillage. Partie I : aspects théoriques”. Frédéric Alauzet and Pascal Frey. In: . No. Rr-4759, 2003,
- [ArtBBP] “Efficient 3D data transfer operators based on numerical integration”. Philippe Bussetta, Romain Boman, and Jean-Philippe Ponthot. In: *International Journal for Numerical Methods in Engineering*. No. 3-4, , pp. 892–929. Wiley Online Library. DOI: [10.1002/nme.4821](https://doi.org/10.1002/nme.4821).
- [ArtBO94] “A methodology for adaptive mesh refinement in optimum shape design problems”. G. Bugeda and E. Oñate. In: *Computing Systems in Engineering*. No. 1, Vol. 5, 1994, pp. 91–102. DOI: [10.1016/0956-0521\(94\)90040-x](https://doi.org/10.1016/0956-0521(94)90040-x).
- [ArtBow81] “Computing dirichlet tessellations”. Adrian Bowyer. In: *The computer journal*. No. 2, Vol. 24, 1981, pp. 162–166. Oxford University Press. DOI: [10.1093/comjnl/24.2.162](https://doi.org/10.1093/comjnl/24.2.162).
- [ArtBR78] “A-posteriori error estimates for the finite element method”. Ivo Babuka and Werner C Rheinboldt. In: *International Journal for Numerical Methods in Engineering*. No. 10, Vol. 12, 1978, pp. 1597–1615. Wiley Online Library.
- [ArtBR79] “Adaptive approaches and reliability estimations in finite element analysis”. I Babuka and WC Rheinboldt. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 17, 1979, pp. 519–540. Elsevier.
- [ArtCas+97] “Anisotropic unstructured mesh adaption for flow simulations”. M. J. Castro-Daz, F. Hecht, B. Mohammadi, and O. Pironneau. In: *International Journal for Numerical Methods in Fluids*. No. 4, Vol. 25, 1997, pp. 475–491. DOI: [10.1002/\(sici\)1097-0363\(19970830\)25:4<475::aid-flid575>3.0.co;2-6](https://doi.org/10.1002/(sici)1097-0363(19970830)25:4<475::aid-flid575>3.0.co;2-6).
- [ArtCea64] “Approximation variationnelle des problèmes aux limites”. Jean Cea. In: *Annales de l’Institut Fourier*. No. 2, Vol. 14, 1964, pp. 345–444. Imprimerie Durand. 28 - LUISANT. DOI: [10.5802/aif.181](https://doi.org/10.5802/aif.181).

- [ArtChi+12] “An adaptive algorithm for cohesive zone model and arbitrary crack propagation”. Vincent Chiaruttini, Dominique Geoffroy, Vincent Riolo, and Marc Bonnet. In: *European Journal of Computational Mechanics*. No. 3-6, Vol. 21, 2012, pp. 208–218. Taylor & Francis. DOI: [10.1080/17797179.2012.744544](https://doi.org/10.1080/17797179.2012.744544).
- [ArtChu80] “Optimal Points Of The Stresses For Triangular Linear Element [J]”. Chen Chuan-miao. In: *Numerical Mathematics A Journal of Chinese Universities*. Vol. 2, 1980,
- [ArtCIA07] “Introduction à l’analyse numérique matricielle et à l’optimisation”. Philippe G CIARLET. In: . 2007, Dunod.
- [ArtCor+19] “Combination of an adaptive remeshing technique with a coupled FEM–DEM approach for analysis of crack propagation problems”. Alejandro Cornejo, Vicente Mataix, Francisco Zárata, and Eugenio Oñate. In: *Computational Particle Mechanics*. 2019, pp. 1–18. Springer. DOI: [10.1007/s40571-019-00306-4](https://doi.org/10.1007/s40571-019-00306-4).
- [ArtFA05] “Anisotropic mesh adaptation for {CFD} computations”. P.J. Frey and F. Alauzet. In: *Computer Methods in Applied Mechanics and Engineering*. No. 48–49, Vol. 194, 2005, pp. 5068–5082. DOI: [10.1016/j.cma.2004.11.025](https://doi.org/10.1016/j.cma.2004.11.025).
- [ArtGJ06] “Plastic model with non-local damage applied to concrete”. Peter Grassl and Milan Jirásek. In: *International Journal for Numerical and Analytical Methods in Geomechanics*. No. 1, Vol. 30, 2006, pp. 71–90. Wiley Online Library. DOI: [10.1002/nag.479](https://doi.org/10.1002/nag.479).
- [ArtLev82] “Stress sampling points for linear triangles in the finite element method”. N Levine. In: *Numerical analysis report*. Vol. 10, 1982, p. 82.
- [ArtLo85] “A new mesh generation scheme for arbitrary planar domains”. SH Lo. In: *International Journal for Numerical Methods in Engineering*. No. 8, Vol. 21, 1985, pp. 1403–1426. Wiley Online Library. DOI: [10.1002/nme.1620210805](https://doi.org/10.1002/nme.1620210805).
- [ArtOB93] “A study of mesh optimality criteria in adaptive finite element analysis”. Eugenio Oñate and G. Bugada. In: *Engineering Computations: Int J for Computer-Aided Engineering*. No. 4, Vol. 10, 1993, pp. 307–321. DOI: [10.1108/eb023910](https://doi.org/10.1108/eb023910).
- [ArtOña+06] “Error estimation and mesh adaptivity in incompressible viscous flows using a residual power approach”. Eugenio Oñate, Joaquin Arteaga, Julio Garca, and Roberto Flores. In: *Computer Methods in Applied Mechanics and Engineering*. No. 4, Vol. 195, 2006, pp. 339–362. DOI: [10.1016/j.cma.2004.07.054](https://doi.org/10.1016/j.cma.2004.07.054).
- [ArtPJR17] “Identifying combinations of tetrahedra into hexahedra: a vertex based strategy”. Jeanne Pellerin, Amaury Johnen, and Jean-Francois Remacle. In: *Procedia engineering*. Vol. 203, 2017, pp. 2–13. Elsevier. DOI: [10.1016/j.proeng.2017.09.779](https://doi.org/10.1016/j.proeng.2017.09.779).
- [ArtPVR18] “There are 174 Subdivisions of the Hexahedron into Tetrahedra”. Jeanne Pellerin, Kilian Verhetsel, and Jean-Francois Remacle. In: . 2018, p. 266. DOI: [10.1145/3272127.3275037](https://doi.org/10.1145/3272127.3275037).
- [ArtRRS95] “Automatic Mesh Generation for Plates and Shells”. Ernst Rank, Martin Rücker, and Manfred Schweingruber. In: *Proc. of 6th ICCCB, Berlin*. 1995,
- [ArtSAS13] “A novel level set-based immersed-boundary method for CFD simulation of moving-boundary problems”. Mukul Shrivastava, Amit Agrawal, and Atul Sharma. In: *Numerical Heat Transfer, Part B: Fundamentals*. No. 4, Vol. 63, 2013, pp. 304–326. Taylor & Francis. DOI: [10.1080/10407790.2013.756258](https://doi.org/10.1080/10407790.2013.756258).
- [ArtTWM82] “Boundary-fitted coordinate systems for numerical solution of partial differential equations—a review”. Joe F Thompson, Zahir UA Warsi, and C Wayne Mastin. In: *Journal of computational Physics*. No. 1, Vol. 47, 1982, pp. 1–108. Elsevier. DOI: [10.1016/0021-9991\(82\)90066-3](https://doi.org/10.1016/0021-9991(82)90066-3).
- [ArtWes+03] “Anisotropic mesh adaption Governed by a Hessian Matrix Metric”. Wilfried Wessner, Hajdin Ceric, Clemens Heitzinger, Andreas Hossinger, and Siegfried Selberherr. In: . 2003,
- [ArtWS98] “Different a posteriori error estimators and indicators for contact problems”. P Wriggers and O Scherf. In: *Mathematical and computer modelling*. No. 4-8, Vol. 28, 1998, pp. 437–447. Elsevier. DOI: [10.1016/s0895-7177\(98\)00133-2](https://doi.org/10.1016/s0895-7177(98)00133-2).

- [ArtYS84] “Automatic three-dimensional mesh generation by the modified-octree technique”. Mark A Yerry and Mark S Shephard. In: *International Journal for Numerical Methods in Engineering*. No. 11, Vol. 20, 1984, pp. 1965–1990. Wiley Online Library. DOI: [10.1002/nme.1620201103](https://doi.org/10.1002/nme.1620201103).
- [ArtZP71] “An automatic mesh generation scheme for plane and curved surfaces by ‘isoparametric’ co-ordinates”. OC Zienkiewicz and DV Phillips. In: *International Journal for Numerical Methods in Engineering*. No. 4, Vol. 3, 1971, pp. 519–528. Wiley Online Library.
- [ArtZZ87] “A simple error estimator and adaptive procedure for practical engineering analysis”. Olgierd C Zienkiewicz and Jian Z Zhu. In: *International journal for numerical methods in engineering*. No. 2, Vol. 24, 1987, pp. 337–357. Wiley Online Library. DOI: [10.1002/nme.1620240206](https://doi.org/10.1002/nme.1620240206).
- [ArtZZ92] “The superconvergent patch recovery (SPR) and adaptive finite element refinement”. OC Zienkiewicz and JZ Zhu. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1-3, Vol. 101, 1992, pp. 207–224. Elsevier. DOI: [10.1016/0045-7825\(92\)90023-d](https://doi.org/10.1016/0045-7825(92)90023-d).
- [ArtZZ95] “Superconvergence and the superconvergent patch recovery”. OC Zienkiewicz and JZ Zhu. In: *Finite elements in analysis and design*. No. 1-2, Vol. 19, 1995, pp. 11–23. Elsevier. DOI: [10.1016/0168-874x\(94\)00054-j](https://doi.org/10.1016/0168-874x(94)00054-j).

Online resources

- [OnIMel+16] *GiD Homepage*. URL: <http://www.gidhome.com/>; <https://www.gidhome.com/support/brand-resources/citing-gid/>.

Part II

Application

Chapter 7

Application

“The hour has arrived to abandon theories and go directly to what is practical.”

Samael Aun Weor
(1917 - 1977 AD, Writer)

7.1 Introduction

This chapter introduces several application test cases, integrating all the formulation and parts developed and presented in this work. The cases presented are mainly taken from the work of Oñate[ArtOZ83]. From this work the cases considered consist in a case with a cylinder punch and a spherical punch. In the original work, the solution was obtained considering an axisymmetric viscous shell formulation when we will consider a full 3D J_2 elasto-plastic model simulation. The solution obtained will be compared with the reference presented in this work.

7.2 Cylinder punch

The geometry of this test is represented in the Figure 7.1. We have transformed the dimensions from the original work in Imperial System to International System. The problem consists in a cylindrical punch, with a blank holder, a die and a sheet of 8.96mm of thickness. In this case the frictional coefficient (μ) is the same in all the contactong interfaces, and equal to 0.2.

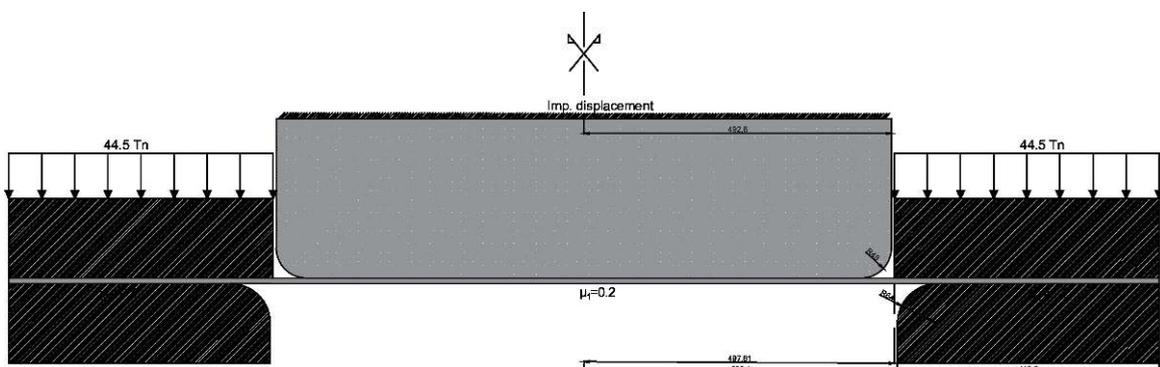
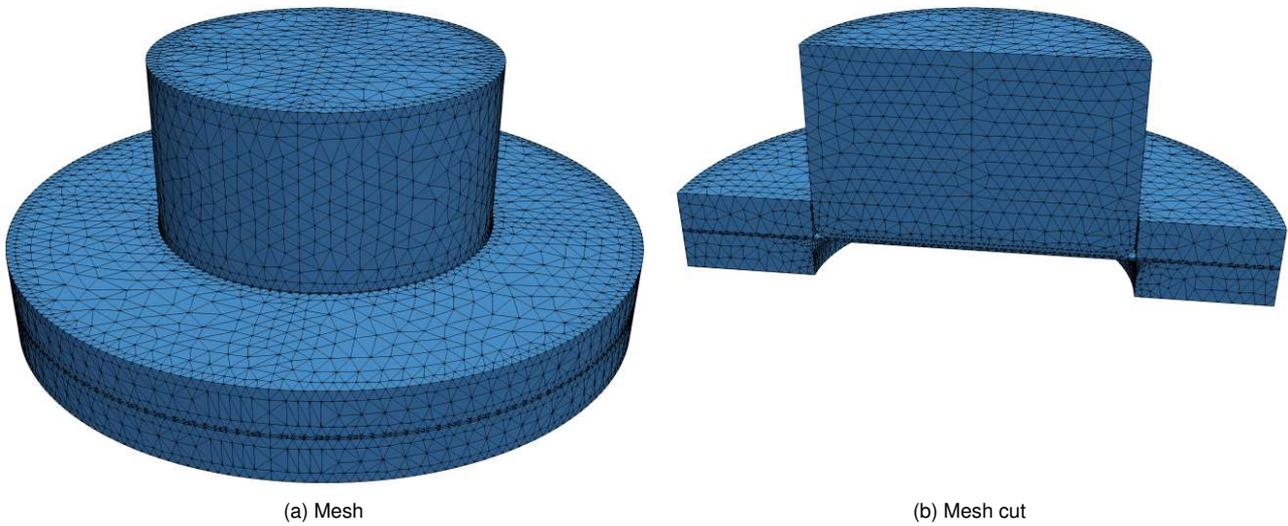


Figure 7.1: Cylinder punch case

Figure 7.2 shows the mesh considered for the simulation of the cylinder punch case. In this case we have

considered the full geometry, instead of considering some simplification taking into account the symmetry of the problem.



(a) Mesh (b) Mesh cut
Figure 7.2: Mesh considered in the cylinder punch case

In the Figure 7.3 we compare the solution obtained with the original experimental results and the numerical solution obtained in the reference. Our solution obtained does not coincide with the experimental solution, but is closer in comparison to the reference numerical solution.

Cylinder punch solution comparison

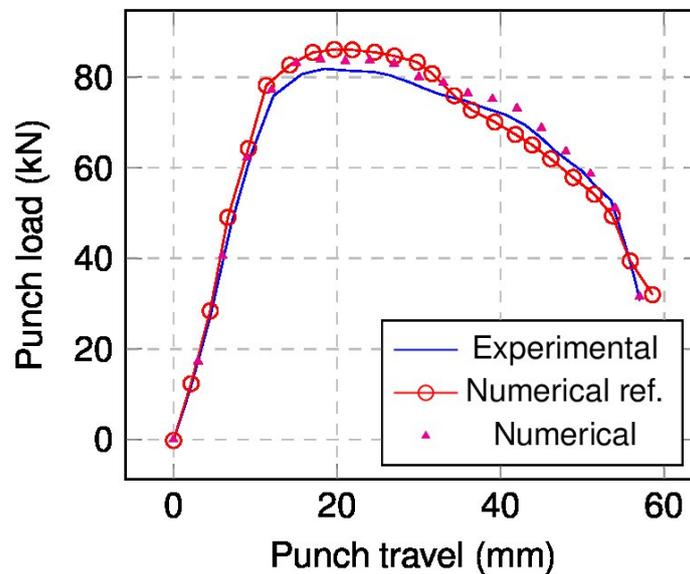


Figure 7.3: Solution for cylinder punch test

7.3 Spherical punch

Figure 7.4 shows the geometry of the spherical punch case. The case consists in a spherical punch, with the corresponding die, a blank holder and a metal sheet. In the original work, the problem was evaluated with three different μ between the sheet and the punch, $\mu_1 = 0.04$, $\mu_1 = 0.2$ and $\mu_1 = 0.5$ in order to be compared with the

experimental result. Here we will just consider only $\mu_1 = \mu_2 = 0.04$, as in the original work the best solution compared with the experimental was obtained for this case.

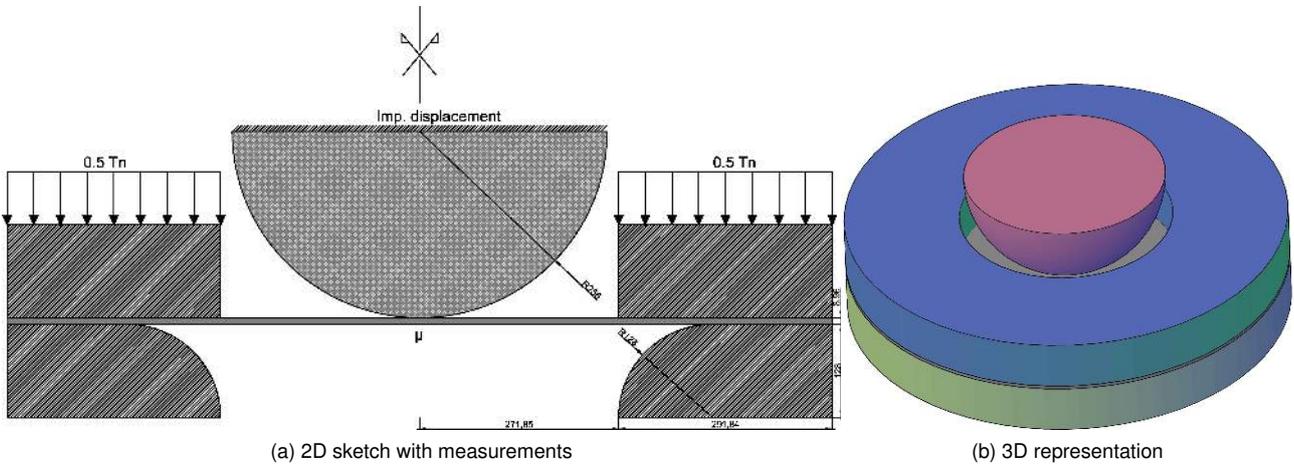


Figure 7.4: Spherical punch case

In this case, instead of a full 3D case, we have considered a quarter of the full geometry, as seen in the Figure 7.5a. The Figure 7.5 also shows the solution obtained at some points of the punch travel.

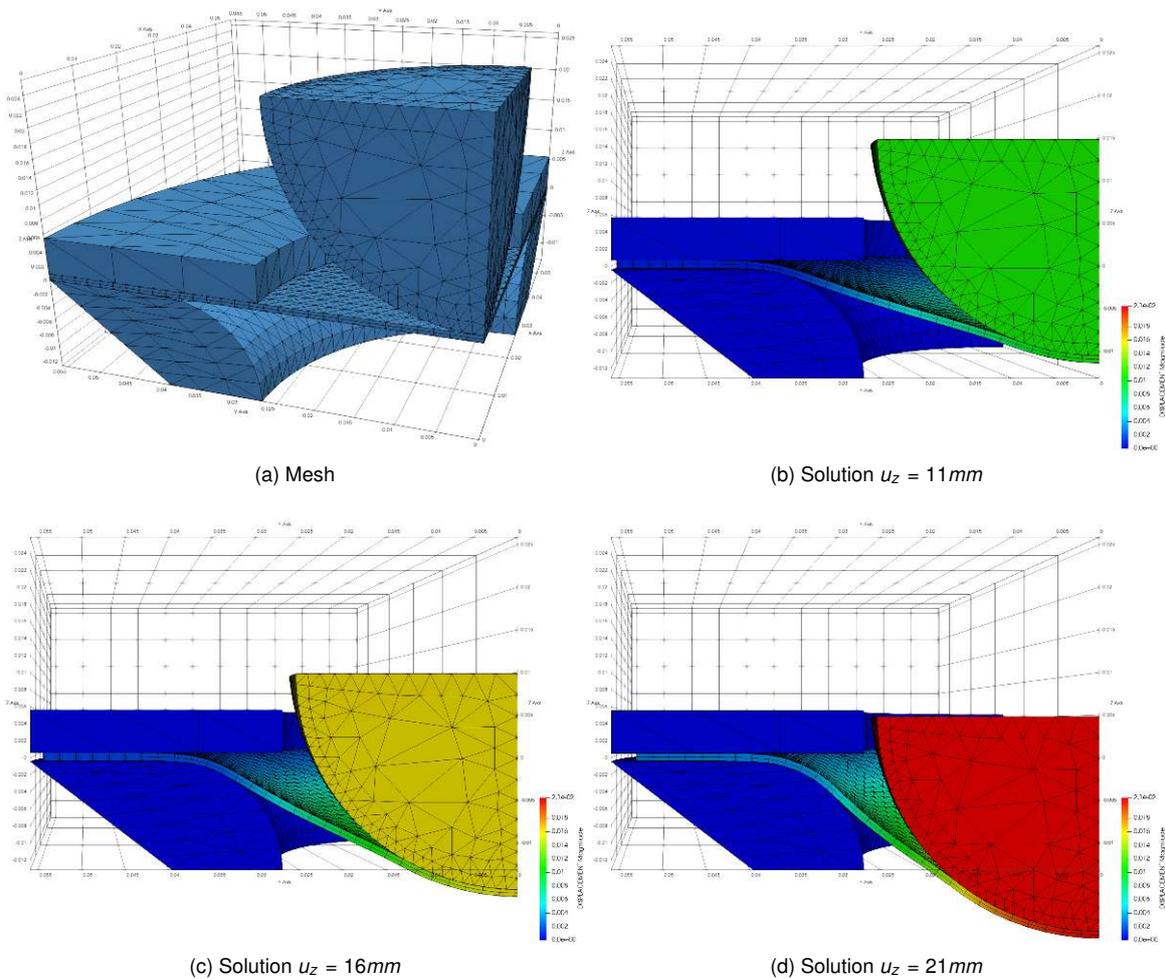


Figure 7.5: 3D solution for the case $\mu_1 = 0.04$ and $\mu_2 = 0.04$

The Figure 7.6 shows a comparison of the solution obtained with the original experimental results and the

numerical solution obtained in the reference for $\mu = 0.04$. Our solution in certain points is closer to the experimental result, but in some of the last steps the difference is greater.

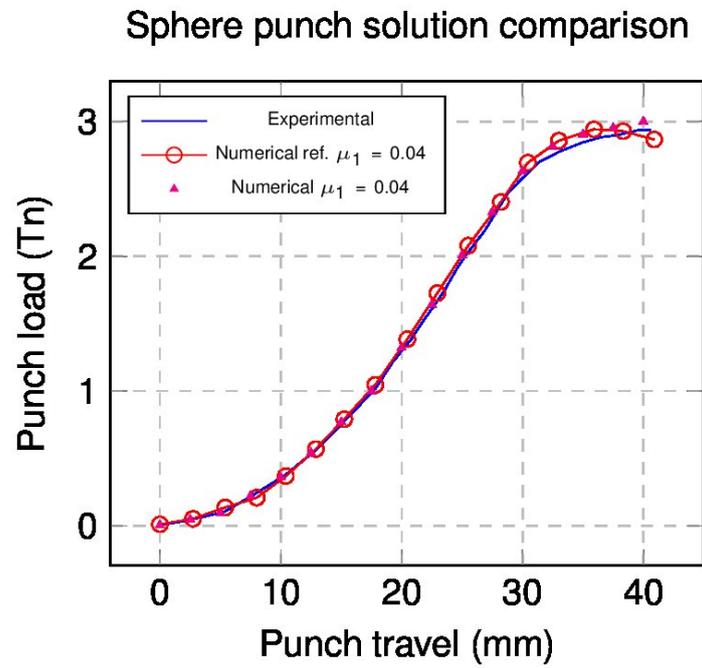


Figure 7.6: Solution for spherical punch test

Bibliography

Articles

- [ArtOZ83] "A viscous shell formulation for the analysis of thin sheet metal forming". E. Onate and O.C. Zienkiewicz. In: *International Journal of Mechanical Sciences*. No. 5, Vol. 25, 1983, pp. 305–335. DOI: [10.1016/0020-7403\(83\)90011-5](https://doi.org/10.1016/0020-7403(83)90011-5).

Part III

Conclusions

Chapter 8

Final conclusions

“Both if you think you can do it
and if you dont, you are in the right
”

Henry T. Ford
(1863 - 1947 AD, Engineer,
Industrialist and Philanthropist)

8.1 Introduction

As can be seen in the previous chapters, the main objectives delimited for this doctorate work have been reached. This means the understanding of the concepts necessary in order to simulate forming processes[BookBan10]. This includes the implementation of a prismatic solid-shell element[ArtOF05; ArtFlo13a; ArtFlo13b] in order to properly simulate the sheet behaviour without the drawbacks of a standard shell element, the proper implementation of a computational contact mechanics algorithm[BookLau10; BookWri06; BookBel+14; ArtYL08; PhDPop12; PhDYas11], the development of elastoplastic CL[BookNPO09] which allow to simulate the forming processes, and the addition of adaptive remeshing techniques to the formula in order to perform the most accurate simulation possible while preserving the minimal computational cost. All of this is not possible without a deep understanding of the FEM applied in the resolution of highly non-linear problems[BookZZT13; BookZTF14; BookBHS07; BookWri08; BookOll14], and how to tackle and couple all these non-linearities between them. In addition, reaching until this point this also means that the structure of the *Kratos* code and C++ programming language have been mastered.

The objectives not reached in the present thesis, as well as other restlessness, are presented in the next chapter 9.Future works. Many significant left open after this work, as the metal forming processes and other similar industrial processes involve problem of a very diverse typology, being very difficult to tackle all of them in just work.

8.2 Rotation free shells and solid-shell elements. Solid-shell

The prismatic solid-shell element implemented (3.4.Prismatic solid-shell) is fully functional, as can be seen in the test cases shown in 3.5.Numerical examples. The element is basically based in the formulation of Flores[ArtFlo13c; ArtFlo13a; ArtFlo13b], but modified in such a way that the methods of **push-forward** and **pull-back** can be used, see A.1.Pull-Back, Push-Forward fundamental concepts, making the element fully compatible with the constitutive laws implemented in *Kratos*. Focusing in the test cases from each one of them, we can conclude that:

- **Cook’s membrane test:** Figures 3.17 represent the behaviour of the *Cook’s* membrane, respectively elastic and elasto-plastic, both obtained considering an implicit scheme. The results present even less than the reference[ArtFlo13b] locking for the elastic case and similar behaviour for the elasto-plastic case.

- **Open ended cylindrical shell test:** The results (Figure 3.23) agree with the solution from [ArtSLL04].
- **Slit test:** Can be seen (Figure 3.27) that with the refinement of the mesh the solution tends to the correct one [ArtSLL04].
- **Panel test:** The panel test is a common test which presents a highly non-linear behaviour, and it is necessary an arc-length to compute. The results obtained, Figures 3.29, show a clear agreement with the reference [ArtSLL04].
- **Sphere test:** The Figure 3.21 presents the results obtained for the semi-spherical shell with a ratio $R/t = 1000$, which agrees with the results from [ArtSLL04].
- **Cone-shell test:** This problem is commonly considered to study the combination both plasticity and geometrical non-linearity, the results shown in Figure 3.31 differ from [ArtKGW06], but with a close behaviour.
- **Wrinkling test:** This behaviour is originated owing to an instability, and needs the consideration of dynamic (kinematic) forces for a correct simulation. The results obtained (Figure 3.33) are in agreement with the reference [ArtFlo13b].

Additionally the element has been proved to be a good alternative to the traditional shells for a **FSI** simulation, in 3.5.14. **FSI-Vein test** a simulation of an elastic vein is performed. The absence of rotational **DOF** as well as the existence of two facets instead of one, differencing the inner and outer face of the structure interacting with the fluid, have proved the suitability of the element for this kind of simulation.

8.3 Contact mechanics

The chapter 4. **Contact mechanics** contains the developments done in this thesis concerning the **CCM**. This chapter constitutes the most relevant piece of work of all the existing document. The chapter starts introducing an historical outline of the **CCM**, as well an introduction on the state-of-art of the contact techniques, compared within them, in order to chose the one finally considered.

This implementation consists in state-of-art of the numerical contact mechanics formulation for implicit simulations, combining the contributions of *Alexander Popp* [PhDPop12; ArtPop+10], *Cavaliere* and *Cardona* [ArtCC12; ArtCC13] and the work of *Yastrebov* [PhDYas11; BookYas13], with off course several contributions of the author.

The formulation, 4.3. **Formulation**, rests in an exact mortar integration of the contact interface. This exact integration in the **FE** geometry allows to obtain the most consistent integration possible between the integration domains, and therefore the most exact solution possible. We have considered different optimisation algorithms, mainly the **LMM**, the **PM** and particularly the consideration of a **ALM** with dual *Lagrange multipliers*, a new contribution of this work, combining the contributions of *Popp* and *Cavaliere*. This dual **LM** allows us to condensate statically the system of equations, taking off the **LM** of the resolution and therefore permitting the consideration of iterative solvers. The formulation has been properly linearised, ensuring the quadratic convergence of the problem. As resolution strategy we have considered a semi-smooth *Newton* method, which consists in an active set strategy, extensible also in the case of frictional problems. The formulation works both for frictionless and frictional problems, the later essential for the simulation of forming processes. In here we have presented the developments corresponding to *Coulomb* friction, but giving the possibility to be extended to any type of frictional model.

In order to properly model the contact simulations, the algorithms of contact detection are detailed. Contact detection is a very relevant subject in the relative of the convergence and resolution of the problem. In 4.4. **Contact detection. Search techniques**, in first place the bounding volumes are presenting, highlighting the **OBB** in 4.4.2.1. **OBB implementation** which is the technique mainly considered. The corresponding section continues with the introduction of the tree structures (4.4.3. **Tree structures**) that can be considered in order to perform a search algorithm. In here also the techniques considered in order to define a consistent gap (4.4.4. **Penetration definition**) and a self-contact algorithm 4.4.5. **Self-contact detection**, both of them newly contributions of this work. Both techniques have been proved successful with a series of tests included in the respective sections.

Respecting the tests, we have evaluated several test, most of them reference in the literature. The most relevant tests can be summarised as follows:

- A basic patch test, [4.5.1.Basic patch test](#) which allows us to evaluate both if the frictionless and frictional formulation work as it supposed. This is expanded with the *Taylor*[ArtTP] patch test [4.5.2.Taylor patch test](#), and it's frictional equivalent [4.5.3.Friction base test](#).
- Then the *Hertz*[ArtHer82] test, the main contact reference problem is tested, [4.5.4.Hertz problem](#), both frictionless and frictional cases. In here, the analytical solution is obtained in both cases.
- The following test is a new contribution of this work. It consists in a pure frictionless case where the energy must be conserved. In here we deduce the analytical solution with the energy conservation principle and compare with the solution obtained. The results agree with the analytical solution, with a very small error, that can be attributed to the numerical dissipation of the time integration scheme. See [4.5.6.Energy conservation](#).
- The double arc benchmark [4.5.7.Double arc benchmark](#), is also a very extended solution in the literature. The solution we obtained corresponds with the expected one, both in the frictionless and in the frictional case.
- The hyperelastic tubes, [4.5.9.Hyperelastic tubes](#) are an interesting solution that includes self-contact. The solution obtained is the expected one.
- The contacting cylinders are another literature reference problem. In [4.5.10.Contacting cylinders](#) the solution obtained is presented.
- The press fit is a frictional validation test, where we obtain almost the same solution as the reference. See [4.5.11.Press fit](#).
- Finally, the ironing tests from [4.5.12.Ironing punch](#), are commonly considered in the literature as validation tests. Our solution corresponds with the one found in the literature.

The chapter concludes with the deduction of the directional derivatives, which are necessary for a proper linearisation of the problem and obtain the desired quadratic convergence of the problem. The derivatives obtained have been shown to converge quadratically as expected and necessary.

8.4 Plasticity

In [5.Plasticity](#) the developments concerning the elasto-plastic models considered is presented. This chapter also starts with a short introduction of the state-of-art of these constitutive models, as well as a very short historical outline.

The main contribution of this work has been, the introduction a modular large strain elasto plastic model. This is noted as modular, as it allows the combination of arbitrary of yield surfaces and plastic potential, the so-called *non-associated* elasto-plastic **CL**. The implementation details of these modular classes are shown in [5.5.2.Class structure in Kratos](#). This implementation takes as base the developments from *Cornejo*[PhDCor], and extends it in order to deal with large strain behaviours.

In order to deal with the computation of the consistent tangent tensor on these non-associate constitutive laws, a numerical approach has been considered in [5.5.3.Numerical implementation tangent constitutive tensor](#). In order to obtain this consistent \mathbb{C} we proceed with a numerical procedure based in the stress vector perturbation method. Two different approaches are shown, one of first order and another one of second order, with its corresponding computational cost, precision and convergence rate.

Additionally, the chapter presents some numerical examples in order to prove that the methodology works. Starting from very simple cases of validation as the minimal cube considered in [5.6.2.Cube minimal example](#), where the **CL** has been compared with a perfect J_2 of reference, providing the correct solution. The next case tested consisted in a more complex tensile test, from [5.6.3.Tensile test](#), where the solution obtained corresponds with the expected one, but also showing a slightly mesh dependency in the solution. The final test consisted in a **CCM** test, [5.6.4.Application in CCM. Gears example](#), consisting in two gears dealing with contact in a very small contact region, showing the possibility of coupling these two kinds of phenomena simultaneously.

8.5 Adaptive remeshing

The chapter [6.Adaptative remeshing](#) contains the developments done in the field of the adaptive remeshing. The chapter includes an initial introduction and state-of-art of the field, that can result very practical in anyone been introduced in the topic. Aside of this, we have shown different approaches, from more conventional **SPR** error measures with isotropic remeshing, as seen in [6.5.SPR based remeshing technique](#); to approaches based in the computation of the *Hessian* error measure, which in contrast provides an **anisotropic** mesh, seen in [6.3.Hessian based remeshing technique](#). Both techniques have been compared, and have been proven to be very capable in order to work with structural problems, but in the case of *Hessian* metric we are able to consider in a larger range of cases.

Additionally both techniques have been adapted to the contact mechanics problem [6.8.Adaptive remeshing methods applied on CCM](#), which is the main contribution of this work. The case of the **SPR** consists in the adaptation of the previous work of *Wriggers*[\[ArtWS98\]](#) to the **ALM** contact formulation developed. The real novelty consists in the *Hessian* error measure for **CCM**, where we have considered the intersection of the different contact related metrics, like the contact pressure and the **VM** stress. Both techniques have ben shown effective, but the results got from the *Hessian* technique show to be more promising.

In addition to the methods previously mentioned, techniques able to deal with both interpolation and extrapolation of integration values have been implemented in order to accommodate some needs during the remeshing process. These techniques are detailed respectively in [6.6.Internal values interpolation](#) and [6.7.Integration points values extrapolation](#).

In the relative of the test cases, the chapter includes an extensive list of tests, this includes structural problems from [6.10.Numerical examples](#) and the **CFD** application examples from [6.11.CFD numerical examples](#). This last category shows how the *Hessian* metric is general enough to deal with a different type of physical problems, particularly interesting the channel cases from [6.11.5.Channel CFD. Hessian of velocity](#). The example more characteristic, and the pride of this work is the adaptive remeshing case of the contacting cylinders in large deformations from [6.10.5.Contacting cylinders with adaptive remeshing](#).

The library considered in order to implement these remeshing techniques, **Mmg**, is introduced in its corresponding [Appendix B.3.Mmg library](#).

8.6 Application cases

In the final chapter [7.Application](#), which additionally corresponds with the final part of this work. What this chapter shows is that an initial capability to perform metal forming processes is proven. The combination of the former points have allowed to perform these simulations. Additionally, as short summary of the chapter, the results obtained differ slightly with the results present in the main reference[\[ArtOZ83\]](#), but in the comparison with the experimental results the solution presented resembles more to the latter.

8.7 Other remarks

In here we will present others remarks, that not constitute the main subject of this work, but we consider important to append to what have been said until this point, in order to have a more accurate picture of what has been done and achieved.

8.7.1 Other developments

The appendices include different developments tangent to main developments already presented. For example, from the mortar formulation considered **CCM** problems can be considered in order to compute mesh tying problems [A.3.Mesh tying](#), or an *explicit* mapping algorithm (**E.Mortar mapper**). This mapping algorithm has allowed to compute a consistent gap with the mortar formulation considered during the contact pair search. Also the details of the mortar segmentation algorithms considered can be found in [A.2.Mortar segmentation](#).

In order to obtain a consistent formulation for the contact mechanics problems, meaning with quadratic convergence, the **AD**[[ArtGru82](#); [ArtKor97](#); [ArtKor02](#); [BookWri08](#)] methodology has been considered. This methodology is presented in an independent Appendix, see [C. Automatic differentiation](#).

Also, as a complement, which helps in a proper understanding of the constraint optimisation considered in the contact mechanics formulations, a complete Appendix has been considered, [D. Constrained optimisation problems](#). Where the mathematical concepts and details of these methodologies are presented.

8.7.2 Implicit approach

All the simulations and tests have been performed with an **implicit** approach, as one of the mains aims of this research is to focus in this kind of solution. Some of the problems computed in this work are highly nonlinear, especially due to the challenging nonlinearities associated with the computational contact and elastoplastic constitutive laws. This has also implied the necessity of improving the implementation of algorithms related with the implicit resolution of the system of equations. This includes the improvement of the **NR** with the inclusion of active-set strategies, necessities in order to properly consider the contact problem in an implicit manner. Other developments not included in this work, but that have helped to improve the implicit approach include the implementation of different constraints imposing techniques with the consideration of **MultiPoint Constraint (MPC)**, as well as different systems of equations assemblers.

8.7.3 Programming

As a final remark, and non-necessarily related with the subject treated in this work, we find relevant to highlight that in order to achieve the results and solutions presented the author has learnt to deal with collaborative development for the **Kratos** project. The structure and fundamentals of the **Kratos** code has been understood, but also the understanding the workflow of a version control software as **git** is. The main characteristics of **Kratos** are introduced in its respective Appendix [B.2. Kratos Multiphysics](#). The author has become an important member of the **Kratos** community, and we feel the need to mention.

Bibliography

Books

- [BookBan10] *Sheet Metal Forming Processes: Constitutive Modelling and Numerical Simulation*. Dorel Banabic. Springer-Verlag Berlin Heidelberg. 1st ed. 2010.
- [BookBel+14] *Nonlinear Finite Elements for Continua and Structures*. Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. Wiley. 2nd ed. 2014.
- [BookBHS07] *Encyclopedia of Computational Mechanics (3 Volume Set)*. René de Borst, T.J.R. Hughes, and Erwin Stein. Wiley. 1st ed. 2007.
- [BookLau10] *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Tod A. Laursen. Springer. 2010.
- [BookNPO09] *Computational Methods for Plasticity Theory and Applications*. E. A. de Souza Neto, D. Perić, and D.R.J. Owen. Wiley. 2009.
- [BookOll14] *Nonlinear Dynamics of Structures*. Sergio Oller. Springer International Publishing. 2014.
- [BookWri06] *Computational Contact Mechanics*. Peter Wriggers. Springer. 2nd. 2006.
- [BookWri08] *Nonlinear finite element methods*. Peter Wriggers. Springer. 1st ed. 2008.
- [BookYas13] *Numerical Methods in Contact Mechanics*. Vladislav A. Yastrebov. Wiley-ISTE. 1st ed. 2013.
- [BookZTF14] *The Finite Element Method for Solid and Structural Mechanics*. O. C. Zienkiewicz, Robert L. Taylor, and David Fox. Butterworth-Heinemann. 7th ed. 2014.
- [BookZZT13] *The Finite Element Method: its Basis and Fundamentals*. O. C. Zienkiewicz, J.Z. Zhu, and Robert L. Taylor. Butterworth-Heinemann. 7th ed. 2013.

Articles

- [ArtCC12] “An augmented lagrangian method to solve three-dimensional nonlinear contact problems”. FJ Calavaleri and Alberto Cardona. In: *Latin American applied research*. No. 3, Vol. 42, 2012, pp. 281–289. SciELO Argentina.
- [ArtCC13] “An augmented Lagrangian technique combined with a mortar algorithm for modelling mechanical contact problems”. FJ Cavalieri and A Cardona. In: *International Journal for Numerical Methods in Engineering*. No. 4, Vol. 93, 2013, pp. 420–442. Wiley Online Library. DOI: [10.1002/nme.4391](https://doi.org/10.1002/nme.4391).
- [ArtFlo13a] “A “Prism” solid element for large strain shell analysis”. Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 253, 2013, pp. 274–286. Elsevier. DOI: [10.1016/j.cma.2012.10.001](https://doi.org/10.1016/j.cma.2012.10.001).
- [ArtFlo13b] “Development of a non-linear triangular prism solid-shell element using ANS and EAS techniques”. Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 266, 2013, pp. 81–97. Elsevier. DOI: [10.1016/j.cma.2013.07.014](https://doi.org/10.1016/j.cma.2013.07.014).
- [ArtFlo13c] “Un Elemento Prisma Triangular De Sólido-lámina Para El Análisis Con Grandes Deformaciones”. Fernando G. Flores. In: *Mecánica Computacional*. Vol. Xxxii, 2013, pp. 63–87.

- [ArtGru82] "Automatic Differentiation: Techniques and Applications. Lecture Notes in Computer Science 120." F. Grund. In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*. No. 7, Vol. 62, 1982, pp. 355–355. DOI: [10.1002/zamm.19820620735](https://doi.org/10.1002/zamm.19820620735).
- [ArtHer82] "Über die berührung fester elastische Körper und über die Harte". H Hertz. In: *Verhandlungen des Vereins zur Beförderung des Gewerbefleisses*. 1882,
- [ArtKGW06] "A robust non-linear solid shell element based on a mixed variational formulation". Sven Klinkel, Friedrich Gruttmann, and Werner Wagner. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1–3, Vol. 195, 2006, pp. 179–201. Elsevier. DOI: [10.1016/j.cma.2005.01.013](https://doi.org/10.1016/j.cma.2005.01.013).
- [ArtKor02] "Multi-language and Multi-environment Generation of Nonlinear Finite Element Codes". J. Korelc. In: *Engineering with Computers*. No. 4, Vol. 18, 2002, pp. 312–327. DOI: [10.1007/s003660200028](https://doi.org/10.1007/s003660200028).
- [ArtKor97] "Automatic generation of finite-element code by simultaneous optimization of expressions". Joe Korelc. In: *Theoretical Computer Science*. No. 1, Vol. 187, 1997, pp. 231–248. DOI: [10.1016/s0304-3975\(97\)00067-4](https://doi.org/10.1016/s0304-3975(97)00067-4).
- [ArtOF05] "Advances in the formulation of the rotation-free basic shell triangle". Eugenio Oñate and Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. No. 21–24, Vol. 194, 2005, pp. 2406–2443. DOI: [10.1016/j.cma.2004.07.039](https://doi.org/10.1016/j.cma.2004.07.039).
- [ArtOZ83] "A viscous shell formulation for the analysis of thin sheet metal forming". E. Onate and O.C. Zienkiewicz. In: *International Journal of Mechanical Sciences*. No. 5, Vol. 25, 1983, pp. 305–335. DOI: [10.1016/0020-7403\(83\)90011-5](https://doi.org/10.1016/0020-7403(83)90011-5).
- [ArtPop+10] "A dual mortar approach for 3D finite deformation contact with consistent linearization". Alexander Popp, Markus Gitterle, Michael W. Gee, and Wolfgang A. Wall. In: *International Journal for Numerical Methods in Engineering*. No. 11, Vol. 83, 2010, pp. 1428–1465. John Wiley & Sons, Ltd. DOI: [10.1002/nme.2866](https://doi.org/10.1002/nme.2866).
- [ArtSLL04] "Popular benchmark problems for geometric nonlinear analysis of shells". K.Y. Sze, X.H. Liu, and S.H. Lo. In: *Finite Elements in Analysis and Design*. No. 11, Vol. 40, 2004, pp. 1551–1569. DOI: [10.1016/j.finel.2003.11.001](https://doi.org/10.1016/j.finel.2003.11.001).
- [ArtTP] "On a patch test for contact problems in two dimensions". Robert L. Taylor and Panagiotis Panagiotopoulos. In: . ,
- [ArtWS98] "Different a posteriori error estimators and indicators for contact problems". P Wriggers and O Scherf. In: *Mathematical and computer modelling*. No. 4-8, Vol. 28, 1998, pp. 437–447. Elsevier. DOI: [10.1016/s0895-7177\(98\)00133-2](https://doi.org/10.1016/s0895-7177(98)00133-2).
- [ArtYL08] "A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations". Bin Yang and TodA. Laursen. In: *Computational Mechanics*. No. 2, Vol. 41, 2008, pp. 189–205. Springer-Verlag. DOI: [10.1007/s00466-006-0116-z](https://doi.org/10.1007/s00466-006-0116-z).

Ph.D.'s thesis

- [PhDCor] "Coupled formulation between the finite element method and the discrete element method for studying multi-fracture processes in solids and structures". Alejandro Cornejo.
- [PhDPop12] "Mortar Methods for Computational Contact Mechanics and General Interface Problems". Alexander Popp. 2012.
- [PhDYas11] "Computational contact mechanics: geometry, detection and numerical techniques". Vladislav A. Yastrebov. 2011.

Chapter 9

Future works

“The future belongs to those who believe in the beauty of their dreams”

Eleanor Roosevelt
(1884 - 1962 AD, American political figure, diplomat and activist)

In this chapter we present the future developments that will follow to this work. The following are ideas that due to the time constraints, and to limit the study of this work, have not been included in this final work. We will split, the future works, relating with each one of the subjects tackled in these pages.

9.1 Solid-shell implementations

- Adapt the element from 3.4.Prismatic solid-shell, in order to properly support the **UL** framework.
- The reduced integration hexahedral solid-shell element from *Flores*[ArtFlo16] could be a good element in order to complement the element presented on this work in 3.4.Prismatic solid-shell.

9.2 Contact mechanics

- The implementations presented on this work are mainly focused on serial implementations and only **OpenMP**. In order to make these implementations truly **High Performance Computing (HPC)** it is necessary to extend the formulation to **Message Passing Interface (MPI)**, particularly using the **Trilinos API** exposed in **Kratos**.
- Improve the robustness of the method with iterative solvers, particularly with **AMG** solvers, that **Kratos** makes use of.
- Trying to bring to **Kratos** the latest implementations emerged in the literature, like the matrix-symmetric contact problem presented in [ArtHWP18].
- Extend the formulations in order to consider different formulations. Like the **Material Point Method (MPM)** or the **Iso-Geometric Analysis (IGA)**[ArtLWZ12a; ArtLWZ12b; ArtSei+16] formulation. These formulations have already initial implementations of contact constraints in the current **Kratos**.
- Extend the contact formulation to other types of elements, like shells or beams[ArtSN17], and combinations of these elements between them.
- Clean up, and refactor of the code in **Kratos**. Particularly reduce the compilation and linking times.

- Consideration of the contact formulation, and couple it with problems of another nature like **FSI**.
- Extension and modularisation of the frictional models, in order to be able to consider general frictional models of any type.
- Application to optimisation problems[Bookaut; BookAll07].

9.3 Plasticity

- The definition of thermomechanical behaviour. The definition of thermic deformations, as well as others thermomechanical behaviours and couplings. Particularly focused in large deformation formulations[ArtLub04].
- The consideration, and extension, of general and large deformation anisotropic material behaviours. This is necessary in order to properly simulate som metal stamping phenomena[ArtCJL02; ArtGW02].
- Coupling of **Kratos** with the code *MFront*[ArtHPF15].
- Extend the testing and validation of the **CL** in the **Kratos** framework.

9.4 Adaptive remeshing

- Extension of the adaptive remeshing techniques to **HPC** and **MPI**.
- Application of the adaptive remeshing techniques developed on this work in problems of different nature and couplings. For example, consider it in order to remesh the background mesh present in the **MPM** formulation.
- Enhance current implementations in order to remesh element types not supported currently, like hexahedra.

9.5 Metal stamping

- Improve and extend the implementations focused on metal stamping simulations.
- Validate the code considering more extended benchmarks.

9.6 Other points

- Refactor, clean up and improvement of the existing code of **Kratos** in all levels.
- Extension of the current **AD**[ArtKor97; ArtKor02] implementations, in order to simplify its use and its generality. It would be nice to apply this technique, not only in the derivation of element, but the definition of **CL** or frictional laws.
- Extend and improve the explicit formulation that was developed and dropped during the development of this work.
- Extend the current **Graphic User Interface (GUI)** implementations.
- Extend **Kratos** integration with new libraries and features.

Bibliography

Books

- [BookAll07] *Conception optimale de structures (Mathématiques et Applications)*. Grégoire Allaire. Springer. 1st ed. 2007.
- [Bookaut] *Shape Optimization by the Homogenization Method*. Grégoire Allaire (auth.) Springer-Verlag New York. 1st ed.

Articles

- [ArtCJL02] “Modeling the Bauschinger effect for sheet metals, part I: theory”. B.K. Chun, J.T. Jinn, and J.K. Lee. In: *International Journal of Plasticity*. No. 5–6, Vol. 18, 2002, pp. 571–595. DOI: [10.1016/s0749-6419\(01\)00046-8](https://doi.org/10.1016/s0749-6419(01)00046-8).
- [ArtFlo16] “A simple reduced integration hexahedral solid-shell element for large strains”. Fernando G. Flores. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 303, 2016, pp. 260–287. DOI: [10.1016/j.cma.2016.01.013](https://doi.org/10.1016/j.cma.2016.01.013).
- [ArtGW02] “Role of plastic anisotropy and its evolution on springback”. Lumin Geng and R.H. Wagoner. In: *International Journal of Mechanical Sciences*. No. 1, Vol. 44, 2002, pp. 123–148. DOI: [10.1016/s0020-7403\(01\)00085-6](https://doi.org/10.1016/s0020-7403(01)00085-6).
- [ArtHPF15] “Implantation de lois de comportement mécanique à l'aide de MFront: simplicité, efficacité, robustesse et portabilité”. Thomas Helfer, Jean-Michel Proix, and Olivier Fandeur. In: . 2015, Giens, France.
- [ArtHWP18] “A truly variationally consistent and symmetric mortar-based contact formulation for finite deformation solid mechanics”. Michael Hiermeier, Wolfgang A. Wall, and Alexander Popp. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 342, 2018, pp. 532–560. DOI: [10.1016/j.cma.2018.07.020](https://doi.org/10.1016/j.cma.2018.07.020).
- [ArtKor02] “Multi-language and Multi-environment Generation of Nonlinear Finite Element Codes”. J. Korelc. In: *Engineering with Computers*. No. 4, Vol. 18, 2002, pp. 312–327. DOI: [10.1007/s003660200028](https://doi.org/10.1007/s003660200028).
- [ArtKor97] “Automatic generation of finite-element code by simultaneous optimization of expressions”. Joe Korelc. In: *Theoretical Computer Science*. No. 1, Vol. 187, 1997, pp. 231–248. DOI: [10.1016/s0304-3975\(97\)00067-4](https://doi.org/10.1016/s0304-3975(97)00067-4).
- [ArtLub04] “Constitutive theories based on the multiplicative decomposition of deformation gradient: Thermoelasticity, elastoplasticity, and biomechanics”. Vlado A Lubarda. In: *Applied Mechanics Reviews*. No. 2, Vol. 57, 2004, pp. 95–108. DOI: [10.1115/1.1591000](https://doi.org/10.1115/1.1591000).
- [ArtLWZ12a] “A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method”. L. de Lorenzis, Peter Wriggers, and G. Zavarise. In: *Computational Mechanics*. No. 1, Vol. 49, 2012, pp. 1–20. Springer. DOI: [10.1007/s00466-011-0623-4](https://doi.org/10.1007/s00466-011-0623-4).

- [ArtLWZ12b] “A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method”. L. De Lorenzis, Peter Wriggers, and G. Zavarise. In: *Computational Mechanics*. No. 1, Vol. 49, 2012, pp. 1–20. Springer-Verlag. DOI: [10.1007/s00466-011-0623-4](https://doi.org/10.1007/s00466-011-0623-4).
- [ArtSei+16] “Isogeometric dual mortar methods for computational contact mechanics”. Alexander Seitz, Philipp Farah, Johannes Krehmeller, Barbara I. Wohlmuth, Wolfgang A. Wall, and Alexander Popp. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 301, 2016, pp. 259–280. DOI: [10.1016/j.cma.2015.12.018](https://doi.org/10.1016/j.cma.2015.12.018).
- [ArtSN17] “Woven fabrics computational simulation using beam-to-beam contacts formulation”. Mauro Takayama Saito and Alfredo Gay Neto. In: *Revista Interdisciplinar De Pesquisa Em Engenharia*. No. 22, Vol. 2, 2017, pp. 09–25.

Appendices

Appendix A

Theoretical complements

“The devil is in the detail.”

Ludwig Mies van der Rohe
(1886 - 1969 AD, German-born
architect)

A.1 Pull-Back, Push-Forward fundamental concepts

In this appendix we introduce the concepts of **pull-back** and **push-forward** necessities to understand the concepts introduced previously, the main reference for this has been taken from [BookBel+14]. These operations allow us to have a unified description between the *Eulerian* and *Lagrangian* tensors. In (A.1) we can appreciate some examples.

A push-forward by \mathbf{F} of the *Lagrangian* vector $d\mathbf{X}$ to the current configuration gives the *Eulerian* vector $d\mathbf{x}$:

$$(A.1a) \quad d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X} \equiv \phi_* d\mathbf{X}$$

The pull-back by \mathbf{F}^{-1} of the *Eulerian* vector $d\mathbf{x}$ to the reference configuration gives $d\mathbf{X}$.

$$(A.1b) \quad d\mathbf{X} = \mathbf{F}^{-1} \cdot d\mathbf{x} \equiv \phi^* d\mathbf{x}$$

Where ϕ_* and ϕ^* represent the **push-forward** and **pull-back** operations respectively.

If we extend these operations, we can consider the **pull-back** and **push-forward** operations on second-order tensors to obtain the relationships between these tensors in the deformed and undeformed configurations. These concepts provide us a mathematically consistent method for define the pseudo-time derivatives of the tensors, called **Lie-derivatives**. Considering all this, the operations can be considered for example to obtain a formulation of **hyperelastic-plastic** constitutive model based on the multiplicative decomposition of the deformation gradient (\mathbf{F}), for more information we address [BookBel+14].

A.2 Mortar segmentation

A.2.1 Introduction

In this section we analyse in detail, the segmentation method chooses in order to compute the mortar contact constraints. In a first section, the discussion will be focused around the integration method considered, in a second section the triangulation methods considered on 3D will be presented.

A.2.2 Exact integration vs Collocation

A.2.2.1 Theory

In this section we study the two different approaches for the mortar method, the exact integration (also denominated *segment base*) and the collocation method (usually called *element base*). The first one as the method indicates, consists on the decomposition and exact integration of the master and slave segments (Figure A.1), and is the approach follows in Popp[PhDPop12]. The second alternative (Figure A.2) consists in consider a big number of GP on the integration of the pair, the GP inside the opposite pair are considered into integration, and the ones outside are discarded.

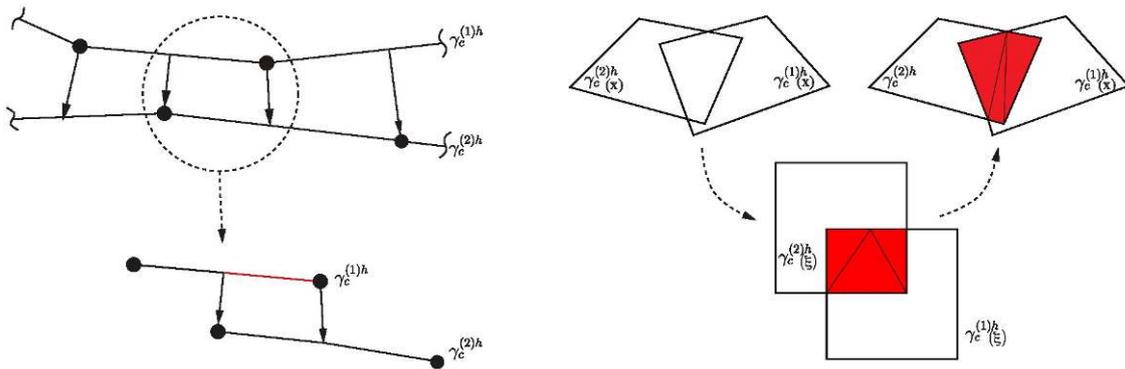


Figure A.1: Exact segmentation method for integration. Figure inspired on Popp[PhDPop12]

On the latter method, the integration points are not distributed following any Gauss quadrature, but distributed uniformly in all the geometry, that's why the method is denominated collocation method. If the number of collocation points is big enough, the result will be close to the exact solution. Apart from the mortar methods, this approach is extensively considering on the isogeometric analysis[ArtAur+10; ArtSch+13].

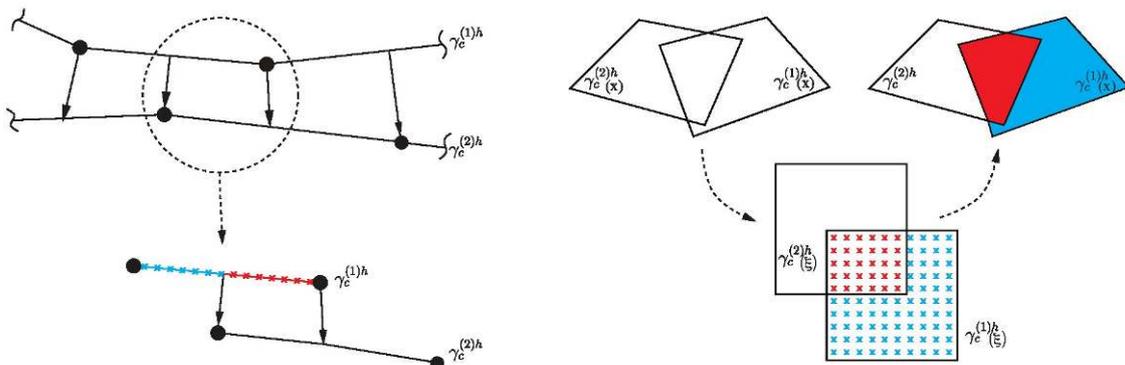


Figure A.2: Collocation method for integration. Figure inspired on Popp[PhDPop12]

From the point of view of the implementation, the *element base* is a significantly easier method to implement, as there are not many modifications to a standard FE implementation, only a check for the inside/outside GP. Additionally the values and derivatives definitions are exactly equal in 2D and 3D, additionally do not depend on the segmentation, being considerably easier.

On the other hand, the exact integration, particularly complex on 3D, which requires to property implement the clipping procedure (4.105) and triangulation (see A.2.3.Delaunay vs Convex polygon construct), in addition to the corresponding derivatives from 4.6.2.Derivatives for 3D contact. This is quire more complex, but as it will be shown (A.2.2.2.Solution study), the costs languish before the advantages.

The results of the study here presented comes from the experience as both approaches have been considered in

the mortar implementation for the contact mechanics. The results obtained with the collocation method drive us take as final approach the exact integration method, and the results and developments present on this work come from this integration method. Similar results were obtained by *Farah*[ArtFPW14], where both methods were tested and the conclusions obtained were similar.

A.2.2.2 Solution study

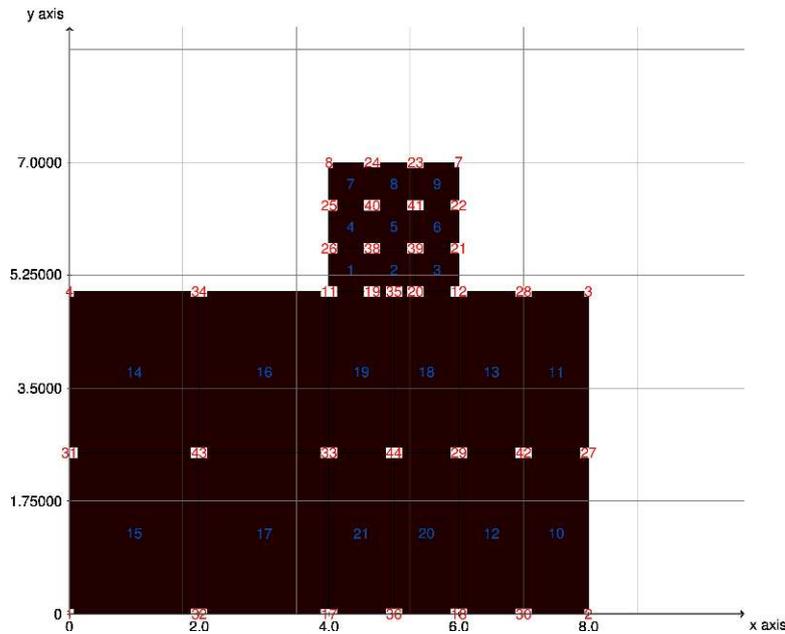
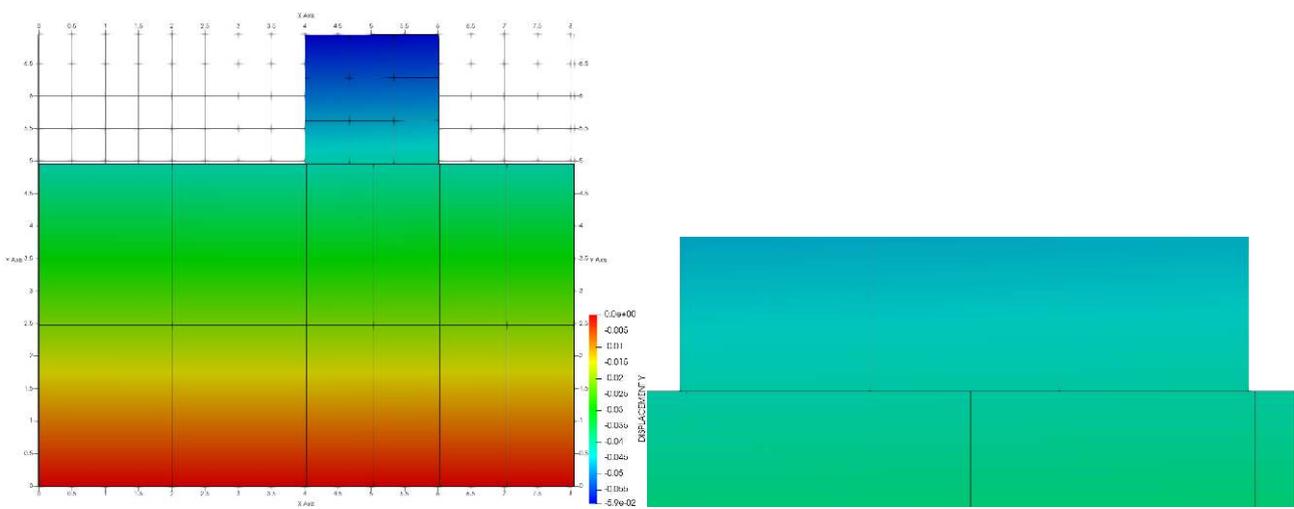


Figure A.3: Taylor patch test[ArtTP]

Here we study how the solution obtained with both methods, on *Farah*[ArtFPW14] work the comparison is done in a more complex setup, but we will show a simpler case which analytic solution, that even in that case does not fulfil the analytic solution. We will consider a classical problem which is used on benchmarking the contact formulation, this method was particularly extended for testing the **NTS** formulations. Figure A.3 shows this test, called Taylor patch test[ArtTP].



(a) Vertical displacement

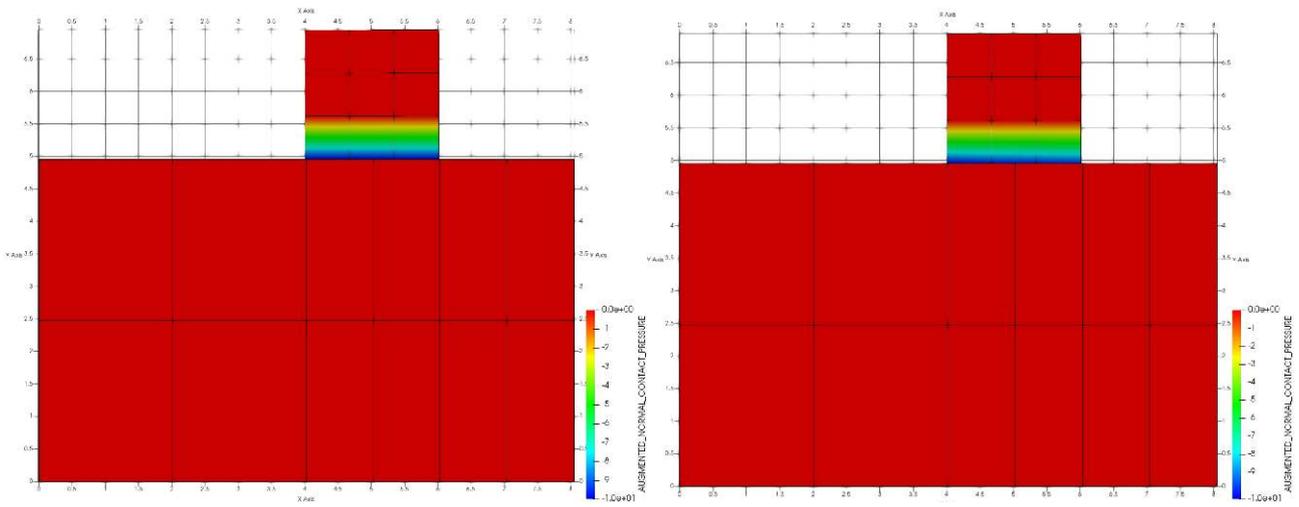
(b) Detail displacement on the interface

Figure A.4: Displacement on Taylor test

The problem consists on two solid blocks contacting on the interface, with a distributed load on the top block. The solution expected is a continuous field of displacements plus a constant contact pressure which coincides

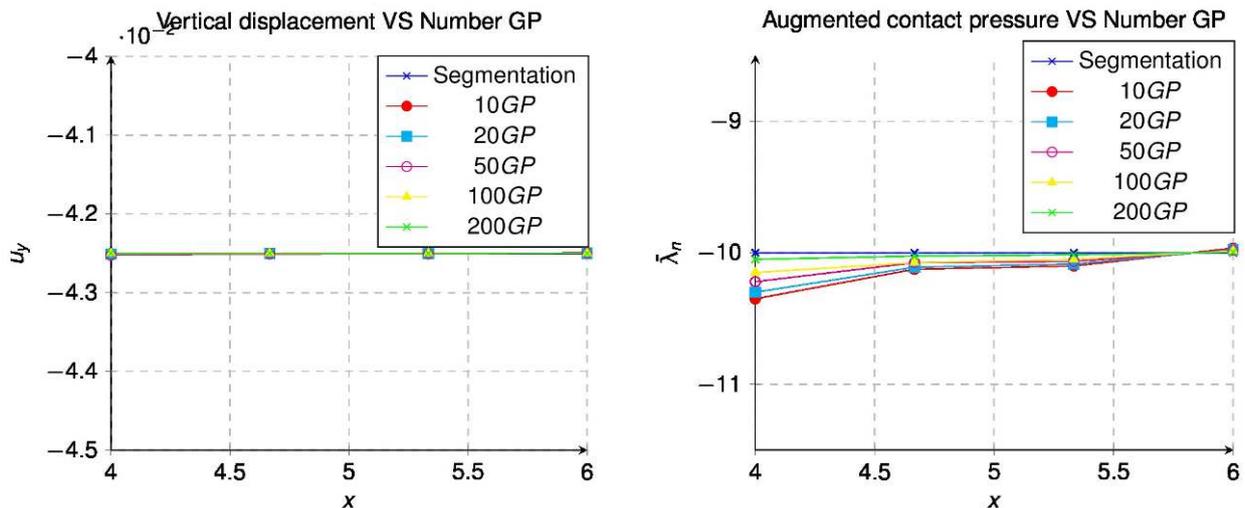
with the applied load in the top. We will consider for this problem $E = 1000$, $\nu = 0.4$ and the distributed load as $\rho = 10Pa$.

In our study we analyse the convergence of $\bar{\lambda}_n$ in the resolution of the problem with a different number of **GP** on the collocation. We do so, because $\bar{\lambda}_n$ takes longer to converge to the analytical solution. On the other hand, displacements converge even considering a low number of **GP**. *Farah*[ArtFPW14] also studies the convergence of $\bar{\lambda}_n$, but additionally checks the convergence of the energy error.



(a) Augmented contact pressure with mortar segmentation (b) Augmented contact pressure with mortar collocation with 200 **GP**
 Figure A.5: Compared solution for the augmented contact pressure

On Figure A.4 we can see the solution of the vertical displacements of the problem, particularly on the subfigure A.4a. Then in the subfigure A.4b the detail on the interface is appreciated, here we can see that in the deformed configuration the interface does not match any more. This loss of matching across of the interfaces is the reason why the collocation method has problems to converge.



(a) Vertical displacement convergence for different number of **GP** (b) Augmented contact pressure convergence for different number of **GP**
 Figure A.6: Convergence of the solution for different number of **GP**

The next Figure shows the solution for the augmented contact pressure, where Figure A.5a shows the exact solution, corresponding with $\bar{\lambda}_n = -10Pa$. The Figure A.5b presents the solution for 200 **GP** with the collocation method. Both solutions look very close, for more details we will address a more detailed plot next.

The last figure, Figure A.6, tries to represent with more detail the convergence of the solution for the vertical displacement and for λ_n . On the first one, Figure A.6a, it can be appreciated that the vertical displacement converges and gives the correct solution with a very small number of GP. On the other hand, Figure A.6b shows the convergence of the augmented contact pressure, and even for a large number of GP the exact solution is not obtained, despite of being close and getting closer with the grow of the number of GP.

A.2.3 Delaunay vs Convex polygon construct

On this section we will study the three different approaches, we can consider in order to build the integration triangles considered on the mortar segmentation from a cloud of points obtained during the clipping procedure. These methods are:

- The *Delaunay* triangulation[Onl]: The procedure (Figure A.7a) generates a set of triangles from a set of points. The method maximises the minimum angle of all the angles of the triangles in the triangulation. In the case of convex polygons (see next point) the number of triangles generated corresponds with $(n - 2)$, being n the number of vertices. The method must not be confused with **Constrained Delaunay Triangulation (CDT)**, which is an improved version of the method allowing to fix segments.
- The convex polygon construct: Also called *fan triangulation*, due to the resulting aspect of this method (Figure A.7b). This method it is only applicable in the case that the points configure a convex polygon, which means only in polygons in which no line segment between two points on the boundary ever goes outside the polygon[Onl]. The most important property of this kind of polygon is that every convex polygon admits a *fan triangulation* in $(n - 2)$ triangles¹.
- The third method is the so-called *centre-based* triangulation, see Figure A.7c. This method is discarded because generates n triangles from n vertices (then more expensive than the previous alternatives). The only advantage compared with the previous alternative is the fact that the aspect ratio of the generated triangles is, in general, better than the one obtained with the *fan triangulation*.

Comparing three methods, the *Delaunay* triangulation it is for sure the most general method, but algorithmically more complex to implement and significantly more expensive in computational cost. This method is in fact implemented as optional in *Kratos*, considering the robust and extensively use *triangle* library. On the other side, the *fan triangulation* is restricted to fewer cases (convex polygons), but the implementation is much simpler, easier and less error-prone. The third method, *centre-based* triangulation, despite of giving a better aspect ratio triangulation, has as the main disadvantage that, as previously mentioned, generates a higher number of triangles, being then from a computational point of view, more expensive.

It is important to highlight the fact that choose between these two methods is easier if we take into consideration that the geometries considered for the contact problems are linear all of them. This implies that the resulting segmentation (Figure A.1) will always provide us a convex polygon, which means **we can apply the fan triangulation method**. For this reason, because for a convex polygon the advantages outweigh the disadvantages.

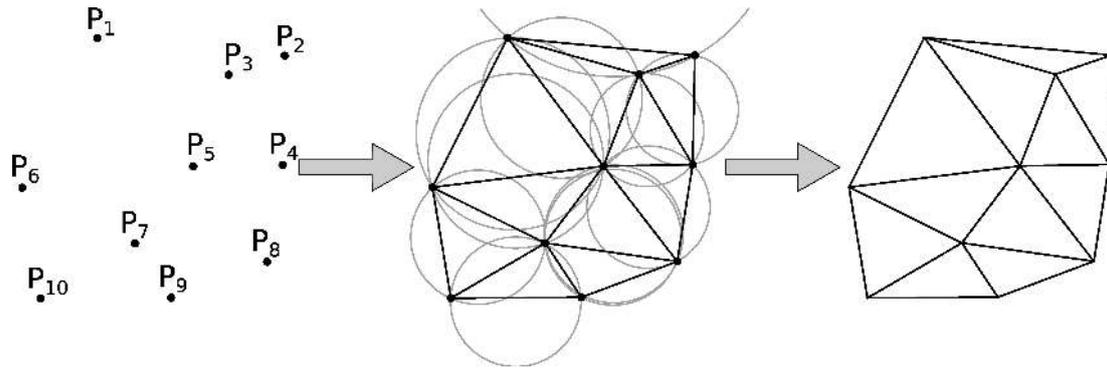
A.3 Mesh tying

A.3.1 Introduction

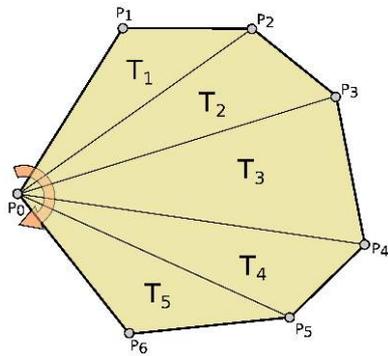
The following section presents the formulation employed in the derivation of the mesh tying condition formulation with dual *Lagrange multiplier*, based in the work of *Alexander Popp*[PhDPop12; ArtPop+10]. The aim of this section is to shortly introduce the mesh tying formulation, which can be considered as the base formulation for the contact formulation from 4.3.Formulation. This formulation is simpler and allows to understand and check how the mortar integration works.

The mesh tying problem is based on the **IBVP** of non-linear solid mechanics and the tied contact constraints. After recapitulating some basic notation and the strong formulation, a weak formulation of the mesh tying problem with two subdomains will be introduced. Here, only the interpretation as constrained minimisation problem is considered, leading to an indefinite saddle point formulation based on *Lagrange multipliers*.

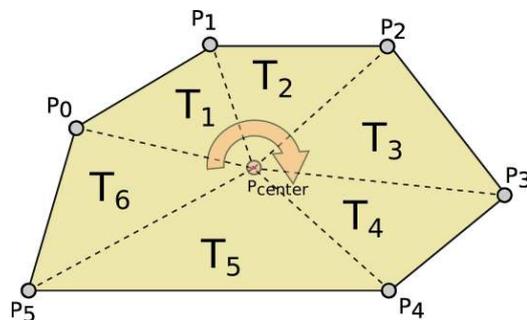
¹This is determined by the results obtained by *Catalan*, particularly in application of the *Catalan* number[ArtKG11]



(a) *Delaunay triangulation*



(b) *Fan triangulation. Inspired on[On]*



(c) *Center-based triangulation*

A.3.2 Strong formulation

On each subdomain Ω_0^i , the initial boundary value problem of finite deformation elastodynamics needs to be satisfied, this means to follow (A.2). The tied contact constraint, also formulated in the reference configuration, is given as (A.2f).

$$\begin{aligned}
 \text{(A.2a)} \quad & \nabla \cdot \mathbf{P}^i + \hat{\mathbf{b}}_0^i = \rho_0^i \ddot{\mathbf{u}}^i \text{ in } \Omega_0^i \times [0, T] \\
 \text{(A.2b)} \quad & \mathbf{u}^i = \hat{\mathbf{u}}^i \text{ on } \Gamma_u^i \times [0, T] \\
 \text{(A.2c)} \quad & \mathbf{P}^i \cdot \mathbf{N}^i = \hat{\mathbf{t}}_0^i \text{ on } \Gamma_\sigma^i \times [0, T] \\
 \text{(A.2d)} \quad & \mathbf{u}^i(\mathbf{X}^i, 0) = \hat{\mathbf{u}}_0^i(\mathbf{X}^i) \text{ in } \Omega_0^i \\
 \text{(A.2e)} \quad & \dot{\mathbf{u}}^i(\mathbf{X}^i, 0) = \hat{\dot{\mathbf{u}}}_0^i(\mathbf{X}^i) \text{ in } \Omega_0^i \\
 \text{(A.2f)} \quad & \mathbf{u}^1 = \mathbf{u}^2 \text{ on } \Gamma_c^i \times [0, T]
 \end{aligned}$$

In the course of deriving a weak formulation, the balance of linear momentum at the mesh tying interface Γ_c^i is typically exploited and a **LM** vector field λ is introduced, thus setting the basis for a mixed variational approach.

A.3.3 Weak formulation

To start the derivation of a weak formulation of (A.2), appropriate solution spaces \mathcal{U}^i and weighting spaces \mathcal{V}^i need to be defined as (A.3).

$$\text{(A.3)} \quad \begin{cases} \mathcal{U}^i = \{ \mathbf{u}^i \in H^1(\Omega) \mid \mathbf{u}^i = \hat{\mathbf{u}}^i \text{ on } \Gamma_u^i \}, \\ \mathcal{V}^i = \{ \delta \mathbf{u}^i \in H^1(\Omega) \mid \delta \mathbf{u}^i = \mathbf{0} \text{ on } \Gamma_u^i \} \end{cases}$$

Additionally the *Lagrange multiplier* vector $\boldsymbol{\lambda} = -\mathbf{t}_c^1$, which enforces the mesh tying constraint (A.2f), represents the negative slave side contact traction \mathbf{t}_c^1 , is chosen from a corresponding solution space denoted as \mathcal{M} . In terms of its classification in functional analysis, this space represents the dual space of the trace space \mathcal{W}^1 of \mathcal{V}^1 . In the given context, this means that $\mathcal{M} = H^{1/2}(\Gamma_c)$ and $\mathcal{L}^1 = H^{1/2}(\Gamma_c)$, where \mathcal{M} and \mathcal{L}^1 denote single scalar components of the corresponding vector-valued spaces \mathcal{M} and \mathcal{W} .

Based on these considerations, a saddle point type weak formulation is derived next. This can be done by extending the standard weak formulation of **NL** solid mechanics as defined to two subdomains and combining it with the **LM** coupling terms introduced in generic form. Find $\mathbf{u}^i \in \mathcal{U}^i$ and $\boldsymbol{\lambda} \in \mathcal{M}$ such that we obtain (A.4).

$$(A.4a) \quad -\delta\mathcal{L}_{kin}(\mathbf{u}^i, \delta\mathbf{u}^i) - \delta\mathcal{L}_{int,ext}(\mathbf{u}^i, \delta\mathbf{u}^i) - \delta\mathcal{L}_{mnt}(\boldsymbol{\lambda}^i, \delta\mathbf{u}^i) = 0 \quad \forall \delta\mathbf{u}^i \in \mathcal{V}$$

$$(A.4b) \quad -\delta\mathcal{L}_\lambda(\mathbf{u}^i, \delta\boldsymbol{\lambda}^i) = 0 \quad \forall \delta\boldsymbol{\lambda}^i \in \mathcal{M}$$

Herein, the kinetic contribution $\delta\mathcal{L}_{kin}$, the internal and external contributions $\delta\mathcal{L}_{int,ext}$ and the mesh tying interface contribution $\delta\mathcal{L}_{mnt}$ to the overall virtual work on the two subdomains, as well as the weak form of the mesh tying constraint $\delta\mathcal{L}_\lambda$, have been abbreviated as (A.5).

$$(A.5a) \quad -\delta\mathcal{L}_{kin} = \sum_{i=1}^2 \left[\int_{\Omega_0^i} \rho_0^i \ddot{\mathbf{u}}^i \cdot \delta\mathbf{u}^i dV_0 \right]$$

$$(A.5b) \quad -\delta\mathcal{L}_{int,ext} = \sum_{i=1}^2 \left[\int_{\Omega_0^i} (\mathbf{S}^i : \delta\mathbf{E}^i - \hat{\mathbf{b}} \cdot \delta\mathbf{u}^i) dV_0 - \int_{\Gamma_\sigma^i} \hat{\mathbf{t}}_0^i \cdot \delta\mathbf{u}^i dA_0 \right]$$

$$(A.5c) \quad -\delta\mathcal{L}_{mnt} = \sum_{i=1}^2 \left[\int_{\Gamma_c^i} \boldsymbol{\lambda} \cdot (\delta\mathbf{u}^1 - \delta\mathbf{u}^2) dA_0 \right]$$

$$(A.5d) \quad -\delta\mathcal{L}_\lambda = \sum_{i=1}^2 \left[\int_{\Gamma_c^i} \delta\boldsymbol{\lambda} \cdot (\mathbf{u}^1 - \mathbf{u}^2) dA_0 \right]$$

The coupling terms on Γ_c also allow for a direct interpretation in terms of variational formulations and the principle of virtual work. Whereas the contribution in (A.5c) represents the virtual work of the unknown interface traction $\boldsymbol{\lambda} = \mathbf{t}_c^1 = \mathbf{t}_c^2$, the contribution in (A.5d) ensures a weak, variational consistent enforcement of the tied contact constraint (A.2f). Nevertheless, the concrete choice of the discrete **LM** space \mathcal{M}_h in the context of mortar finite element discretisations is decisive for the stability of the method and for optimal a priori error bounds. Finally, it is pointed out that the weak formulation (A.4a) and (A.4b) possesses all characteristics of saddle point problems and *Lagrange multiplier* methods.

A.3.4 Discretisation and numerical integration

The part relative to the dual **LM** it is detailed in the respective section 4.3.3.4.1. *Dual Lagrange multipliers*, and the *Mortar operators* from 4.3.3.4.2. *Mortar operators*. This development is identical for the mesh tying formulation, and will not be repeated.

Finally, once computed the mortar operators, the resulting system for mesh tying corresponds with (A.6). This can be statically condensed, as *Alexander Popp* shows in [PhDPop12]. With this the **LM** can be removed of the system thanks to the diagonal property of **D** when considering the dual **LM**, making a pure displacement formulation (in case the tied **DOF** corresponds with the displacements).

$$(A.6) \quad \begin{bmatrix} \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{N}\mathcal{M}} & \mathbf{K}_{\mathcal{N}\mathcal{S}} & \mathbf{0} \\ \mathbf{K}_{\mathcal{N}\mathcal{N}} & \mathbf{K}_{\mathcal{M}\mathcal{M}} & \mathbf{0} & -\mathbf{M}^T \\ \mathbf{K}_{\mathcal{S}\mathcal{N}} & \mathbf{0} & \mathbf{K}_{\mathcal{S}\mathcal{S}} & \mathbf{D}^T \\ \mathbf{0} & -\mathbf{M} & \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{d}_{\mathcal{N}} \\ \Delta\mathbf{d}_{\mathcal{M}} \\ \Delta\mathbf{d}_{\mathcal{S}} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_{\mathcal{N}} \\ \mathbf{r}_{\mathcal{M}} \\ \mathbf{r}_{\mathcal{S}} \\ \mathbf{r}_{\boldsymbol{\lambda}} \end{bmatrix}$$

A.3.5 Numerical example

E Solid 1	ν Solid 1	E Solid 2	ν Solid 2
$2 \cdot 10^8 Pa$	0.35	$2 \cdot 10^8 Pa$	0.35

Table A.1: Parameters considered for the mesh tying example

The following example, with the mesh presented on Figure A.7, consists in a L shaped solid, where we consider as a different body the circular section from the angled corner. The mesh interface between the solids does not match. The problem is formulated as solid formulated on **TL** framework, considering *Neo-Hookean* hyperelastic material with the properties from the Table A.1. We apply in the upper corner a vertical displacement equal to t , with t going from $[0, 2]s$, the lower corner is fixed.

The solution obtained is shown in Figure A.8, it can be seen that the continuity in the interface is preserved, despite the large displacement solution. From a practical point of view is like considering a continuous element of the whole domain. This is basically the type of formulation originally considered in the original *Mortar* for **DDM**[BookWoh01; BookTos05].

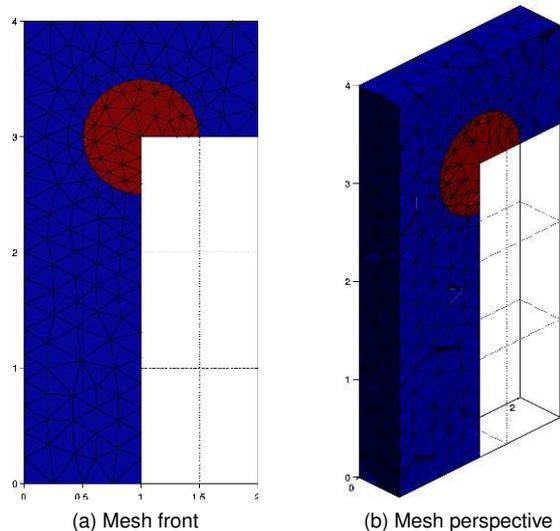


Figure A.7: Mesh tying example

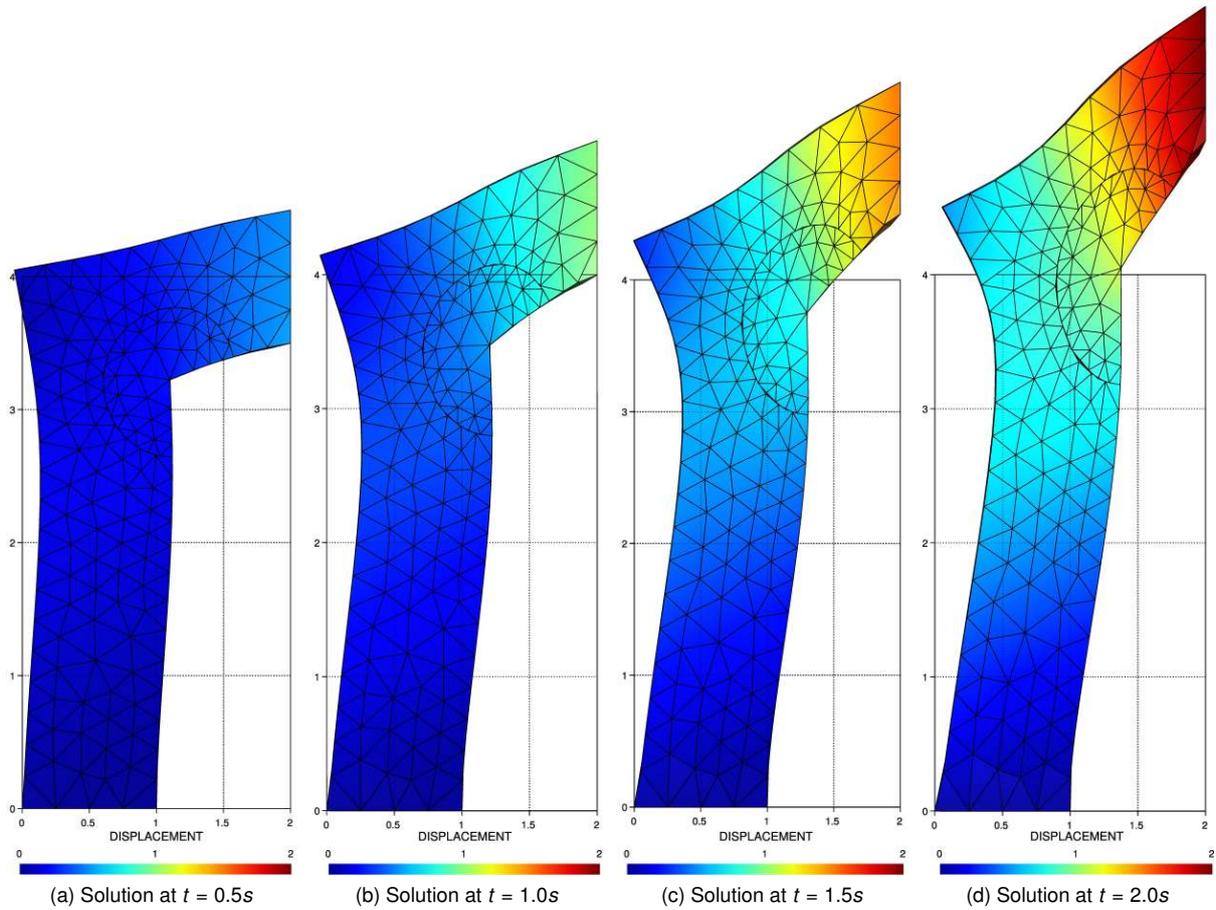


Figure A.8: Solution for mesh tying example

Bibliography

Books

- [BookBel+14] *Nonlinear Finite Elements for Continua and Structures*. Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. Wiley. 2nd ed. 2014.
- [BookTos05] *Domain Decomposition Methods — Algorithms and Theory*. Andrea Toselli Olof B. Widlund (auth.) Springer-Verlag Berlin Heidelberg. 1st ed. 2005.
- [BookWoh01] *Discretization Methods and Iterative Solvers Based on Domain Decomposition*. Barbara I. Wohlmuth. Springer-Verlag Berlin Heidelberg. 1st ed. 2001.

Articles

- [ArtAur+10] “Isogeometric collocation methods”. F Auricchio, L Beirao Da Veiga, TJR Hughes, A_ Reali, and G Sangalli. In: *Mathematical Models and Methods in Applied Sciences*. No. 11, Vol. 20, 2010, pp. 2075–2107. World Scientific. DOI: [10.1142/s0218202510004878](https://doi.org/10.1142/s0218202510004878).
- [ArtFPW14] “Segment-based vs. element-based integration for mortar methods in computational contact mechanics”. Philipp Farah, Alexander Popp, and Wolfgang A. Wall. In: *Computational Mechanics*. No. 1, Vol. 55, 2014, pp. 209–228. DOI: [10.1007/s00466-014-1093-2](https://doi.org/10.1007/s00466-014-1093-2).
- [ArtKG11] “Some divisibility properties of Catalan numbers”. Thomas Koshy and Zhenguang Gao. In: *The Mathematical Gazette*. Vol. 95, 2011, pp. 96–102. DOI: [10.1017/s002555720000245x](https://doi.org/10.1017/s002555720000245x).
- [ArtPop+10] “A dual mortar approach for 3D finite deformation contact with consistent linearization”. Alexander Popp, Markus Gitterle, Michael W. Gee, and Wolfgang A. Wall. In: *International Journal for Numerical Methods in Engineering*. No. 11, Vol. 83, 2010, pp. 1428–1465. John Wiley & Sons, Ltd. DOI: [10.1002/nme.2866](https://doi.org/10.1002/nme.2866).
- [ArtSch+13] “Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations”. Dominik Schillinger, John A Evans, Alessandro Reali, Michael A Scott, and Thomas JR Hughes. In: *Computer Methods in Applied Mechanics and Engineering*. Vol. 267, 2013, pp. 170–232. Elsevier. DOI: [10.1002/pamm.201310049](https://doi.org/10.1002/pamm.201310049).
- [ArtTP] “On a patch test for contact problems in two dimensions”. Robert L. Taylor and Panagiotis Panagiotopoulos. In: . ,

Ph.D.’s thesis

- [PhDPop12] “Mortar Methods for Computational Contact Mechanics and General Interface Problems”. Alexander Popp. 2012.

Online resources

- [Onl] *Wikipedia. Convex polygon*. URL: https://en.wikipedia.org/wiki/Convex_polygon.

- [Onl] *Wikipedia. Delaunay triangulation.* URL:
https://en.wikipedia.org/wiki/Delaunay_triangulation.
- [Onl] *Wikipedia. Fan triangulation.* URL:
https://en.wikipedia.org/wiki/Fan_triangulation.

Appendix B

Implementation

“If I have seen further, it is by standing on the shoulders of Giants.”

*Isaac Newton
(1643 - 1727 AD, English
mathematician, physicist,
astronomer, theologian)*

B.1 Introduction

The following appendix introduces the main codes used on the developments of this work. The first one is the multiphysics framework *Kratos* (B.2.Kratos Multiphysics), where most of the developments were done. The next section is the remeshing library denominated *Mmg* (B.3.Mmg library) considered on the adaptive remeshing chapter (6.Adaptative remeshing).

B.2 Kratos Multiphysics

B.2.1 Introduction



Figure B.1: *Kratos* Multiphysics logo. Image from [Kratos web](#)

The implementation shown in this research has been done on the *Kratos* multi-physics framework. *Kratos*[PhDDad07; ArtDRO10] is a framework for building multidisciplinary finite element programs. We can summarise the following features:

- **Kernel:** The kernel and application approach is used to reduce the possible conflicts arising between developers of different fields.
- **Object Oriented (OO):** The modular design, hierarchy and abstraction of these approaches fit to the generality, flexibility and reusability required for the current and future challenges in numerical methods. Summarised as **Multi-Layer Design Approach (MLDA)**. The main code is developed in *C++* and the *Python* language is used for scripting. The *C++* practises are focus on modern guidelines[BookAle01; BookMey14]

- **Open source:** The **Berkeley Software Distribution (BSD)** licence allows to use and distribute the existing code without any restriction, but with the possibility to develop new parts of the code on an open or close basis depending on the developers. Additionally **Kratos** can be obtained for free.
- **HPC:** **Kratos** is build-in as a powerful multiphysics **HPC**[ArtDad+13]. **Kratos** is based on **Trilinos MPI** and **OpenMP**.

B.2.2 Framework (API)

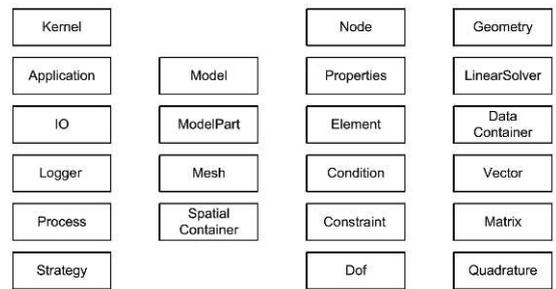
B.2.3 Introduction. Main classes

The **OO** design from Figure B.2a provides different classes necessary for a **FE** simulation which ensure the reusability and extensibility of the code. Here, basic numerical concepts are represented by classes like `Vector`, `Matrix` and `Quadrature`. `Node`, `Element`, `Condition`, and `DoF` are defined directly from **FE** concepts. `Model`, `Mesh` and `Properties` are from the practical methodology used in **FE** modelling complemented by `ModelPart`, and `SpatialContainer`, for organising better all data necessary for analysis. `IO`, `LinearSolver`, `Process`, and `Strategy` represent the different steps of a **FE** program flow. Finally, `Kernel` and `Application` are defined for library management and its interface definition.

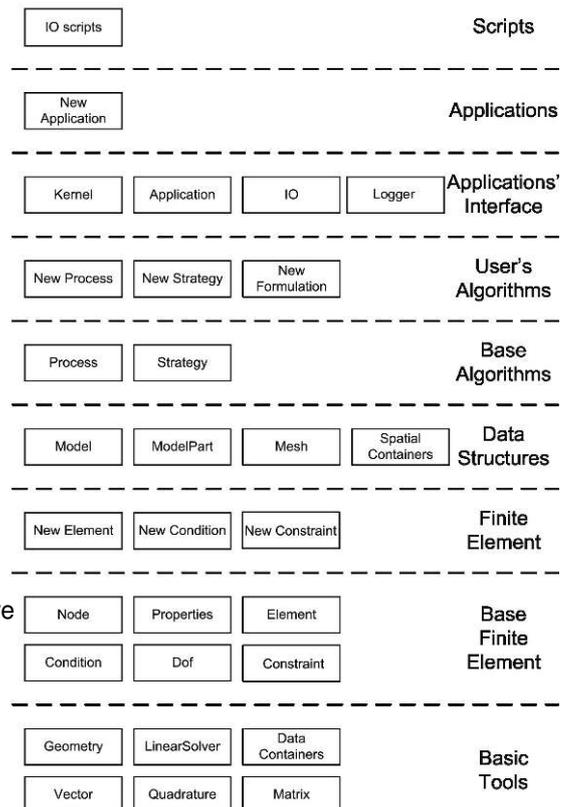
In the **MLDA** each object only interfaces with other objects in its layer or in layers below its layer (Figure B.2b). This approach is used to reduce the dependency of different parts which is beneficial for the reduction of conflicts between different parts of the code[ArtDRO10].

The **Applications** considered and developed for this work are the following:

- **Core:** Where the main classes are declared, Figure B.2a. It also includes several algorithms and utilities common to all the **Applications**, like the **NR** or the `BuilderAndSolver`.
- **StructuralMechanicsApplication:** Where the elements, conditions and **CL** are implemented. Also other utilities and strategies are implemented here. The elements from 3. **Rotation-free shells and solid-shell elements** and the **CL** from 5. **Plasticity** can be found here.
- **ContactStructuralMechanicsApplication:** Where the conditions, strategies and algorithms presented on 4. **Contact mechanics** can be found.
- **MeshingApplication:** All the developments relative with **Mmg** and the metric and remeshing techniques¹ from 6. **Adaptative remeshing**. The details of the **Mmg** integration can be found in B.3.2. **Integration between Mmg and Kratos**.



(a) **Kratos** classes



(b) **Kratos MLDA**

Figure B.2: Source[ArtDRO10], updated with the recent classes added

The following sections will summarise the main classes existing in **Kratos**.

¹The **SPR** error algorithms are located in the `StructuralMechanicsApplication` and `ContactStructuralMechanicsApplication`.

B.2.4 Model

B.2.4.1 Introduction

On Figure B.3 an example of the data structure of the `Model`. The `Model` stores the whole model to be analysed, and manages the different `ModelPart` used in the simulation.

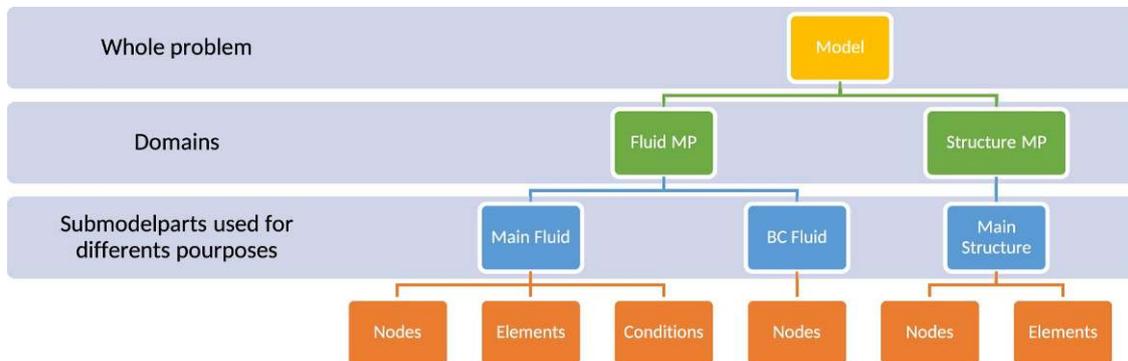


Figure B.3: Model data structure

B.2.4.2 ModelPart

The `ModelPart` holds all data related to an arbitrary part of model. It stores all existing components and data like `Nodes`, `Properties`, `Elements`, `MasterSlaveConstraint`, `Conditions` and solution data related to a part of the `Model`. The entities stored on the `ModelPart` are:

- `Node` It is a point with additional facilities. Stores the nodal data, historical nodal data, and list of *DoF*.
- `Condition` encapsulates data and operations necessary for calculating the local contributions of `Condition` to the global system of equations.
- `Element` encapsulates the elemental formulation in one object and provides an interface for calculating the local matrices and vectors necessary for assembling the global system of equations. It holds its geometry that meanwhile is its array of `Nodes`.
- `MasterSlaveConstraint` encapsulates the **MPC** that are used in order to impose relationships between **DOF**.
- `Properties` encapsulates data shared by different `Elements` or `Conditions`. It stores any type of data.

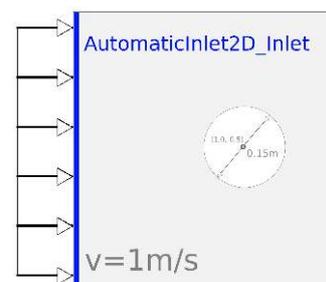
B.2.4.3 Submodelparts

In our implementations we use processes (Figure B.4a) to set the **BC** (both *Neumann* or *Dirichlet*), as idealised on Figure B.4b. In order to be able to assign this **BC** we need to subdivide the `ModelPart` into different pieces that we can call submodelparts.

```

"python_module" : "apply_inlet_process",
"kratos_module" : "KratosMultiphysics.FluidDynamicsApplication",
"help"          : [],
"process_name"  : "ApplyInletProcess",
"Parameters"   : {
  "model_part_name" : "AutomaticInlet2D_Inlet",
  "variable_name"   : "VELOCITY",
  "modulus"         : 1.0,
  "direction"       : "automatic_inwards_normal",
  "interval"        : [0, "End"]
}
    
```

(a) Example of **BC** in *json* format



(b) Submodelpart **BC**

Figure B.4: Concept of submodelpart for **BC**

B.2.5 Base classes

In here we detail some relevant classes which define the basic behaviour of *Kratos*. The classes used in the **FEM** are already defined in [B.2.4.2.ModelPart](#). Apart of these classes, we have another relevant base class, as the base `Process`, which defines the common interface of the processes. The `Process` is a class which evaluates different types of algorithms, like searches, interpolate values, etc.

B.2.6 Geometry

The `Geometry` is the base class which defines all the common methods of interest in a geometry, such as volume, area, shape functions, etc. The standard geometries store an array of nodes or points.

B.2.7 Strategy

The strategies are responsible on solving the system of equations. There are 4 types of strategies:

- `BuilderAndSolver`: This is the base class which defines the necessary methods to assemble the system of equations. In order to do so it asks the `Element`, `Condition` and `MasterSlaveConstraint` for its local contribution and assemble them in a global system depending of the `DoF` order.
- `Scheme`: This class provides the implementation of the basic tasks that are needed by the solution strategy in order to integrate the solution over the time. Check algorithms on [2.4.4.Time integration schemes](#).
- `ConvergenceCriteria`: This is the base class to define the different convergence criterion considered. Here the residual/solution increment or any other criteria is taken into account in order to check if converged.
- `SolvingStrategy`: This is the base class from which we will derive all the strategies (line-search, **NR**, etc.). See [2.4.3.Solution of the non-linear-equilibrium equations system](#). It is the class responsible of actually solving the problem. It integrates all the former classes.

B.2.8 IO classes

In this class we can include the classes dedicated to read external files, such as the `*.mdpa` files, default input for *Kratos*. We can include the utilities used to read `*.json` files. These are the default file types used to define the parameters of the problem and the material `Properties`. Additionally, in the **IO** category, we can include classes such the `Logger` which is used to show/register/save the simulated information.

B.2.9 Testing framework

The testing framework is very relevant in *Kratos*, and have allowed a faster in *Kratos* than the former to the introduction of this technology. `unittest` are extensibility used on **Information and Communications Technology (ICT)** to ensure code quality. An `unittest` is a software testing method by which individual units of the source code, hence the name comes, are tested to determine whether they behave as intended. We can differentiate in *Kratos* two different approaches for the `unittest`:

- **Python tests**: These tests are based in the [Python unittest framework](#), which was originally inspired by `JUnit`. With this framework we can test and check methods exported to `Python` and run full short problems in order to check that the results obtained are the expected.
- **C++ tests**: This is a custom *Kratos* testing framework where the test is fully written in `C++`.

B.3 Mmg library

B.3.1 What is Mmg and how does it work?

Mmg is an open source software for anisotropic automatic *remeshing for unstructured meshes* based on *Delaunay* triangulation. It is licenced under a [GNU Lesser General Public License \(LGPL\)](#) licence and it has been integrated

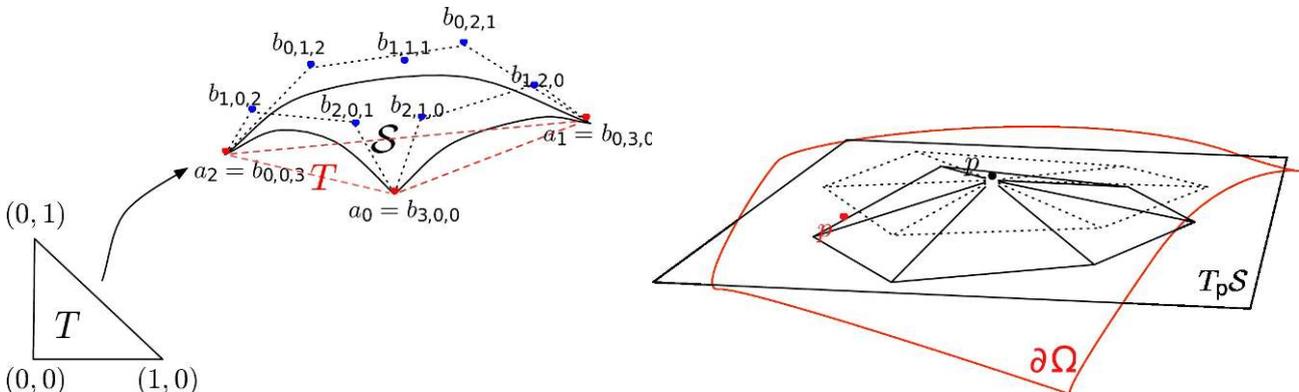


Figure B.5: *Mmg* logo. Image from [Mmg web](#)

in *Kratos*[ArtDRO10] via the `mmg_process.h` in the `MeshingApplication`. It provides 3 applications and 4 libraries:

- The `mmg2d` application and the `libmmg2d` library: adaptation and optimisation of a two-dimensional triangulation and generation of a triangulation from a set of points or from given boundary edges.
- The `mmgs` application and the `libmmgs` library: adaptation and optimisation of a surface triangulation and isovalue discretisation.
- The `mmg3d` application and the `libmmg3d` library: adaptation and optimisation of a tetrahedral mesh and implicit domain meshing.
- The `libmmg` library gathering the `libmmg2d`, `libmmgs` and `libmmg3d` libraries.

The *Mmg* remeshing process modifies the mesh[BookDob12][ArtDDF14] iteratively until be in agreement with the prescribed sizes on the idealised (Figure B.6) contour (and directions in case of anisotropic mesh). The software reads the mesh and the metric, then the mesh is modified using local mesh modifications of which an intersubsection procedure based on anisotropic *Delaunay* kernel.



(a) A piece of parametric Bézier cubic surface associated to triangle T (b) The resulting configuration of the vertex relocation procedure
 Figure B.6: *Mmg* idealised geometry. Source[ArtDDF14]

We can resume the algorithm in following steps:

1. *Mmg* tries to have a good approximation of the surface (with respect to the *Hausdorff* parameter).
2. It remeshes on geometric criteria. *Mmg* scans the surface tetrahedra and splits the tetrahedra using predefined patterns if the *Hausdorff* distance[BookRW09] between the surface triangle of the tetra and its curve representation doesn't respect the *Hausdorff* parameter.
3. The library scans again the surface tetrahedra and collapses all the edges at a *Hausdorff* distance smaller than a threshold defined in function of the *Hausdorff* parameter.
4. Next it intersects the provided metric and a surface metric computed at each point from the *Hausdorff* parameter and the curvature tensor at the point.

5. Then *Mmg* smooths the metric to respect the gradation parameter. The metrics are iteratively propagated until the respect of the gradation everywhere.
6. Following it remeshes the surface tetrahedra in order to respect the new metric.
7. Finally, it remeshes both the volume and surface to have edges between 0.6 and 1.3 (in the metric). The long edges are cut and short ones are deleted (collapsed).

B.3.2 Integration between *Mmg* and *Kratos*

B.3.2.1 Introduction

In order to understand the integration between *Kratos* and *Mmg* is important to understand the data structure of *Kratos*. We address in first place to the corresponding section about `Model` structure on the *Kratos* API (B.2.4.Model).

B.3.2.2 Class structure

The implementation of *Mmg* into *Kratos* is done in the `MeshingApplication`. The **Unified Modeling Language (UML)** diagram presented on Figure B.7 shows how the implementation is done.

In here we see like there is an auxiliary class called `MMGUtilities`, which is called by the `MMGProcess`, which is the process called for remeshing and `MMGIO` which can be used to read write *Mmg* IO files.

`MMGUtilities` is the class responsible to interact with the *Mmg* API. Due to structure in *Mmg* library, a template argument must be considered in order to call the respective methods for each one of the specialised includes (`libmmg2d.h`, `libmmg3d.h` and `libmmsg.h`).

Additionally the `MMGProcess` integrates all the steps during the remesh process, including the value interpolation and the internal variables recovery from 6.6. **Internal values interpolation**. After its execution all the solvers and the processes must be initialised.

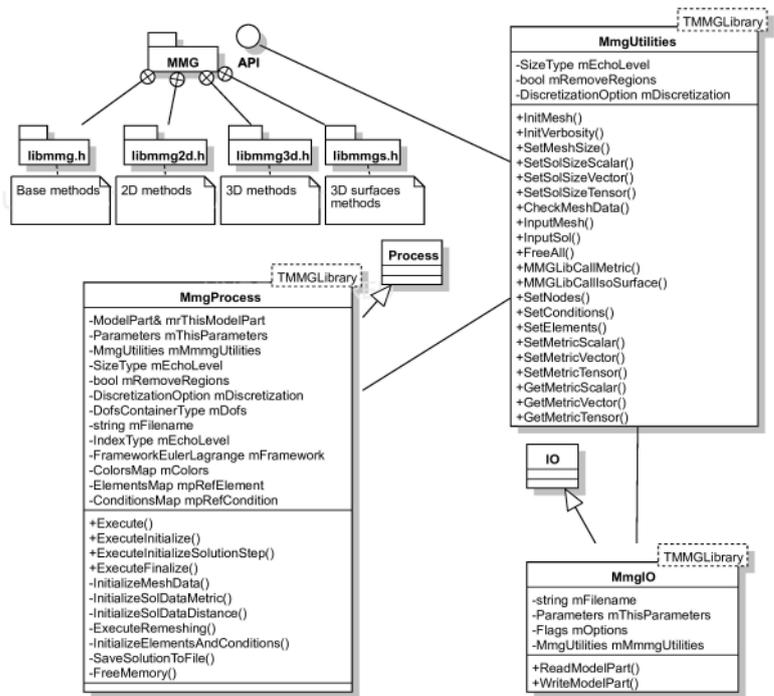


Figure B.7: *Mmg-Kratos* integration

B.3.2.3 Submodelpart recovery

As we have previously mentioned, on B.2.4.3. **Submodelparts** we need to subdivide the mesh into submodelparts so we can apply **BC**. In order to preserve that information after remeshing we need to create an identification system, so we are able to create a unique *ID* that will allow us to reconstruct the submodelpart structure after remeshing, this methodology is commonly called *colour identification*. This is thanks to the **API** of *Mmg* which allows to preserve a reference number, which works as an *ID*. In Figure B.8 we see the concept of this idea.

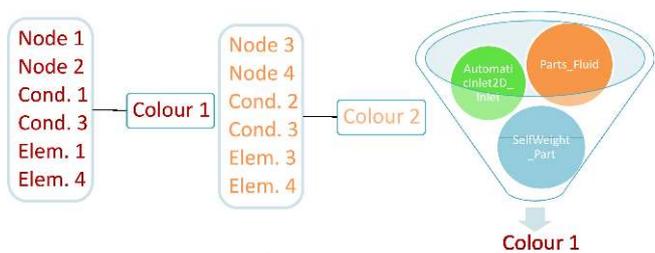


Figure B.8: Concept of colours

In *Kratos* this is implemented in the `AssignUniqueModelPartCollectionTagUtility` class, that can be found [here](#). This class computes a map of collections of submodelparts, in order to do that, a tag is assigned to each node, condition and element in order to get the collections where it belongs. The main `ModelPart` tag is 0, then each submodelpart has 1, 2... Finally, a combination of these submodelparts has another tag, greater than the former ones.

The class make use of `C++ std::unordered_map`. First the list of tags of each `ModelPart` is created, with the corresponding entities belonging to it. Then with this list map, the submodelparts combination tags are computed. Finally, with this, the list containing the tags of each submodelparts combination and the entities contained in it. The following block shows the way it works:

```

1  void AssignUniqueModelPartCollectionTagUtility::ComputeTags(
2  IndexIndexMapType& rNodeTags,
3  IndexIndexMapType& rCondTags,
4  IndexIndexMapType& rElemTags,
5  IndexStringMapType& rCollections
6  )
7  {
8  // Initialize and create the auxiliary maps
9  IndexIndexSetMapType aux_node_tags, aux_cond_tags, aux_elem_tags;
10
11 // We compute the list of submodelparts and subsubmodelparts
12 const StringVectorType& r_model_part_names = GetRecursiveSubModelPartNames(mrModelPart);
13
14 // Initialize the collections
15 IndexType tag = 0;
16 for (IndexType i_sub_model_part = 0; i_sub_model_part < r_model_part_names.size(); ++i_sub_model_part) {
17     rCollections[i_sub_model_part].push_back(r_model_part_names[i_sub_model_part]);
18
19     if (tag > 0) {
20         ModelPart& r_sub_model_part = GetRecursiveSubModelPart(mrModelPart, r_model_part_names[i_sub_model_part]);
21
22         /* Nodes */
23         NodesArrayType& r_nodes_array = r_sub_model_part.Nodes();
24         const auto it_node_begin = r_nodes_array.begin();
25         for (IndexType i_node = 0; i_node < r_nodes_array.size(); ++i_node)
26             aux_node_tags[(it_node_begin + i_node)->Id()].insert(tag);
27
28         /* Conditions */
29         ConditionsArrayType& r_conditions_array = r_sub_model_part.Conditions();
30         const auto it_cond_begin = r_conditions_array.begin();
31         for (IndexType i_cond = 0; i_cond < r_conditions_array.size(); ++i_cond)
32             aux_cond_tags[(it_cond_begin + i_cond)->Id()].insert(tag);
33
34         /* Elements */
35         ElementsArrayType& r_elements_array = r_sub_model_part.Elements();
36         const auto it_elem_begin = r_elements_array.begin();
37         for (IndexType i_elem = 0; i_elem < r_elements_array.size(); ++i_elem)
38             aux_elem_tags[(it_elem_begin + i_elem)->Id()].insert(tag);
39     }
40
41     ++tag;
42 }
43
44 // Now detect all the cases in which a node or a cond belongs to more than one part simultaneously
45 std::unordered_map<std::set<IndexType>, IndexType, KeyHasherRange<std::set<IndexType>, KeyComparatorRange<std::set<IndexType>>> combinations;
46
47 /* Nodes */
48 for(auto& r_aux_node_tag : aux_node_tags) {
49     const std::set<IndexType>& r_value = r_aux_node_tag.second;
50     if (r_value.size() > 1) combinations[r_value] = 0;
51 }
52
53 /* Conditions */
54 for(auto& r_aux_cond_tag : aux_cond_tags) {
55     const std::set<IndexType>& r_value = r_aux_cond_tag.second;
56     if (r_value.size() > 1) combinations[r_value] = 0;
57 }
58
59 /* Elements */
60 for(auto& r_aux_elem_tag : aux_elem_tags) {
61     const std::set<IndexType>& r_value = r_aux_elem_tag.second;
62     if (r_value.size() > 1) combinations[r_value] = 0;
63 }
64
65 /* Combinations */
66 for(auto& combination : combinations) {
67     const std::set<IndexType>& r_key_set = combination.first;

```

```

68     for(IndexType it : r_key_set)
69         rCollections[tag].push_back(rCollections[it][0]);
70     combinations[r_key_set] = tag;
71     ++tag;
72 }
73
74 // The final maps are created
75 /* Nodes */
76 for(auto& r_aux_node_tag : aux_node_tags) {
77     const IndexType key = r_aux_node_tag.first;
78     const std::set<IndexType>& r_value = r_aux_node_tag.second;
79
80     if (r_value.size() == 0)
81         rNodeTags[key] = 0; // Main Model Part
82     else if (r_value.size() == 1) // A Sub Model Part
83         rNodeTags[key] = *r_value.begin();
84     else // There is a combination
85         rNodeTags[key] = combinations[r_value];
86 }
87
88 /* Conditions */
89 for(auto& r_aux_cond_tag : aux_cond_tags) {
90     const IndexType key = r_aux_cond_tag.first;
91     const std::set<IndexType>& r_value = r_aux_cond_tag.second;
92
93     if (r_value.size() == 0)
94         rCondTags[key] = 0; // Main Model Part
95     else if (r_value.size() == 1) // A Sub Model Part
96         rCondTags[key] = *r_value.begin();
97     else // There is a combination
98         rCondTags[key] = combinations[r_value];
99 }
100
101 /* Elements */
102 for(auto& r_aux_elem_tag : aux_elem_tags) {
103     const IndexType key = r_aux_elem_tag.first;
104     const std::set<IndexType>& r_value = r_aux_elem_tag.second;
105
106     if (r_value.size() == 0)
107         rElemTags[key] = 0; // Main Model Part
108     else if (r_value.size() == 1) // A Sub Model Part
109         rElemTags[key] = *r_value.begin();
110     else // There is a combination
111         rElemTags[key] = combinations[r_value];
112 }
113
114 // Clean up the collections
115 for (auto& r_collection : rCollections) {
116     std::unordered_set<std::string> aux_set;
117     for (auto& r_name : r_collection.second) {
118         aux_set.insert(r_name);
119     }
120     std::vector<std::string> aux_vector;
121     for (auto& r_name : aux_set) {
122         aux_vector.push_back(r_name);
123     }
124     r_collection.second = aux_vector;
125 }
126 }

```

Bibliography

Books

- [BookAle01] *Modern C++ design: generic programming and design patterns applied*. Andrei Alexandrescu. Addison-Wesley. 2001.
- [BookDob12] *MMG3D: user guide*. Cécile Dobrzynski. 2012.
- [BookMey14] *Effective modern C++: 42 specific ways to improve your use of C++ 11 and C++ 14*. Scott Meyers. O'Reilly Media, Inc. 2014.
- [BookRW09] *Variational analysis*. R Tyrrell Rockafellar and Roger J-B Wets. Springer Science & Business Media. 2009.

Articles

- [ArtDad+13] “Migration of a generic multi-physics framework to HPC environments”. Pooyan Dadvand, Riccardo Rossi, Marisa Gil, Xavier Martorell, J Cotela, Edgar Juanpere, Sergio R Idelsohn, and Eugenio Oñate. In: *Computers & Fluids*. Vol. 80, 2013, pp. 301–309. Elsevier.
- [ArtDDF14] “Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems”. C. Dapogny, C. Dobrzynski, and P. Frey. In: *Journal of Computational Physics*. Vol. 262, 2014, pp. 358–378. DOI: [10.1016/j.jcp.2014.01.005](https://doi.org/10.1016/j.jcp.2014.01.005).
- [ArtDRO10] “An object-oriented environment for developing finite element codes for multi-disciplinary applications”. Pooyan Dadvand, Riccardo Rossi, and Eugenio Oñate. In: *Archives of computational methods in engineering*. No. 3, Vol. 17, 2010, pp. 253–297. Springer.

Ph.D.'s thesis

- [PhDDad07] “A framework for developing finite element codes for multi-disciplinary applications.” Pooyan Dadvand. Universitat Politècnica de Catalunya. 2007.

Appendix C

Automatic differentiation

“God does not care about our mathematical difficulties - he integrates empirically.”

Albert Einstein
(1879 - 1955 AD, German-born theoretical physicist)

C.1 Introduction

The **AD**[ArtGru82; ArtKor97; ArtKor02; BookWri08], also called **algorithmic differentiation** or **computational differentiation**. It consists in a set of techniques to numerically evaluate the derivative of a function specified by a computer program. This is possible to the main principle which defines the derivative operation, which is the differentiation of an algorithm by the use of the chain rule. This operation is completely linear, so it can be easily adapted to any problem, nevertheless of the complexity. As said Korelc[BookJo16] says, the main advantages of the **AD** are:

- Automatic generated codes are highly efficient if the software employed does the corresponding optimisations properly.
- The symbolic formulation is more compressed, for example we only work on the definition of the energy potential, and thus gives fewer possibilities of an error.
- The code generated can be adjusted to specific problems, leading to on-demand numerical code generation.
- The code can be generated for different environments and multi-languages from the same symbolic description.
- Some complex and error-prone operations, such algebraic operations are done automatically.
- Simplifies the implementation of multiphysics problems, as the symbolic definitions are purely mathematical and not related to any specific problem.

Of course the methodology is not free of controversy or drawbacks, the quantity of code generated can be enormous and less general compared with the manual counterpart, this implies bigger source code files and higher compilation and linking times.

In order to compute the **AD** we need a **Symbolic and Algebraic Computational Systems (SAC)** for the manipulation of mathematical expressions in symbolic form. Some know commercial **SAC** systems are *Mathematica* or *Maple*. From the **Open Source Software (OSS)** side, we can find the open-source symbolic *Python* based package,

Sympy[Onl]. **SAC** systems cannot be used in directly to define the solution of the problem, as in the case of complex engineering problems, the uncontrollable growth of expressions and consequently redundant operations leads to inefficient codes.

In addition we can mention the **Hybrid Object-Oriented (HOO)**[BookJo16] approach, an intermediate solution. This approach has brought a new perspective for the development of complex software and several **OO FE** environments have been developed. The **HOO** systems are in general restricted to a particular type of formulations where the strong form to the element equations can be defined. With this the expression growth problem mentioned for **SAC** is reduced, since the symbolic code derivation is used on the resolution of simpler sub-problems. The most relevant examples of this approach can be found on the **OSS** environments of *FreeFem++* (Hecht[ArtHec+09]) and *FEniCS* (Logg[ArtLog07]).

C.2 Mathematical concepts

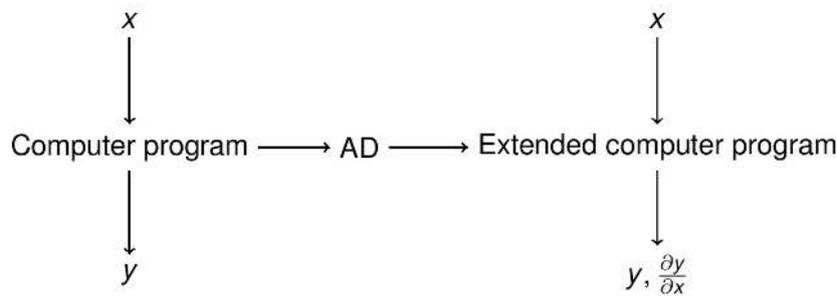


Figure C.1: **AD** concept

The **NR** method (C.1) gives us **quadratic convergence**(C.2).

$$(C.1) \quad \left\{ \begin{array}{l} \text{We start with } \mathbf{u}_0 \\ \mathbf{R}(\mathbf{u}_i + \Delta \mathbf{u}_i) \approx \mathbf{R}(\mathbf{u}_i) + \frac{\partial \mathbf{R}}{\partial \mathbf{u}}(\mathbf{u}_i) \Delta \mathbf{u}_i = 0 \\ \text{Solving} \rightarrow \Delta \mathbf{u}_i = - \left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right)^{-1} \mathbf{R}(\mathbf{u}_i) \\ \mathbf{u}_{i+1} = \mathbf{u}_i + \Delta \mathbf{u}_i \\ \text{Convergence when: } \|\Delta \mathbf{u}_i\| < \epsilon_u \text{ or/and } \|\mathbf{R}\| < \epsilon_r \end{array} \right.$$

$$(C.2) \quad \lim_{k \rightarrow \infty} \frac{\|X_{k+1} - X_{sol}\|}{\|X_k - X_{sol}\|^2} = M, \text{ with } M > 0$$

Then (C.1) gives us the definition of the **RHS** and the **LHS** (C.3). In consequence it is possible to use the **AD** to obtain the exact definition of the system and achieve **quadratic convergence**

$$(C.3) \quad \mathbf{RHS} = \frac{\partial \delta \mathcal{W}}{\partial \delta \mathbf{u}}, \quad \mathbf{LHS} = - \frac{\partial \mathbf{RHS}}{\partial \mathbf{u}}$$

Sometimes the **LHS** matrix is called *tangent matrix*, this is due to the mathematical concept underlying the derivation of the **RHS** vector necessary to obtain the consistent **LHS** from (C.3), where in a **Single Degree Of Freedom (SDOF)** problem the graphic representation of derivatives corresponds with the *tangent* in the given functional (See Figure C.2).

From a developer point of view the main advantages of this technique is the fact that it reduces the development time, reduces the computation time, and gives us robust and consistent code where the step of consistent linearisation

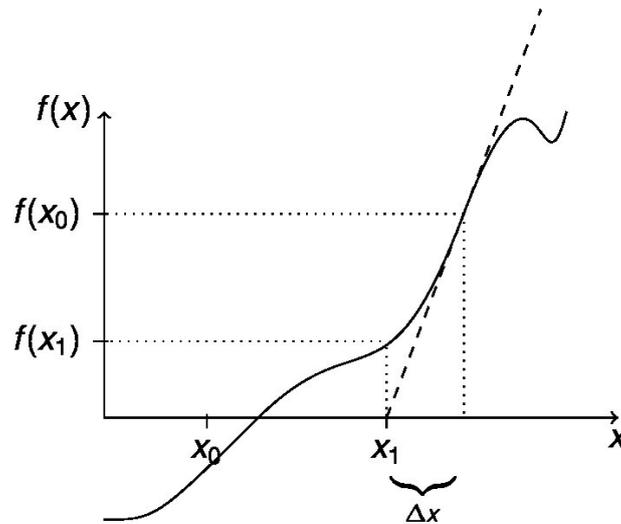


Figure C.2: NR representation for a SDOF

is considerably simplified. Furthermore, the time that will be required modify the code will be only the needed to modify the function to derive with **AD**, or the only thing that is actually debugged is the functional, not the generated code. This technique has been already used successfully for contact problems by *Jakub Lengiewicz*[[ArtLKS11](#); [ArtSLK10](#)].

C.3 Implementation

In the implementations presented in this work for the **AD** derivations, we have used *Sympy*[[Onl](#)]. The quality of the code obtained is lesser than in the case of the code developed by *Korelc*[[ArtKor97](#); [ArtKor02](#)] *AceGen*[[Onl](#)], but we have a full control of the whole workflow and the implementations actually done.

In the following section, we will introduce the principles of **AD** ([C.3.1.Principles of Automatic Differentiation](#)), which will help us to understand the design choices taken of the last section. The last section ([C.3.2.Kratos integration](#)) we introduce the integration between *Kratos* and *Sympy* and the modular design followed in order to consider properly the derivative definition without explicit definition of the values involved and get a fully quadratic convergence.

C.3.1 Principles of Automatic Differentiation

C.3.1.1 Derivation modes

As we said previously, the **AD** is based on the execution of elementary operations with known derivatives, each one of them can be evaluated exactly with the chain rule, independently of the complexity of the formulation. We can define two procedures[[BookJo16](#)], the **forward** mode and the **backward** mode. We can show this using the example shown in [[BookJo16](#)], taking as reference the equation from ([C.4a](#)).

$$(C.4a) \quad f = bc \text{ with } b = \sum_{i=1}^n a_i^2 \text{ and } c = \sin(b)$$

Considering a_i as n independent variables. The **forward** mode (C.4b) accumulates the derivatives of intermediate variables with respect to the independent variables.

$$(C.4b) \quad \begin{aligned} \nabla b &= \frac{db}{da_i} = 2x_i \text{ for } i = 1, 2, \dots, n \\ \nabla c &= \frac{dc}{da_i} = \cos(b)\nabla b_i \text{ for } i = 1, 2, \dots, n \\ \nabla f &= \frac{df}{da_i} = \nabla b_i c + b\nabla c_i \text{ for } i = 1, 2, \dots, n \end{aligned}$$

On the other hand, the **backward** mode, (C.4c), propagates adjoints $\bar{x} = \frac{\partial f}{\partial x}$, which are the derivatives of the final values, with respect to intermediate variables.

$$(C.4c) \quad \begin{aligned} \bar{f} &= \frac{df}{df} = 1 && 1 \\ \bar{c} &= \frac{df}{dc} = \frac{\partial f}{\partial c} \bar{f} = b\bar{f} && 1 \\ \bar{b} &= \frac{df}{db} = \frac{\partial f}{\partial b} \bar{f} + \frac{\partial c}{\partial b} \bar{c} = c\bar{f} + \cos(c)\bar{c} && 1 \\ \nabla f = \bar{a}_i &= \frac{\partial b}{\partial a_i} \bar{b} = 2a_i \bar{b} && \text{for } i = 1, 2, \dots, n \end{aligned}$$

The implementations on C.3.2.Kratos integration consider basically the **forward** mode. A proper implementation [BookJo16] makes use of both approaches and combine them in order to obtain the most efficient approach according to the estimated work ratio (C.5). On this equation the work ratio of the **forward** mode is proportional to the number of independent variables, and in the case of **backward** mode to the number of scalar-valued functions.

$$(C.5) \quad w_{ratio}(f(a)) = \frac{cost(f(a), \frac{\partial f}{\partial a})}{cost(f(a))}$$

C.3.1.2 Automatic Differentiation exceptions

Here we introduce the concept of **AD** exceptions, these exceptions appear on the derivative of an implicit dependence. For example, let's consider the following nonlinear equation (C.6a), which we will solve via the **NR** method. If we assume that (C.6a) depends on an additional independent variable \mathbf{a} , as shown on (C.6b). If \mathbf{u} depends on \mathbf{a} , this will introduce an additional implicit dependence on \mathbf{R} (C.6c). This kind of implicit dependency appears in many non-linear problems, for example on sensibility analysis or in order to obtain a fully consistent tangent on elasto-plasticity problems or in contact problems in large deformations and the boundary moves significantly. In order to compute the derivatives of this implicit dependence, we can apply the chain rule as in (C.6d). This can be solved by a direct application of the **AD**, but an alternative approach is to take into account the **AD** exceptions.

$$(C.6a) \quad \mathbf{R}(\mathbf{u}) = \mathbf{0}$$

$$(C.6b) \quad \mathbf{R}(\mathbf{u}, \mathbf{a}) = \mathbf{0}$$

$$(C.6c) \quad \mathbf{R}(\mathbf{u}(\mathbf{a}), \mathbf{a}) = \mathbf{0}$$

$$(C.6d) \quad \frac{D\mathbf{R}}{D\mathbf{a}} = \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{a}} + \frac{\partial \mathbf{R}}{\partial \mathbf{a}} = \mathbf{0}$$

$$(C.6e) \quad \frac{\partial \mathbf{u}}{\partial \mathbf{a}} = - \left(\frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right)^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{a}}$$

If \mathbf{R} is defined by an algorithm then we can obtain the tangent matrix needed in the **NR** scheme can be obtained by **AD**. Let's define the *computational derivative* concept (C.7a). Then we can define the computational derivative of (C.6b) on (C.7b). It is here where we can proceed in two ways, or directly deriving and solving in each step (C.6e), which can be very costly, or we can apply the **AD** exceptions.

$$(C.7a) \quad \frac{\hat{\delta}(\cdot)}{\hat{\delta}(\cdot)}$$

$$(C.7b) \quad \frac{\partial \mathbf{R}}{\partial \mathbf{u}} := \frac{\hat{\delta} \mathbf{R}}{\hat{\delta} \mathbf{u}}, \text{ and } \frac{\partial \mathbf{R}}{\partial \mathbf{a}} := \frac{\hat{\delta} \mathbf{R}}{\hat{\delta} \mathbf{a}}$$

There are two types of **AD** exceptions, the local exception (C.8a) or the global exception (C.8b). The difference between both exceptions is basically as the name indicates that in the case global exception the exceptions are defined globally, hence valid for every **AD** procedure call, while it applies only for the specific call. The approach followed in our implementation would be closer to this last exception, as the derivatives are defined in a modular way. For more details follow the next section.

$$(C.8a) \quad \frac{Df}{Da} := \frac{\hat{\delta} f(\mathbf{a}, \mathbf{u}(\mathbf{a}))}{\hat{\delta} \mathbf{a}} \Big|_{\frac{D\mathbf{u}}{Da} = - \left(\frac{\hat{\delta} \mathbf{R}}{\hat{\delta} \mathbf{u}} \right)^{-1} \frac{\hat{\delta} \mathbf{R}}{\hat{\delta} \mathbf{a}}}$$

$$\mathbf{u} := \mathbf{u} \Big|_{\frac{D\mathbf{u}}{Da} = - \left(\frac{\hat{\delta} \mathbf{R}}{\hat{\delta} \mathbf{u}} \right)^{-1} \frac{\hat{\delta} \mathbf{R}}{\hat{\delta} \mathbf{a}}}$$

$$(C.8b) \quad \dots$$

$$\dots$$

$$\frac{Df}{Da} := \frac{\hat{\delta} f(\mathbf{a}, \mathbf{u}(\mathbf{a}))}{\hat{\delta} \mathbf{a}}$$

C.3.2 Kratos integration

The classical way of optimising expressions[BookJo16] in **Computer Algebra Systems (CAS)** are searching for common sub-expressions before the generation of code is done. This can be insufficient when applied to general non-linear problems. This means that in the case of using *Sympy* environment directly, it is not possible to obtain complex or highly non-linear **FE** with this approach.

The design of our **AD** implementation has in mind these restrictions, and because of this, several utilities which integrate the *Sympy* environment has been implemented into *Kratos*. These utilities introduce simpler techniques than the ones developed *Korelc* for *AceGen*¹, but enough in order to obtain a relatively complex formulation.

The approach followed in order to take into account the **AD** exceptions (C.3.1.2. Automatic Differentiation exceptions) is the definition of the derivatives (4.6. Derivatives for contact mechanics linearization) as independent computations.

In order to do that, on the *Sympy* implementations we have defined an implicit dependency respect of the **DOF** of the terms with derivatives, defined in a standalone way. Those algebraic expressions are carried during all the **AD** procedures and finally replaced with the corresponding exceptions on the last stage of the code generation.

¹ Like the techniques of Simultaneous Stochastic Simplification[BookJo16].

This integration can be represented in a **UML** diagram, Figure C.3. On this Figure we can distinguish the previously stated steps. We define a base template in **C++** code, which is empty, and the **AD** generator created based on *Sympy* fills the template based in a given *Galerkin* functional. The derivatives are computed a priori, and the code generated includes them implicitly until the **C++** code conversion steps, where are replaced with the externally computed values.

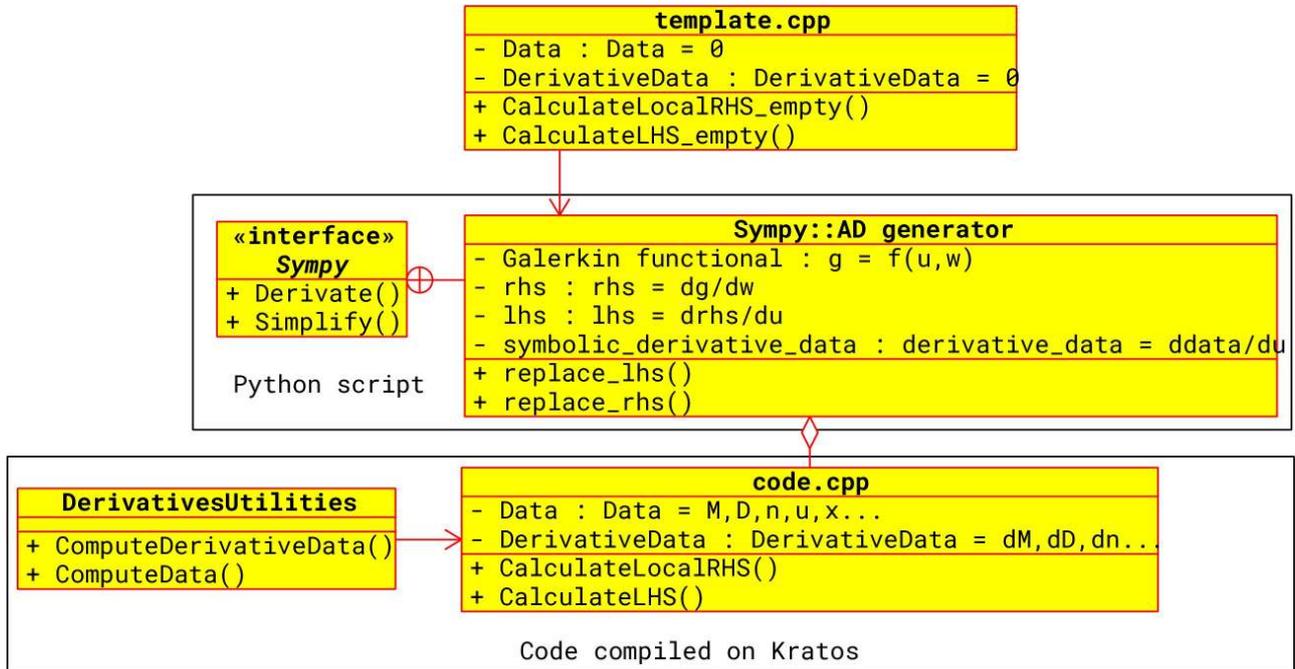


Figure C.3: **AD** workflow in a **UML** diagram

Bibliography

Books

- [BookJo16] *Automation of Finite Element Methods*. Peter Wriggers (auth.) Joe Korelc. Springer International Publishing. 1st ed. 2016.
- [BookWri08] *Nonlinear finite element methods*. Peter Wriggers. Springer. 1st ed. 2008.

Articles

- [ArtGru82] “Automatic Differentiation: Techniques and Applications. Lecture Notes in Computer Science 120.” F. Grund. In: *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*. No. 7, Vol. 62, 1982, pp. 355–355. DOI: [10.1002/zamm.19820620735](https://doi.org/10.1002/zamm.19820620735).
- [ArtHec+09] “FreeFem++, ver. 3.7”. F Hecht, O Pironneau, A Le Hyaric, and K Ohtsuka. In: . 2009,
- [ArtKor02] “Multi-language and Multi-environment Generation of Nonlinear Finite Element Codes”. J. Korelc. In: *Engineering with Computers*. No. 4, Vol. 18, 2002, pp. 312–327. DOI: [10.1007/s003660200028](https://doi.org/10.1007/s003660200028).
- [ArtKor97] “Automatic generation of finite-element code by simultaneous optimization of expressions”. Joe Korelc. In: *Theoretical Computer Science*. No. 1, Vol. 187, 1997, pp. 231–248. DOI: [10.1016/s0304-3975\(97\)00067-4](https://doi.org/10.1016/s0304-3975(97)00067-4).
- [ArtLKS11] “Automation of finite element formulations for large deformation contact problems”. Jakub Lengiewicz, Joe Korelc, and Stanisaw Stupkiewicz. In: *International Journal for Numerical Methods in Engineering*. No. 10, Vol. 85, 2011, pp. 1252–1279. Wiley Online Library. DOI: [10.1002/nme.3009](https://doi.org/10.1002/nme.3009).
- [ArtLog07] “Automating the finite element method”. Anders Logg. In: *Archives of Computational Methods in Engineering*. No. 2, Vol. 14, 2007, pp. 93–138. Springer. DOI: [10.1007/s11831-007-9003-9](https://doi.org/10.1007/s11831-007-9003-9).
- [ArtSLK10] “Sensitivity analysis for frictional contact problems in the augmented Lagrangian formulation”. Stanisaw Stupkiewicz, Jakub Lengiewicz, and Joe Korelc. In: *Computer Methods in Applied Mechanics and Engineering*. No. 33-36, Vol. 199, 2010, pp. 2165–2176. Elsevier. DOI: [10.1016/j.cma.2010.03.021](https://doi.org/10.1016/j.cma.2010.03.021).

Online resources

- [Onl] *AceGen. Multi-language, Multi-environment Numerical Code Generation*. URL: <https://www.wolfram.com/products/applications/acegen/>.
- [Onl] *SymPy is a Python library for symbolic mathematics*. URL: <https://www.sympy.org/>.

Appendix D

Constrained optimisation problems

“The mathematical sciences particularly exhibit order, symmetry, and limitation; and these are the greatest forms of the beautiful.”

Aristotle
(384 - 322 BC, Greek philosopher)

D.1 Introduction

The following appendix introduces the concepts concerning the constraint optimisation needed in the resolution of the contact problems. These optimisation methods are of general use from a mathematical point of view and can be found extensively on the literature [BookLei04; ArtQS99], especially to the kind of problems we are interested to solve that are the contact problems [BookAnd01; BookYas13]. The main problematic motivating us to study these constraint optimisation methodologies is the existence of many physical phenomena such as the impact, friction, among others; which can be studied with mathematical models with some kind of discontinuous or non-smooth behaviour. In detail, this problem suffers of bifurcation points, therefore, may lead to non-smooth dynamic systems, where the dynamic in each mode is associated with a different set of smooth differential equations [BookLei04].

The methods we are going to present in this annexe are the following ones:

- **Penalty method** (also known as exterior point method [PhDYas11]). This is probably the simplest and probably most extended method. The last statement comes particularly true when the method can deal with explicit contributions, then used on explicit simulations. For more details and deeper explanation see [D.2. Penalty method](#).
- **Lagrange multiplier method**. The **LM** can be given a rigorous justification within the context of the variational calculus [BookCA01], and in contrast to penalty method gives an exact solution. How it is derived and several additional details are presented on [D.3. Lagrange Multiplier method](#).
- **Augmented Lagrange Multiplier method**. The **ALM** basically combines the former methods, for a deeper understanding see [D.4. Augmented Lagrange Multiplier method](#).

There is another method that we consider is important to highlight, this is the resolution of **MPC** (see [D.5. MultiPoint Constraint \(Master-Slave elimination method\)](#)) with the *master-slave elimination* method. This method is not an optimisation method by itself, but can be used in order to impose constraints of interest in the resolution of mechanical problems. The main reason to mention it in addition to the previous methods, it is due to the fact is available on

B.2.Kratos *Multiphysics* natively. Unfortunately, it is not particularly good to deal with complex contact constraints as it will be shown later.

There are many methods omitted on this list¹, as we said we will present and study just the methods used in the developments concerning this work.

D.2 Penalty method

D.2.1 Introduction

The **PM** method can be idealised as the presence of fictitious elastic structural elements (Figure D.1) which enforce the constraint approximately for mechanical problems[BookCA01]. In the most general case, it can be defined as penalty functional appended to our functional of interest, which increases accordingly to how severely the constraint is violated[BookKO88].

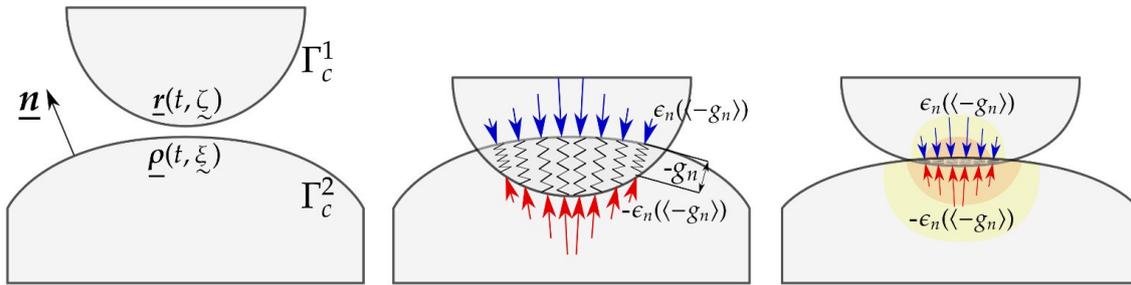


Figure D.1: Analogy between *penalty* method and springs. Image from [BookYas13]

D.2.2 Formulation

In order to formulate a generic constrained problem and solve it with the **PM** we just need to add the following to our functional (D.1).

We start with the following base functional and the following constraint. It can be extended to any number of constraints, in fact, at contrary to other optimisation methods **LM** can deal with overconstrained problems (this does not mean it will fulfil all of them satisfactory).

$$(D.1a) \quad \begin{cases} f(w, x) = \text{Base functional} \\ \text{constraint} \leq 0 \end{cases}$$

Where w and x are the test function and **DOF** respectively. Then we add the following penalty functional to our base functional. The objective of this penalty term is precisely to penalise the infringement of this constraint.

$$(D.1b) \quad f_p(w, x) = f(w, x) + \frac{\varepsilon}{2} \max(0, \text{constraint})^2$$

Where ε is our penalty *parameter*. Simple, yet powerful and generic.

As in any functional, in order to deduce the corresponding **LHS** and **RHS** we do the following (D.2) (as follows it will be the same for the rest of methods). This is already stated on the appendix concerning **AD** (C. Automatic differentiation).

$$(D.2a) \quad \text{RHS}(x) = \frac{\partial f_p(w, x)}{\partial w}$$

$$(D.2b) \quad \text{LHS}(x) = -\frac{\partial \text{RHS}(x)}{\partial x}$$

¹ In [PhDYas11; BookWri06] we can see the following methods as an extension of the list presented: *Barrier* method (also known as interior points method), perturbed *Lagrangian*[ArtSWT85a], *Nitsche*[ArtNit71] method, *cross-constraint* method.

D.2.3 Applicability on contact problems

The penalty method is general and powerful enough to be applied on contact problems. So in case we want to introduce it into a contact problem we can do the following (D.12).

We define the non-penetration condition, together with the **HSM**.

$$(D.3a) \quad g \geq 0, \quad \sigma_n \leq 0, \quad g\sigma_n = 0, \quad \sigma_t = 0$$

In order to fulfil these conditions, we can define the contact pressure as a continuous function of the penetration.

$$(D.3b) \quad \begin{cases} g \geq 0, & \sigma_n = 0, & g\sigma_n = 0 \\ g < 0, & \sigma_n = \varepsilon_n(-g) < 0, & g\sigma_n \neq 0 \end{cases}$$

This approximation implies that the non-penetration condition is not respected, but the penetration movement is resisted, the deeper is the penetration the stronger is the reaction. Then the energy accumulated on these continuous linear *springs* is:

$$(D.3c) \quad f_p(x) = f(x) - \int_0^{-\langle -g \rangle} \varepsilon_n \langle -g' \rangle dg' = \int_0^{-\langle -g \rangle} \varepsilon_n g' dg' = \frac{1}{2} \varepsilon_n \langle -g \rangle^2$$

Where $\langle \cdot \rangle$ are the *Macaulay* brackets. Integrating over the boundary, over the normal direction, the contribution to the balance of virtual works writes as:

$$(D.3d) \quad \delta f_p(x) = \delta f(x) + \int_{\Gamma_c^1} \varepsilon_n (-g_n) \delta g_n d\Gamma_c^1 = \int_{\Gamma_c^1} \varepsilon_n (\langle -g_n \rangle) \delta g_n d\Gamma_c^1$$

D.2.3.1 Adapted Penalty Method

On (D.12) the ε is constant, in contrast we can adapt dynamically the value of ε in order to improve the convergence of the system. *Bussetta*[ArtBMP] presents the **Adapted Penalty Method (APM)** as a plausible method from doing so. The concept is summarised on the Figure D.2, where the function \mathcal{F} that defines the ε varies in function of the penetration (g_n), see (D.4).

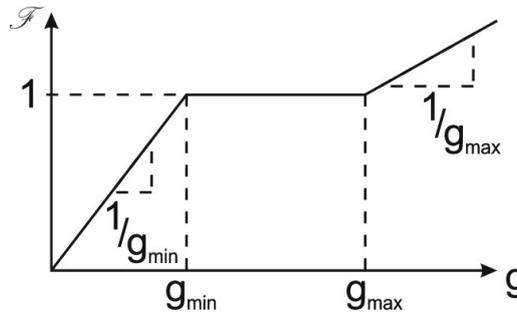


Figure D.2: Representation of \mathcal{F} . Image from [ArtBMP]

$$(D.4a) \quad \varepsilon_{n_{i+1}} = \mathcal{F}(|g_i|, g_{\min}, g_{\max}) \varepsilon_{n_i}$$

$$(D.4b) \quad \mathcal{F}(|g_i|, g_{\min}, g_{\max}) = \begin{cases} \frac{|g_i|}{g_{\max}} & \text{if } |g_i| > g_{\max} \\ \frac{|g_i|}{g_{\min}} & \text{if } |g_i| < g_{\min} \\ 1 & \text{else} \end{cases}$$

This is important because as we have previously stated, the choice of the penalty coefficients is of the utmost importance in order to get an effective solution. The smaller these coefficients are, the result obtained may not respect the imposed contact constraint, allowing penetrations. On the other hand, a large value of these coefficients may induce numerical oscillations and *ill-conditioning* of the system of equations, preventing the convergence of the algorithm. There are other alternative proposals to this in order to improve the **PM**, with more layers of complexity, like for example the consideration of a double penalty[PhDHet14].

D.3 Lagrange Multiplier method

D.3.1 Introduction

The **LMM**[BookIK08], named after *Joseph-Louis Lagrange*, is used on optimisation theory to find the *extremum*, *saddle point*, of a functional subjected to **equality constraints**[BookAkh88]. The basic idea is to convert a constrained problem into a form such that the derivative test of an unconstrained problem can still be applied[OniLag], the obtained functional is called *Lagrangian* ($\mathcal{L}(x, \lambda)$). On this *Lagrangian* functional, we add a new unknown called *Lagrange multiplier*, usually referred as λ . The great advantage of this method is that it allows the optimisation to be solved without explicit parameterisation in terms of the constraints.

D.3.2 Formulation

The *Lagrangian* is constructed in the following manner (D.5b), this for problems where the constraint is not limited by any inequality. For problems where we deal with inequality constraints, we need to formulate the problem following some kind of active set strategies[PhDYas11], we can define three types of **PDASS**(D.5d), graphical examples for a better understanding of the problem will be presented on the section D.7.2.Non-linear spring contact problem with wall.

We want to minimize f subjected to g .

$$(D.5a) \quad \min_{g(x)=0} f(x) \rightarrow \nabla \mathcal{L}_\lambda(x, \lambda) = 0$$

Where the *Lagrangian* is defined as follows:

$$(D.5b) \quad \mathcal{L}_\lambda(x, \lambda) = f(x) + \lambda g(x)$$

The gradient (∇) of \mathcal{L} can be defined as:

$$(D.5c) \quad \nabla \mathcal{L}_\lambda(x, \lambda) = \begin{bmatrix} \frac{\partial \mathcal{L}_\lambda}{\partial x} \\ \frac{\partial \mathcal{L}_\lambda}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x)}{\partial x} + \lambda \frac{\partial g(x)}{\partial x} \\ g(x) \end{bmatrix} = 0$$

In case we want to consider to solve a problem with inequality constraint we can consider the following three active set strategies (**PDASS**), where $\langle \cdot \rangle$ denotes the *Macaulay* brackets, used to describe the ramp function:

$$(D.5d) \quad \begin{cases} \text{Active set strategy 1: } g(x) > 0 : f(x) & g(x) \leq 0 : \mathcal{L}_\lambda(x, \lambda) \\ \text{Active set strategy 2: } \lambda > 0 : f(x) & \lambda \leq 0 : \mathcal{L}_\lambda(x, \lambda) \\ \text{Active set strategy 3: } g(x) > 0 \text{ and } \lambda > 0 : f(x) & g(x) \leq 0 \text{ or } \lambda \leq 0 : \mathcal{L}_\lambda(x, \lambda) \end{cases}$$

These strategies, combined with (D.5b) can be expressed the following way, with the *Macaulay* bracket.

$$(D.5e) \quad \begin{cases} \text{Active set strategy 1: } \mathcal{L}_\lambda(x, \lambda) = f(x) - \langle -\lambda \rangle g(x) \\ \text{Active set strategy 2: } \mathcal{L}_\lambda(x, \lambda) = f(x) - \lambda \langle -g(x) \rangle \\ \text{Active set strategy 3: } \mathcal{L}_\lambda(x, \lambda) = f(x) - \langle -\lambda \rangle g(x) - \lambda \langle -g(x) \rangle - \langle \lambda \rangle \langle g(x) \rangle \end{cases}$$

The influence of each of the **PDASS** can be summarised as:

1. This one is based on the check of the violation of the g , it is the most commonly used due to its robustness[PhDYas11], but with a higher number of **NL** iterations in order to converge.
2. This one checks the positivity of λ . This one may lead to the continuous switch between the base functional (f) and the *Lagrangian* \mathcal{L}_λ , lacking the robustness provided by the first strategy, even despite of having a faster rate of convergence on ideal conditions.
3. The third alternative provides the robustness of the first strategy and the rate of convergence of the second one, but also may diverge faster in case the initial solution is far from the final one. This last point forces to consider a slow loading of the problem, or a fine-tuning in the **BC**.

The solution of our *minimisation* problem is a stationary point in (D.5b), but usually not all stationary problems of the *Lagrangian*(D.5b) are solutions of the initial *minimisation* problem. The resulting system of equations has a **higher number of unknowns** with the corresponding additional computational cost. Additionally if we compute the *Hessian* (**H**) of the *Lagrangian* functional, we will obtain the corresponding **LHS** of our optimisation problem, being the *gradient* equivalent to minus **RHS**.

First define **H** as the derivative of the *gradient*

$$(D.6a) \quad \mathbf{H}(x) = \nabla g(x) = \begin{bmatrix} \nabla g_1(x)^T \\ \nabla g_2(x)^T \\ \vdots \\ \nabla g_n(x)^T \end{bmatrix}$$

Then we can compute the **H** from the gradient (D.5c).

$$(D.6b) \quad \mathbf{H}(\mathcal{L}_\lambda(x, \lambda)) = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_\lambda}{\partial x^2} & \frac{\partial^2 \mathcal{L}_\lambda}{\partial \lambda \partial x} \\ \frac{\partial^2 \mathcal{L}_\lambda}{\partial x \partial \lambda} & \frac{\partial^2 \mathcal{L}_\lambda}{\partial \lambda^2} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x^2} & \frac{\partial g(x)}{\partial x} \\ \frac{\partial g(x)}{\partial x} & \mathbf{0} \end{bmatrix}$$

On (D.6) the **H** of \mathcal{L} is shown, and at (D.6b) we can see the *typical* **LHS** structure for a problem solved with **LMM**. In here the equations have a zero diagonal for each multiplier term. Thus, special care is needed in the solution process to avoid division by the zero diagonal[BookZTF14]. This issue is partially solved with the use of **ALM**.

D.3.3 Applicability on contact problems

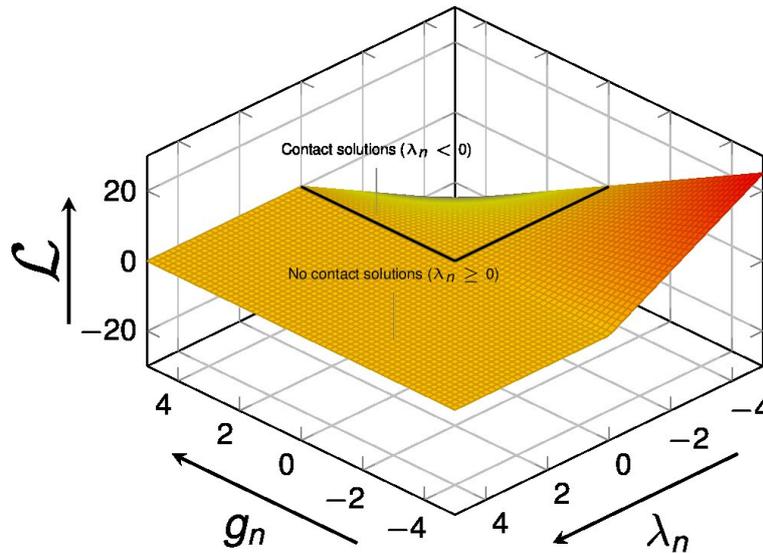


Figure D.3: Lagrangian function for the contact problem

Following the same reasoning of D.2.3.Applicability on contact problems, we can define a functional for contact mechanics using the **LMM**. The **HSM** conditions (D.3a) are still applied here, then we can obtain the corresponding *Lagrangian* (D.7).

Defining λ_n as the normal *Lagrange* multiplier which defines the normal contact pressure.

$$(D.7a) \quad \mathcal{L}_\lambda(x, \lambda) = f(x) + \int_{\Gamma_c^1} \lambda g_n(x) d\Gamma_c^1$$

The variation of the *Lagrangian* then can be expressed as:

$$(D.7b) \quad \delta \mathcal{L}_\lambda(x, \lambda) = \delta f(x) + \int_{\Gamma_c} g_n(x) \delta \lambda + \lambda \delta g_n(x) d\Gamma_c = 0$$

Where λ_n and g_n have to be fulfilled the minimisation problem with inequality constraints (D.3a). This means that from (D.5d) the preferable strategy would be the third one.

We can represent (D.7) on the *locus*² of Figure D.3, where (D.7) is represented on function of g_n and λ_n . On this figure we can appreciate the fact that on tension state ($\lambda_n > 0$) there is no solution to the functional (zero solutions), this generates a discontinuity which affects the resolution the resulting system of equations. This last problematic can be solved precisely with the application of **ALM**.

D.4 Augmented Lagrange Multiplier method

D.4.1 Introduction

The **ALM** was originally introduced by *Arrow and Solow*[ArtAS58], and later improved by *Powell and Hestenes*[ArtPow69; ArtHes69], reason why was originally named as *the multiplier method of Hestenes and Powell*. The method consists on a regularised version of **LMM**[PhDYas11]. This regularization occurs via the inclusion of **penalty** parameter, but once the problem has reached convergence the influence of the **penalty** parameters disappears, resulting in exact fulfilment of the constraint.

D.4.2 Formulation

D.4.2.1 Standard formulation

The *Lagrangian* is constructed in the following manner (D.8b), this is basically the combination of the **PM** and the **LMM** (D.8a). This method was generalised by *Rockafellar*[ArtRoc73b; ArtRoc73a] for inequality constraints (D.8c), in this particular case $g(x) \geq 0$.

The *Lagrangians* for the **PM** and **LMM** as previously redefined are:

$$(D.8a) \quad f_p(x) = f(x) + \frac{1}{2}\varepsilon g(x)^2; \quad \mathcal{L}_\lambda(x, \lambda) = f(x) + \lambda g(x)$$

Combination of previous methods, we obtain the augmented *Lagrangian*.

$$(D.8b) \quad \mathcal{L}_{\bar{\lambda}}(x, \lambda) = \mathcal{L}_\lambda(x, \lambda) + \frac{1}{2}\varepsilon g(x)^2 = f(x) + \lambda g(x) + \frac{1}{2}\varepsilon g(x)^2$$

Expressed as inequality constraints $g(x) \geq 0$:

$$(D.8c) \quad \mathcal{L}_{\bar{\lambda}}(x, \lambda) = f(x) - \frac{1}{2\varepsilon} \left(\lambda^2 - \langle -(\lambda + \varepsilon g(x)) \rangle^2 \right)$$

In an expanded form, it rewrites as:

$$(D.8d) \quad \mathcal{L}_{\bar{\lambda}}(x, \lambda) = f(x) + \begin{cases} \lambda g(x) + \frac{1}{2}\varepsilon g(x)^2 & , \lambda + \varepsilon g(x) \leq 0 \\ -\frac{1}{2\varepsilon}\lambda^2 & , \lambda + \varepsilon g(x) > 0 \end{cases}$$

We can call $\lambda + \varepsilon g(x)$ as $\bar{\lambda}$, or augmented *Lagrangian*. The gradient (∇) of (D.8d) can be expressed as:

$$(D.8e) \quad \nabla \mathcal{L}_{\bar{\lambda}}(x, \lambda) = \nabla f(x) + \begin{cases} \begin{bmatrix} (\lambda + \varepsilon g(x)) \frac{\partial g(x)}{\partial x} \\ g(x) \end{bmatrix} & , \bar{\lambda} \leq 0 \\ \begin{bmatrix} 0 \\ -\frac{\lambda}{\varepsilon} \end{bmatrix} & , \bar{\lambda} > 0 \end{cases}$$

Equivalent to what is done in (D.6) we can obtain the *Hessian* (**H**) of (D.8b). In (D.9) we see that in this case, we have partially solved the issue concerning to the zero terms appearing on the diagonal, compared to the previously obtained **H**. The problematic have been solved partially because only affects to the **inactive** set ($\bar{\lambda} > 0$), in the case

²A locus is the set of all points (usually forming a curve or surface) satisfying some condition.

of the active set we still have to deal with this issue. A **Generalized Newton-Raphson method (GNM)** was proposed by *Alart and Curnier* for nonsmooth potential for the **ALM**[ArtAla88; ArtAC91; ArtAla97].

$$(D.9) \quad \mathbf{H}(\mathcal{L}_{\bar{\lambda}}(x, \lambda)) = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_{\bar{\lambda}}}{\partial x^2} & \frac{\partial^2 \mathcal{L}_{\bar{\lambda}}}{\partial \lambda \partial x} \\ \frac{\partial^2 \mathcal{L}_{\bar{\lambda}}}{\partial x \partial \lambda} & \frac{\partial^2 \mathcal{L}_{\bar{\lambda}}}{\partial \lambda^2} \end{bmatrix} = \begin{cases} \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x^2} + \frac{\partial^2 g(x)}{\partial x^2} (\lambda + \varepsilon g(x)) + \varepsilon \left(\frac{\partial g(x)}{\partial x} \right)^2 & \frac{\partial g(x)}{\partial x} \\ \frac{\partial g(x)}{\partial x} & 0 \end{bmatrix} & , \bar{\lambda} \leq 0 \\ \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x^2} & 0 \\ 0 & \frac{1}{\varepsilon} \end{bmatrix} & , \bar{\lambda} > 0 \end{cases}$$

D.4.2.2 Uzawa iteration

So as we have seen, we still have some problematic from the standard **LMM**, like the zero diagonal terms or the non-positivity of the *Lagrange* multiplier. In order to solve that, on 1958 the Japanese economist *Uzawa* presented[ArtAS58; ArtAHU58] an alternative approach in order to solve this problematic following an iterative approach, known as *Uzawa* algorithm[Onl].

Let's decompose the increment of the *Lagrangian* into two different components, one for the x **DOF** and the other for the update of the *Lagrange* multiplier (D.10).

Calling λ_i and x_i the respective solution for the **NL** iteration number i . Then we can define the augmented *Lagrangian* in a given increment of a **NL** iteration as:

$$(D.10a) \quad \mathcal{L}_{\bar{\lambda}}(x_i + \Delta x_i, \lambda_i + \Delta \lambda_i) \approx \mathcal{L}_{\bar{\lambda}}(x_i + \Delta x_i, \lambda_i) + \left. \frac{\partial \mathcal{L}_{\bar{\lambda}}(x, \lambda_i)}{\partial \lambda} \right|_{\lambda_i} \Delta \lambda_i + O(\lambda_i^2) = 0$$

We can do the same for the Δx term, and expand it.

$$(D.10b) \quad \begin{aligned} \mathcal{L}_{\bar{\lambda}}(x_i + \Delta x_i, \lambda_i) &\approx \mathcal{L}_{\bar{\lambda}}(x_i, \lambda_i) + \left. \frac{\partial \mathcal{L}_{\bar{\lambda}}(x, \lambda_i)}{\partial x} \right|_{x_i} \Delta x_i = 0 \\ \left[f(x_i) + \lambda_i g(x_i) + \frac{1}{2} \varepsilon g(x_i)^2 \right] &+ \left[\frac{\partial f(x)}{\partial x} + [\lambda_i + \varepsilon g(x)] \frac{\partial g(x)}{\partial x} \right] \Big|_{x_i} \Delta x_i + O(x_i^2) = 0 \end{aligned}$$

Finally, concerning the *Lagrange* multiplier we apply an update procedure:

$$(D.10c) \quad \lambda_{i+1} = \lambda_i + \varepsilon g(x_i), \quad \Delta \lambda_i = \varepsilon g(x_i)$$

Convergence of this method is *linear* for the part relative to the *Lagrange* multiplier ($O(\lambda_i)$), as we have removed the second order part of (D.10a). Besides the advantage of the method is that the \mathcal{L} is smooth so that a standard **NR** is applicable.

For the present work, the standard approach has been taken into consideration. On the contact chapter of this work the details concerning why the standard **ALM** approach has been chosen instead of the *Uzawa* iteration, check the corresponding section for further details, 4.3.3.2.1.3.ALM.

D.4.3 Applicability on contact problems

Following a similar criteria to the one employed on (D.7), the equivalent formulation can be obtained for the **ALM**, see (D.11). In contrast to the *locus* for the **LMM**, the corresponding *locus* for the **ALM** (Figure D.4) is C^1 differentiable saddle-point, with the corresponding advantages for the resolution of the problem. Like the existence of a system of equations contributions (**LHS**, **RHS**) even for $\bar{\lambda}_n > 0$.

Again, considering λ_n as the normal contact stress, and calling $\bar{\lambda}_n$ the augmented contact stress, computed as $\bar{\lambda} = \lambda + \varepsilon g_n(x)$.

$$(D.11a) \quad \mathcal{L}_{\bar{\lambda}}(x, \lambda) = f(x) + \int_{\Gamma_c} \begin{cases} \lambda g_n(x) + \frac{1}{2} \varepsilon g_n(x)^2 & , \bar{\lambda} \leq 0 \\ -\frac{1}{2\varepsilon} \lambda^2 & , \bar{\lambda} > 0 \end{cases} d\Gamma_c^1$$

The variation of the augmented *Lagrangian* then can be expressed as:

$$(D.11b) \quad \delta \mathcal{L}_{\bar{\lambda}}(x, \lambda) = \delta f(x) + \int_{\Gamma_c} \begin{cases} \delta \lambda g_n(x) + (\lambda + \varepsilon g_n(x)) \delta g_n(x) & , \bar{\lambda} \leq 0 \\ -\frac{\lambda}{\varepsilon} \delta \lambda & , \bar{\lambda} > 0 \end{cases} d\Gamma_c = 0$$

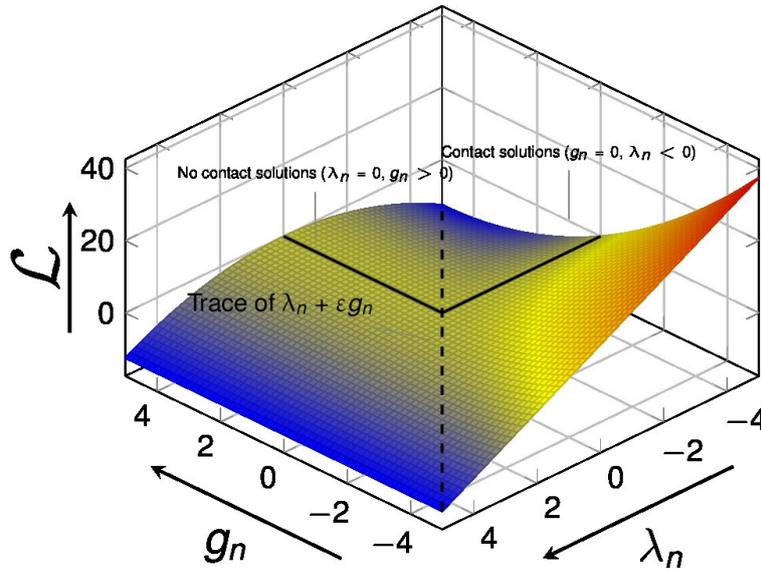


Figure D.4: Augmented Lagrangian function for the contact problem

D.4.3.1 Adapted Augmented Lagrangian Method

Algorithm 7 Adaptation of normal penalty coefficient[ArtBMP]

Require: ε_n, g_i and g_{i-1}

```

1: procedure ADAPTATION OF NORMAL PENALTY COEFFICIENT
2:   if  $g_i \times g_{i-1} < 0$  then
3:     if  $g_i \times g_{i-1} < 0$  then
4:        $\varepsilon_n = |(\varepsilon_n g_{i-1}) / g_i \times (|g_i| + g_{max}) / (g_i - g_{i-1})|$ 
5:     else
6:        $\varepsilon_n = |\varepsilon_n g_{i-1} / (10 g_i)|$ 
7:   else if  $g_i > g_{max}$  then
8:     if  $|g_i - g_{i-1}| > \max(g_i/10, g_{i-1}/10, 5g_{max})$  then
9:        $\varepsilon_n = 2\varepsilon_n$ 
10:    else if  $|g_i| = |g_{i-1}| \pm 1\% < 10g_{max}$  then
11:       $\varepsilon_n = \varepsilon_n \left( \sqrt{(|g_i| / g_{max} - 1) + 1} + 1 \right)^2$ 
12:    else if  $g_i > g_{max}$  then
13:       $\varepsilon_n = 2\varepsilon_n (g_{i-1} / g_i)$ 
14:    else
15:       $\varepsilon_n = \varepsilon_n \left( \left( \sqrt{(|g_i| / g_{max} - 1) + 1} \right) \right)$ 
16:  else
17:     $\varepsilon_n = \varepsilon_n$ 

```

On the expressions from (D.11) it is assumed that the ε is constant, but there are techniques which adapt dynamically the value of ε in order to improve the convergence of the system. We can mention in particular the work of *Bussetta*[PhDBus; ArtBMP] who created **AALM**, an **ALM** with a corresponding algorithm in order to update the ε in function of the current penetration (g_n). On Algorithm 7 the procedure is detailed, here three cases can be distinguished: The sign of g_n changes ($g_i \times g_{i-1} < 0$), the absolute value of g_n is more relevant than the defined limit ($g_i > g_{max}$), or the last case where the absolute value of g_n is smaller than the limit ($g_i > g_{max}$).

D.5 MultiPoint Constraint (Master-Slave elimination method)

D.5.1 Introduction

The **MPC** is a type of **MultiFreedom Constraint (MFC)**³ where each several displacements (or **DOF**) per node [BookCA01]. The nature of these constrains can be *linear* if all the displacement components appear linearly on the **LHS**, and *non-linear* otherwise. These relationships can be solved using the former optimisation methods, but in D.5.2. Formulation we will present the *master-slave* method from *Felippa* [BookCA01] book.

D.5.2 Formulation

Following the notation is taken from [BookCA01], and can be summarised in the following way (D.12).

Our system of equations looks like (D.12a).

$$(D.12a) \quad \text{LHS} \Delta \mathbf{x} = \text{RHS} \text{ onwards } \text{LHS} = \mathbf{A} \text{ and } \text{RHS} = \mathbf{b}$$

If we define relation matrix T as the following:

$$(D.12b) \quad \mathbf{x} = \mathbf{T} \bar{\mathbf{x}} + \mathbf{g}$$

That in a incremental approach leaves the following:

$$(D.12c) \quad \mathbf{x}_i = \mathbf{x}_{i-1} + \Delta \mathbf{x} = \mathbf{T} \bar{\mathbf{x}} + \mathbf{g} \rightarrow \mathbf{x}_{i-1} + \Delta \mathbf{x} = \mathbf{T} (\bar{\mathbf{x}}_{i-1} + \Delta \bar{\mathbf{x}}) + \mathbf{g}$$

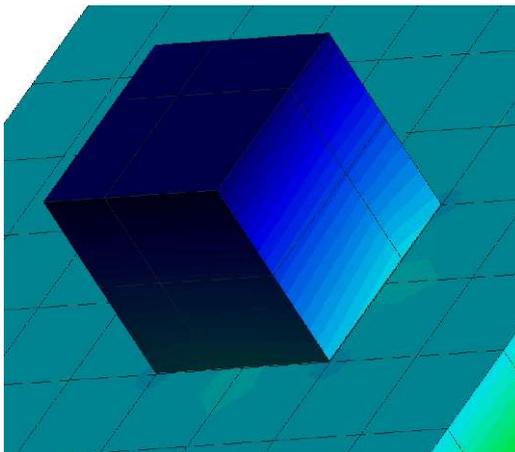
If the previous time step is **converged** then $\mathbf{x}_{i-1} = \mathbf{T} \bar{\mathbf{x}}_{i-1} + \mathbf{g}$ then simply:

$$(D.12d) \quad \Delta \mathbf{x}_i = \mathbf{T} \Delta \bar{\mathbf{x}}_i + \begin{cases} i = 0, \mathbf{g} \\ i \neq 0, \mathbf{0} \end{cases}$$

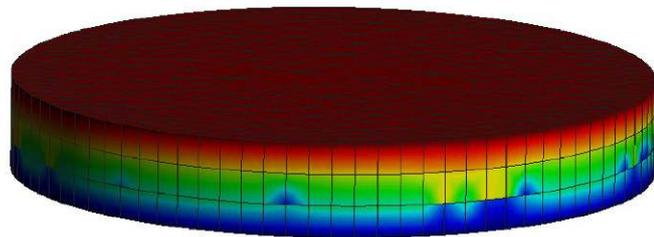
This brings us to the following eliminated system:

$$(D.12e) \quad \bar{\mathbf{A}} \Delta \bar{\mathbf{x}} = \bar{\mathbf{b}} \text{ in which } \bar{\mathbf{A}} = \mathbf{T}^T \mathbf{A} \mathbf{T}, \bar{\mathbf{b}} = \mathbf{T}^T \left(\mathbf{b} - \begin{cases} i = 0, \mathbf{A} \mathbf{g} \\ i \neq 0, \mathbf{0} \end{cases} \right)$$

D.5.3 Applicability on contact problems



(a) Patch test not passing



(b) Not matching meshes giving bad quality results

Figure D.5: Patch tests solutions obtained with *Master-Slave* elimination method

In order to consider the **MPC** to compute the contact constraint, we need to *deactivate* them once the residuals concerning the corresponding **DOF** indicates the contact constraint is on tension, and them not respecting the unidirectionality of the contact constraint which requires compression.

³This can be defined as functional equation where two or more **DOF** are written in a functional which connects them.

Unfortunately, the **MPC** method presented with *Master-Slave* elimination method does not give a good solution once the meshes are not matching (similar to a **NTN** method) for a problem of deformable domains. The solutions presented on Figure D.5 represent the solution obtained with this method. The weights have been obtained with a mortar segmentation, therefore exact, and the results obtained are not good.

The first case, Figure D.5a, represents a 3D patch test, with not matching meshes, in this case the constraint is fulfilled, but as both domains are deformable the resulting solution is not continuous. The solution from Figure D.5b deals with the same problem, but with a higher number of **DOF**, then the bad quality of the results is even more noticeable.

In order to obtain good results with the methods, the *master* domain must be *rigid*, or *quasi-rigid*, or the meshes to match between the interface of the domains.

D.6 Summary of the different methods

The presented methods can be summarised in the following table, see D.1, inspired and expanded from [BookCA01]. **ALM** and **LMM** are virtually identical except for the dependence of the **ALM** of the penalty (not affecting the final solution). Additionally the **ALM** retains partially the positive definiteness due to the fact that adds relatively small terms on the diagonal of the **LHS** matrix.

Method	<i>Master-Slave Elimination</i>	PM	LMM	ALM
Generality	Fair	Excellent	Excellent	Excellent
Ease of implementation	Poor to fair	Good	Fair	Fair
Sensitivity to user decisions	High	High	Small to none	Small
Accuracy	Variable	Variable	Excellent	Excellent
Sensitivity as regards constraint dependence	High	None	High	High
Retains positive definiteness	Yes	Yes	No	Partially

Table D.1: Comparison of different methods presented

Apart from **ALM** there are other methods that try to tackle the issues presented by **LMM**. In short we can list the following two methods:

- **Double Lagrange multiplier:** The *Lagrange doubles*⁴ (used in the *Castem2000* code) avoids the issues from the **LMM** provoked by the non-positive **LHS** matrix obtained with the method. This method duplicates the number of *Lagrange* multipliers and adds some dummy terms to the **LHS** matrix, which increases the computational cost but makes the **LHS** positive defined. For further details see [Onl].
- **Perturbed Lagrangian:** *Simo, Wriggers and Taylor* proposed a *perturbed Lagrangian* formulation for the solution of contact problems. That formulation can be classified as a stabilised method, preserving the stability of the discretised problem if the penalty parameter, ε , is small enough. For further details see [ArtSWT85b].

D.7 Numerical examples

In this section we compare the different approaches introduced (except for D.5.MultiPoint Constraint (*Master-Slave elimination method*)). The problems vary on complexity, and helps us to understand the way these different methods behave when computing the corresponding contribution of the constraints.

D.7.1 Initial spring wall problem

We start with a very simple case from [PhDYas11]. This problem consists on a **SDOF** problem, where we idealised a single spring with a *wall* constraint that does not allow a penetration, see Figure D.6, which means that $x \geq 0$. The function can be defined as follows (D.13), where k represents the stiffness of the spring. The base functional can be defined as $f(x) = \frac{1}{2}k(x+1)^2$. Taking $k = 2$.

⁴Not to be confused with the *dual Lagrange multiplier* presented on this work.

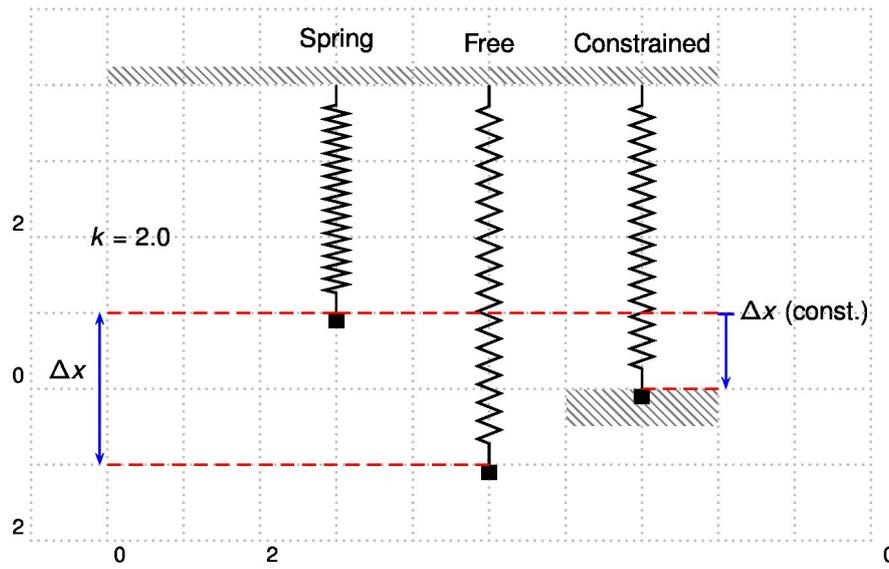


Figure D.6: Simplified contact problem within spring and a wall

(D.13)
$$\begin{cases} f(x) = \frac{1}{2}k(x + 1)^2 \\ \text{Subjected to } x \geq 0 \end{cases}$$

D.7.1.1 Penalty method

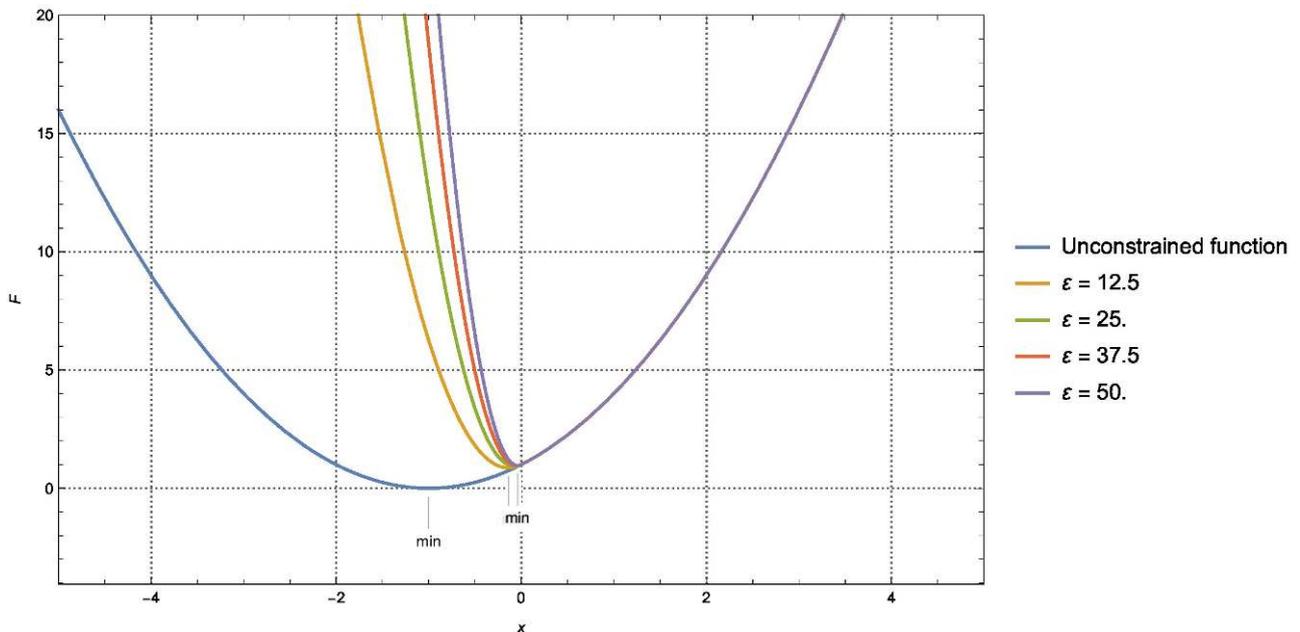


Figure D.7: Solution for a SDOF solved with PM, comparing different values of ϵ

This can be formulated as follows for PM.

(D.14)
$$f_p(x) = \frac{1}{2}k(x + 1)^2 + \frac{\epsilon}{2} (\max \{0, x\})^2$$

Taking (D.14) and applying (D.2) iteratively with a NR we can solve the problem. On Figure D.7 we can appreciate the solutions obtained for different values of ϵ . It can be seen as ϵ grows the solution comes closer to the actual

Iteration	x	f_p
1	1	4
2	-1	10^6
3	-2×10^{-6}	8.40465×10^{-11}

Table D.2: Convergence on f_p with **PM** for (D.13)

solution of the problem. This problem takes in total three **NL** iterations to converge starting with $x = 1$ for a $\varepsilon = 10^6$, see Table D.2, and even with this value the constraint is not completely satisfied.

D.7.1.2 Lagrange multiplier method

Now we can formulate the corresponding *Lagrangian* of (D.13) for the **LMM**. The (D.15) shows the proper potential of our problem with **LMM**. In this example we will only show one of the possible alternatives in order to compute the λ constraint, the next section D.7.2. Non-linear spring contact problem with wall will show the three possible alternatives available when computing the **LMM** contribution.

(D.15)
$$\mathcal{L}_\lambda(x) = \frac{1}{2}k(x + 1)^2 + \min \{0, \lambda\} x$$

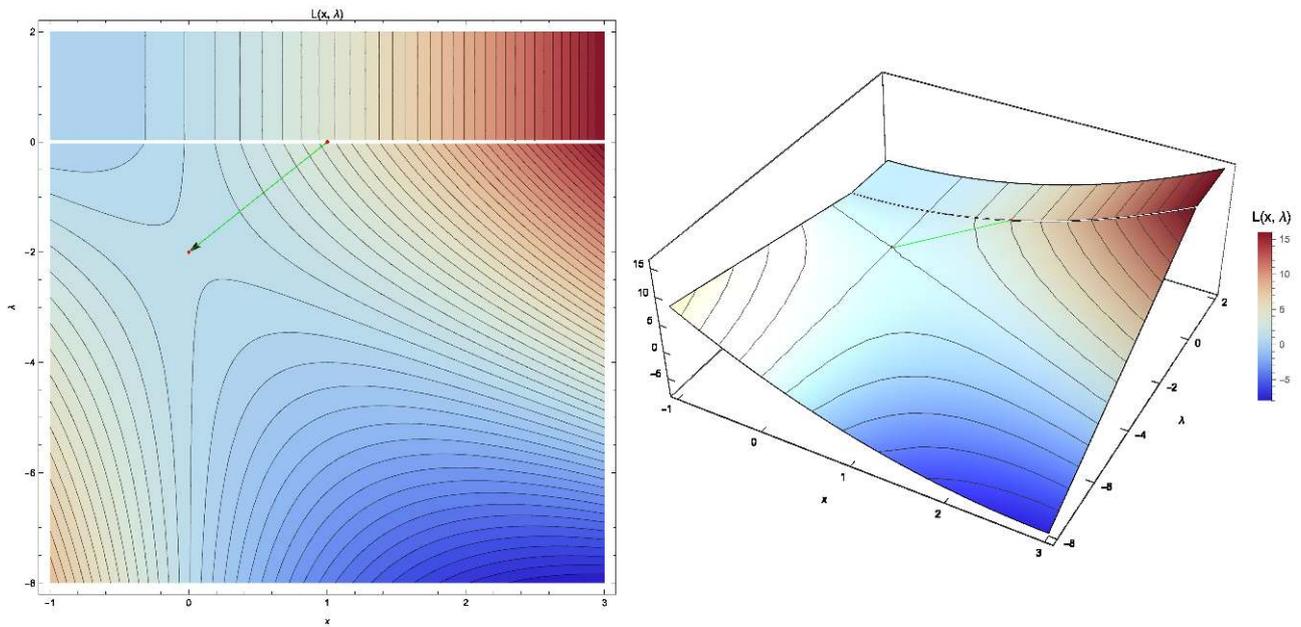


Figure D.8: Solution for a **SDOF** solved with **LMM**

Iteration	x	λ	\mathcal{L}_λ
1	1	0	4.12311
2	0	-2	0

Table D.3: Convergence on \mathcal{L}_λ with **LMM** for (D.13)

On the other hand, the convergence of (D.13) with **LMM** is achieved in only 2 iterations, see Table D.3, reaching additionally the exact solution and satisfying exactly the constraint imposed. On the Figure D.8 we can see the full space of solutions $x - \lambda$, including the respective constraint and appreciate the convergence in only two **NL** iterations.

D.7.2 Non-linear spring contact problem with wall

The following example[PhDYas11; BookYas13], again as seen on Figure D.6, is a simple unidimensional constrained optimisation problem, which can show us the influence of each one of the methodologies when solving these kinds of

problems. In this case the base functional is replaced as (D.16). Additionally we will consider the auxiliary variable $x = u + 1$, this will help us to simplify the some expressions.

$$(D.16) \quad f(x) = \frac{1}{4}x^4 = \frac{1}{4}(u + 1)^4$$

D.7.2.1 Initial wall

We start with a simple case where the wall is fixed to its initial position, then having as constraint (D.17).

$$(D.17) \quad \text{Subject to: } u \leq 0 \text{ or } x \leq 1$$

D.7.2.1.1 Lagrange multiplier method :

Here we can analyse each one of the strategies we can consider for the **LMM**, as presented in D.3.2.Formulation. In summary, the three options consisted on check the positivity/negativity of the constraint (g), the λ or a combination of both former methods.

D.7.2.1.1.1 Active set 1 :

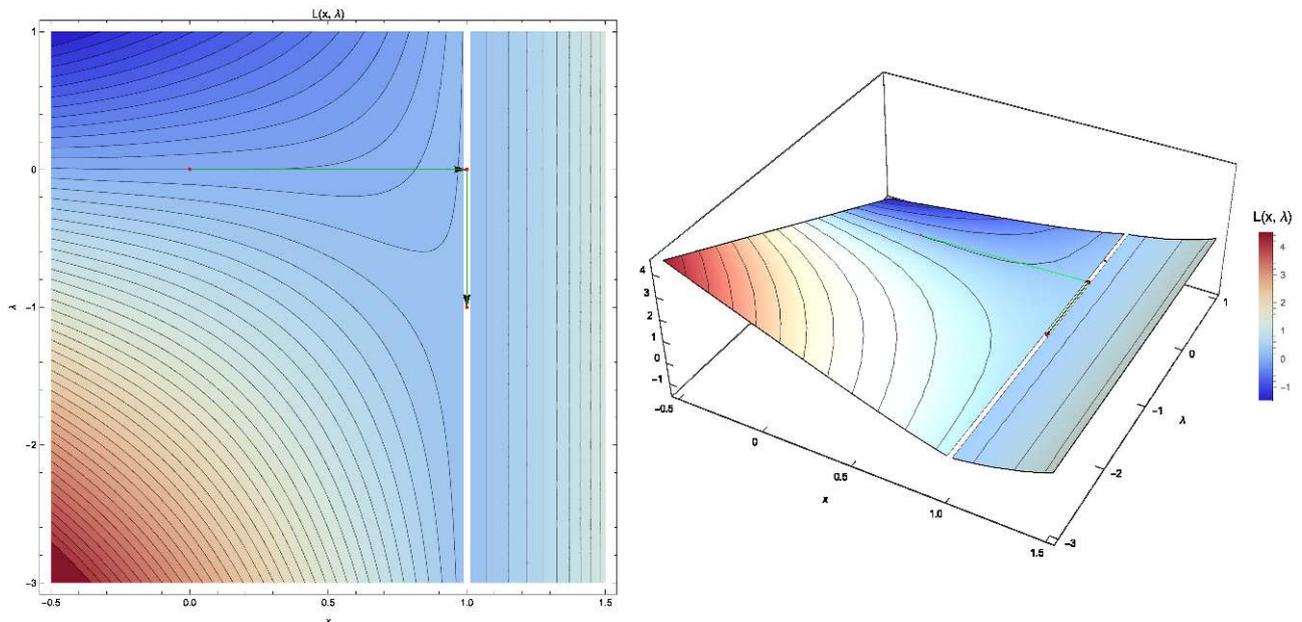


Figure D.9: Solution for a **NL SDOF** solved with **LMM**. Active set 1

As stated on D.3.2.Formulation, the first strategy consists on checks g , as seen on (D.18). The convergence of this method is shown on Table D.4, this is presented graphically on the Figure D.9, where the arrows represent each iteration.

$$(D.18a) \quad \mathcal{L}(x, \lambda) = \frac{1}{4}x^4 + \lambda u, \quad x \leq 1$$

Expressed with the *Macaulay* bracket ($\langle \rangle$)

$$(D.18b) \quad \mathcal{L}(x, \lambda) = \frac{1}{4}x^4 - \lambda \langle -u \rangle$$

Iteration	x	λ	Error
1	0.0	0.0	1.0
2	1.0	0.0	1.0
3	1.0	1.0	2.2204510^{-16}

Table D.4: First active set strategy convergence on \mathcal{L}_λ with **LMM** for (D.16)

D.7.2.1.1.2 Active set 2 :

In this second case, what is checked is the negativity of λ (D.19). For this case, still quite simple, the **NL** iterations give the same result, see Table D.5, which coincides with Table D.4. The convergence path is presented on Figure D.10.

(D.19a)
$$\mathcal{L}(x, \lambda) = \frac{1}{4}x^4 + \lambda u, \quad \lambda \leq 0$$

Expressed with the *Macaulay* bracket ($\langle \cdot \rangle$)

(D.19b)
$$\mathcal{L}(x, \lambda) = \frac{1}{4}x^4 - \langle -\lambda \rangle u$$

Iteration	x	λ	Error
1	0.0	0.0	1.0
2	1.0	0.0	1.0
3	1.0	1.0	2.2204510^{-16}

Table D.5: Second active set strategy convergence on \mathcal{L}_λ with **LMM** for (D.16)

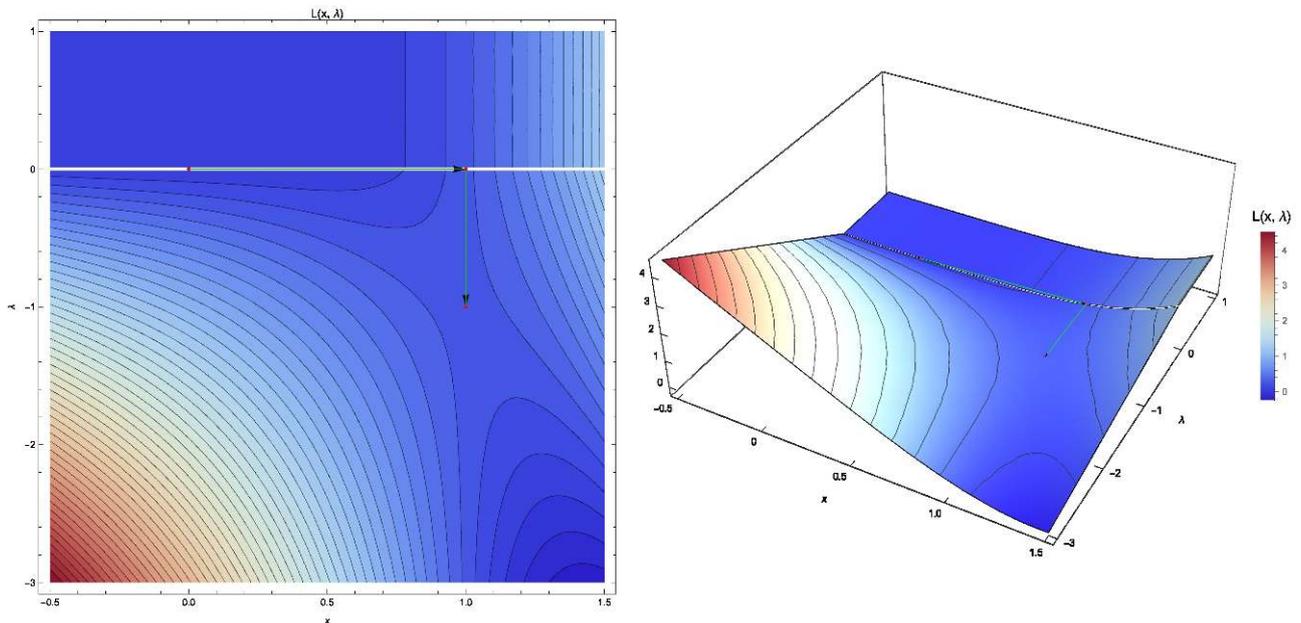


Figure D.10: Solution for a **NL SDOF** solved with **LMM**. Active set 2

D.7.2.1.1.3 Active set 3 :

In the last active set case, which checks simultaneously both g and λ (D.20). Again, for this simple case, the **NL** iterations give the same convergence path, see Table D.6, which coincides with Table D.4 and Table D.5. The path coincides, see Figure D.11.

(D.20a)
$$\mathcal{L}(x, \lambda) = \frac{1}{4}x^4 + \lambda u, \quad \lambda \leq 0 \quad x \leq 1$$

Expressed with the *Macaulay* bracket ($\langle \cdot \rangle$)

$$(D.20b) \quad \mathcal{L}(x, \lambda) = \frac{1}{4}x^4 - \langle -\lambda \rangle u - \lambda \langle -u \rangle + \langle -\lambda \rangle \langle -u \rangle$$

Iteration	x	λ	Error
1	0.0	0.0	1.0
2	1.0	0.0	1.0
3	1.0	1.0	2.2204510^{-16}

Table D.6: Third active set strategy convergence on \mathcal{L}_λ with **LMM** for (D.16)

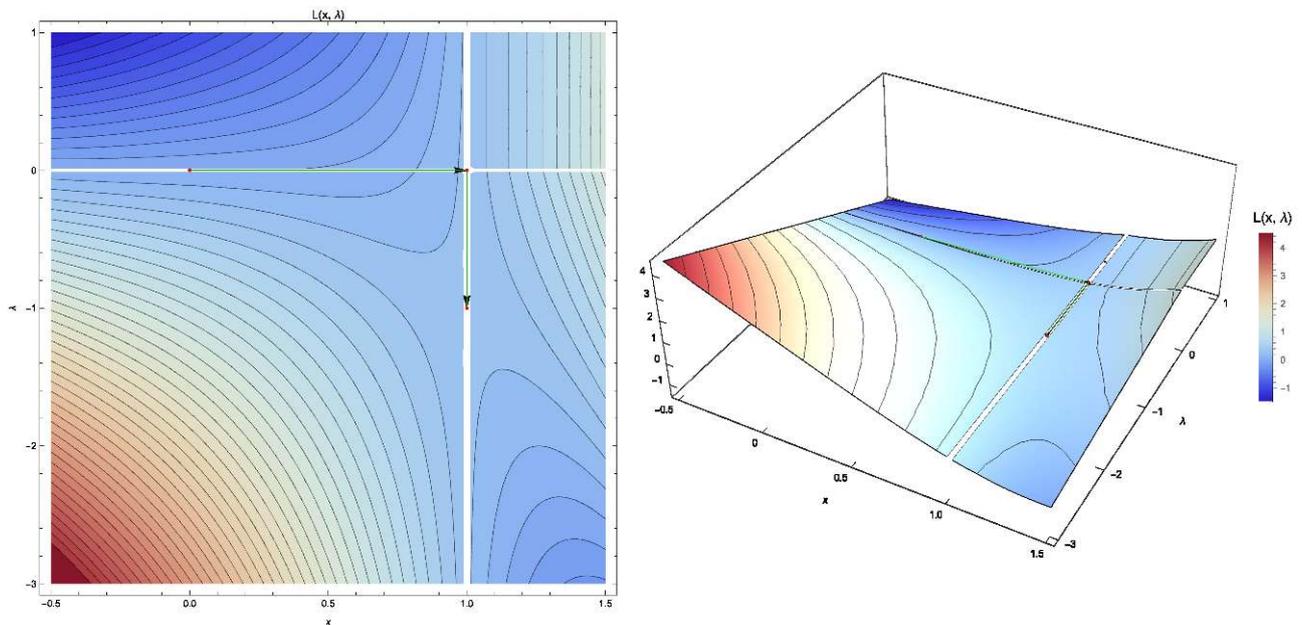


Figure D.11: Solution for a **NL SDOF** solved with **LMM**. Active set 3

D.7.2.2 Moved wall

In order to make the problem more difficult and appreciate better the differences across methodologies, we will partially move the wall which defines our constraint. This example is taken from *Yastrebov*[PhDYas11], where other moving configurations are presented apart of the following.

$$(D.21) \quad \text{Subject to: } u + 0.9 \leq 0.0$$

Moving the wall $0.9m$, the resulting constraint will be (D.21). The resulting case is significantly more complex, and as it will be shown in the next sections, the convergence rate changes from one method to others, some of them even diverging the solution.

D.7.2.2.1 Penalty method :

In the case of the **PM** no tuning is required and the convergence is obtained in a straightforward manner. The only thing that affects the solution is ε value. On Figure D.12 different solutions are presented for different values of ε , the trend to the exact solution can be appreciated.

For a value of $\varepsilon = 10^6$ the correct solution can be achieved, see Table D.7. In here it can be appreciated that on the iteration 7 we sudden increase on the error, this is due to entering on the penalised region ($u > 0.9$). In any case the number of **NL** iteration needed is quite high, 9 iterations, for a **SDOF** problem.

Iteration	u	Error
1	0.0	1.0
2	-0.333333	0.296296
3	-0.555556	0.0877915
4	-0.703704	0.0260123
5	-0.802469	0.00770735
6	-0.868313	0.00228366
7	-0.912209	12208.5
8	-0.9	0.0000410749
9	-0.9	1.7180710^{-12}

Table D.7: Convergence with **PM** for (D.21)

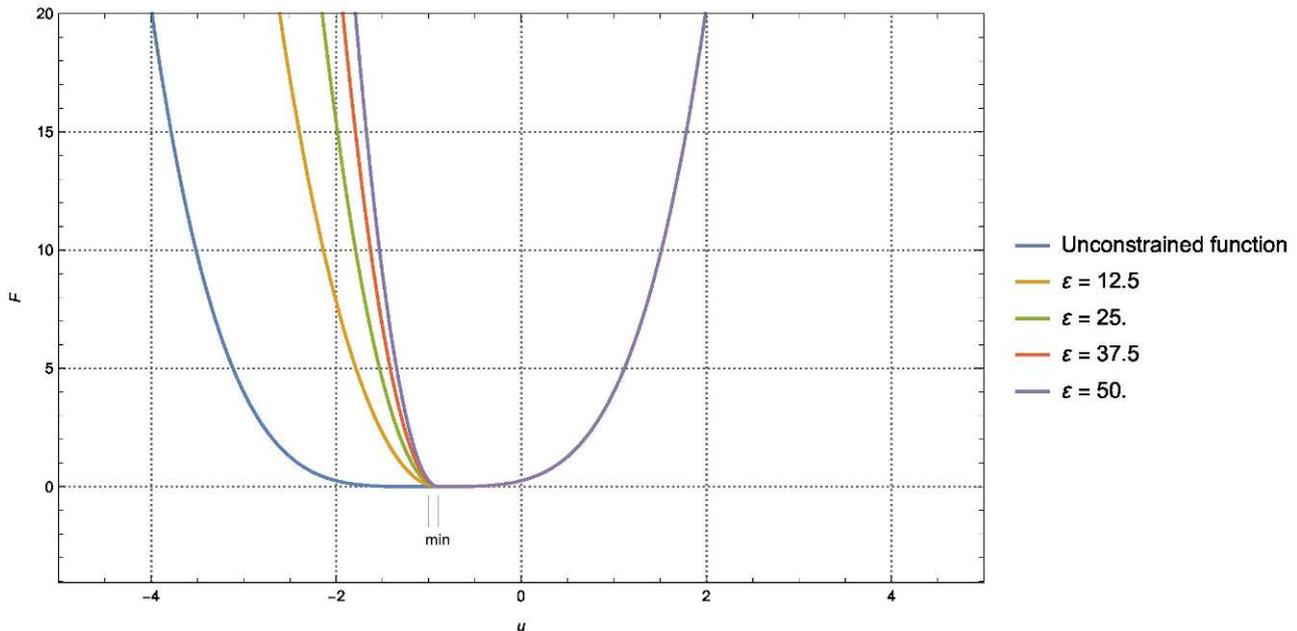


Figure D.12: Solution for a **NL SDOF** problem solved with **PM**, comparing different values of ϵ . Moved wall

D.7.2.2.2 Lagrange multiplier method :

Following the same reasoning from [D.7.2.1.1.Lagrange multiplier method](#), and applying it for (D.21). On this section the \mathcal{L} to be derived are not detailed, as it follows the same principle. The exact solution of the problem is $u = -0.9$ and $\lambda = -0.001$.

D.7.2.2.2.1 Active set 1 :

Iteration	u	λ	Error
1	$-\frac{1}{3}$	0	$\frac{8}{27}$
2	-0.555556	0	0.0877915
3	-0.703704	0	0.0260123
4	-0.802469	0	0.00770735
5	-0.868313	0	0.00228366
6	-0.912209	0	0.0122272
7	-0.9	-0.000958925	0.000041075
8	-0.9	-0.001	2.168410^{-19}

Table D.8: First active set strategy convergence on \mathcal{L}_λ with **LMM** for (D.16) with moved wall

The Table [D.8](#) presents the convergence path that is also represented on [Figure D.13](#). Compared with the unmoved wall, the problem requires a higher number of iterations in order to converge, and it basically moves across

the u axis, but at the end it finally converges.

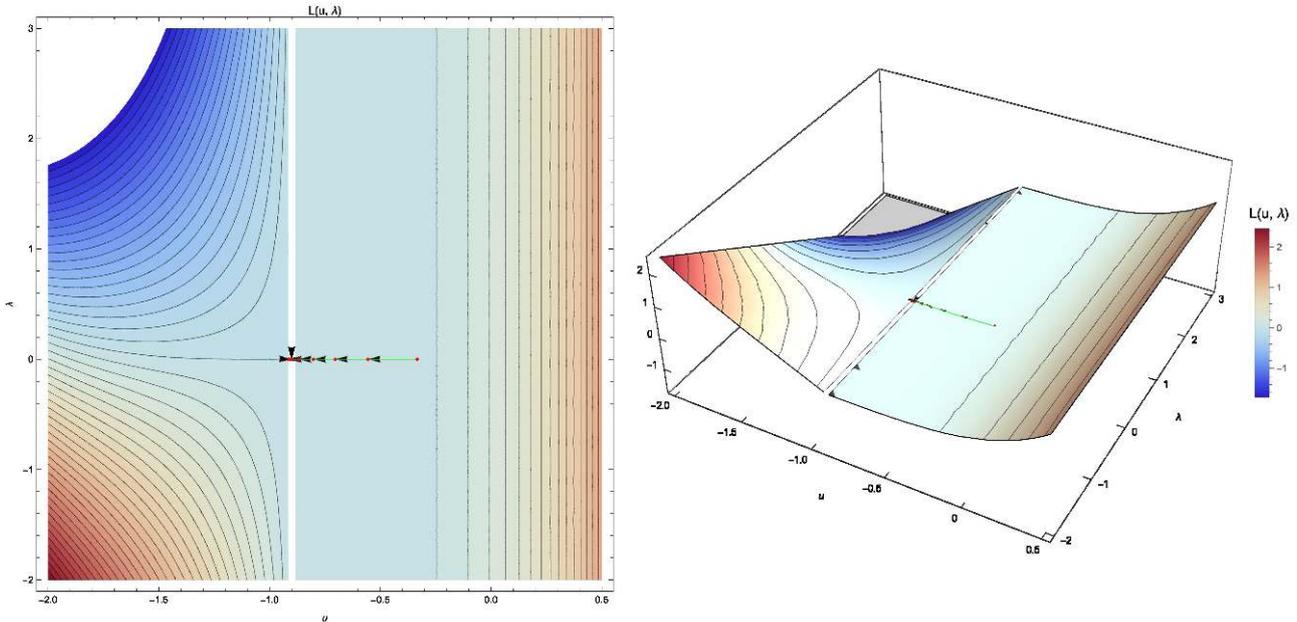


Figure D.13: Solution for a **NL SDOF** solved with **LMM**. Active set 1. Moved wall

D.7.2.2.2 Active set 2 :

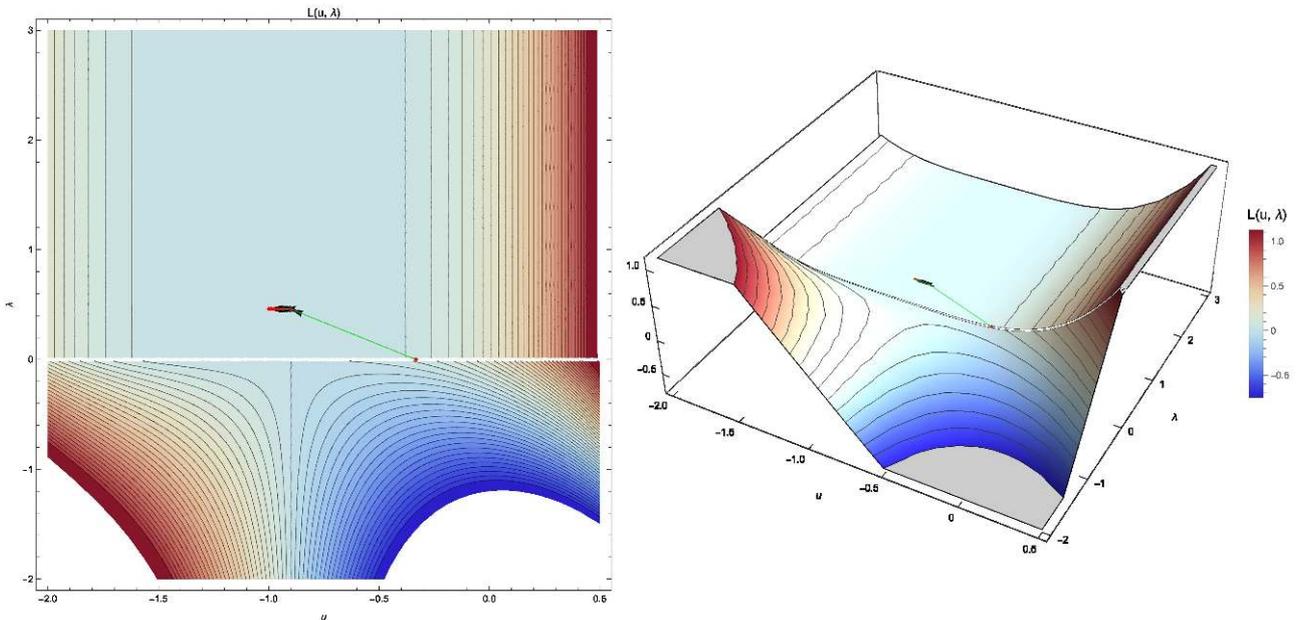


Figure D.14: Solution for a **NL SDOF** solved with **LMM**. Active set 2. Moved wall

On this case the problem doesn't even converge to the right solution, as can be seen on Table D.9, and from Figure D.14 we can notice as it doesn't move arrived certain point. If we extend the number of **NL** iterations to 20 we don't get the proper solution either, the **NL** loop remains around the same values.

On Figure D.14 another relevant point is the fact that the functional is almost flat, explaining part of the difficulty coming from this active set strategy, it is not possible to find a local *minima* on these cases.

In the last active set we will see as this problematic is no more. Other initial conditions can be tried for the \mathcal{L} , but the idea is to compare the presented methods across them, and showing the problematic presented on D.3.2. Formulation, where we introduced that the second active set converges faster than the first one, but in case we are far from the solution we can deal with the divergence on the **NR** iteration.

Iteration	u	λ	Error
1	$-\frac{1}{3}$	0	0.639455
2	-0.9	0.459259	0.001
3	-0.933333	0.459259	0.000296296
4	-0.955556	0.459259	0.0000877915
5	-0.97037	0.459259	0.0000260123
6	-0.980247	0.459259	$7.70735 \cdot 10^{-6}$
7	-0.986831	0.459259	$2.28366 \cdot 10^{-6}$
8	-0.991221	0.459259	$6.76639 \cdot 10^{-7}$
9	-0.994147	0.459259	$2.00486 \cdot 10^{-7}$
10	-0.996098	0.459259	$5.94032 \cdot 10^{-8}$

Table D.9: Second active set strategy convergence on \mathcal{L}_λ with **LMM** for (D.16) with moved wall

D.7.2.2.2.3 Active set 3 :

On the case of the third active set strategy on Table D.10 we have exactly the same convergence path as Table D.8, despite of that the graphical representation differs, Figure D.15.

Iteration	u	λ	Error
1	$-\frac{1}{3}$	0	$\frac{8}{27}$
2	-0.555556	0	0.0877915
3	-0.703704	0	0.0260123
4	-0.802469	0	0.00770735
5	-0.868313	0	0.00228366
6	-0.912209	0	0.0122272
7	-0.9	-0.000958925	0.000041075
8	-0.9	-0.001	$2.168410 \cdot 10^{-19}$

Table D.10: Third active set strategy convergence on \mathcal{L}_λ with **LMM** for (D.16) with moved wall

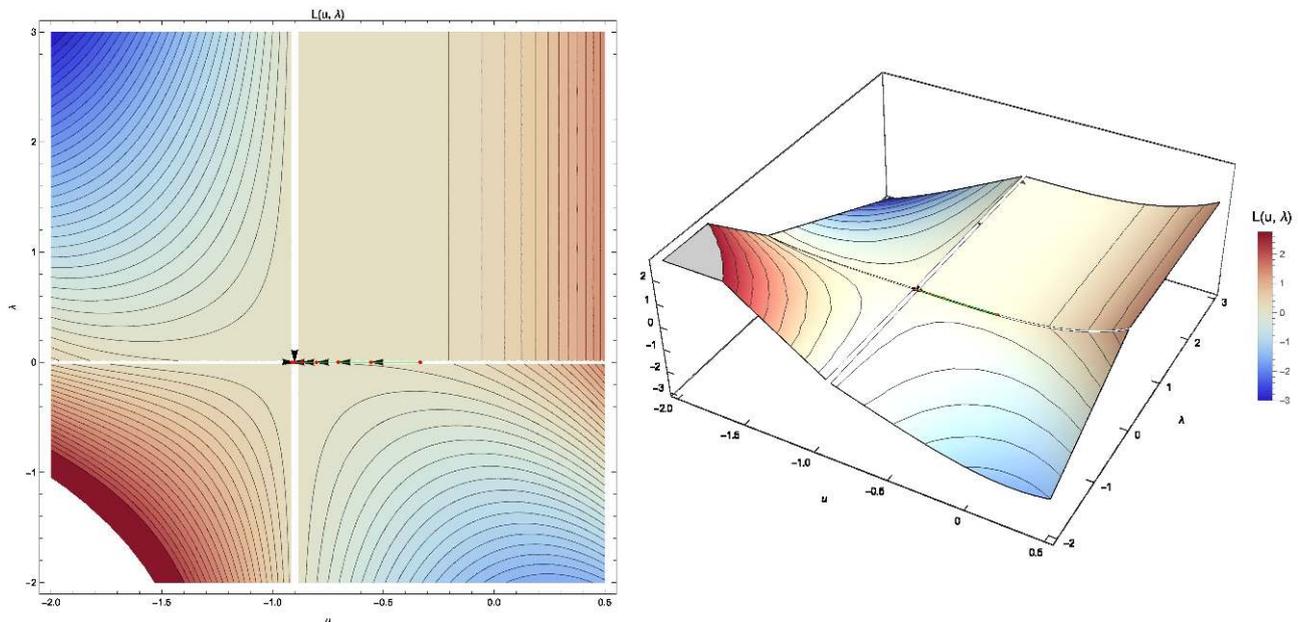


Figure D.15: Solution for a **NL SDOF** solved with **LMM**. Active set 3. Moved wall

We can infer in a simplified manner that the third active set strategy will behave as the best option of the two other alternative strategies, despite this is not always true as already stated on D.3.2.Formulation. The following section will introduce the **ALM** resolution, so it will be possible to compare with the presented here for **LMM**, which in addition to the convergence of the problem improves the **conditioning** of the system of equations.

D.7.2.2.3 Augmented Lagrange multiplier method :

The equation (D.22) shows the resulting \mathcal{L} for the **ALM** method for this problem.

$$(D.22) \quad \mathcal{L}_{\bar{\lambda}}(u, \lambda) = \begin{cases} \frac{1}{2}k(u+1)^4 + \lambda(u+0.9) + \frac{1}{2}\varepsilon(u+0.9)^2, & \lambda + \varepsilon(u+0.9) \leq 0 \\ \frac{1}{2}k(u+1)^4 - \frac{1}{2\varepsilon}\lambda^2 & \lambda + \varepsilon(u+0.9) > 0 \end{cases}$$

We will see like the different values of ε give us a similar convergence rate, but the **condition number** (κ) of the **LHS** varies in function of ε . In order to properly prove that we need the definition of κ (D.23), the proper definition of κ defines it as the ratio of the largest to smallest **singular value** in the singular value decomposition of a matrix, this means it requires to compute the **SVD** of the **LHS**, which is an expensive operation and complex. We can simplify and estimate κ as the ratio between the largest to smallest **eigenvalue**, which are not necessarily coincident.

$$(D.23) \quad \kappa(\mathbf{A}) \approx \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

In order to do so, we need then to define the **LHS** from (D.22), considering (D.2), we can deduce (D.24), recalling the augmented *Lagrange* multiplier as $\bar{\lambda} = \lambda + \varepsilon(u+0.9)$.

$$(D.24a) \quad \mathbf{RHS}(u, \lambda) = \delta \mathcal{L}_{\bar{\lambda}}(u, \lambda) = \begin{cases} \begin{bmatrix} 2k(u+1)^3 + \bar{\lambda} \\ u+0.9 \end{bmatrix}^T \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix} = 0, & \bar{\lambda} \leq 0 \text{ (Active)} \\ \begin{bmatrix} 2k(u+1)^3 \\ -\frac{\lambda}{\varepsilon} \end{bmatrix}^T \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix} = 0, & \bar{\lambda} > 0 \text{ (Inactive)} \end{cases}$$

$$(D.24b) \quad \mathbf{LHS}(x) = \Delta \delta \mathcal{L}_{\bar{\lambda}}(u, \lambda) = \begin{cases} \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix}^T \begin{bmatrix} 6k(u+1)^2 + \varepsilon & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix}, & \bar{\lambda} \leq 0 \text{ (Active)} \\ \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix}^T \begin{bmatrix} 6k(u+1)^2 & 0 \\ 0 & -\frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix}, & \bar{\lambda} > 0 \text{ (Inactive)} \end{cases}$$

From (D.24b) we can calculate κ , in the case of **inactive set** is trivial as the **LHS** is diagonal (D.25a). On this resulting expression, κ grows linearly with ε . For the **active set** case, we obtain (D.25b) as expression defining κ . On this case the condition number of the **LHS** grows quadratically to the ε . In conclusion, for a high ε , in comparison with the stiffness of the problem, κ become very high which means we have an **ill-conditioned** problem, affecting the precision of the solution and its convergence; but, on the other hand, the energy functional becomes smooth, as we will state next.

$$(D.25a) \quad \kappa_{inactive}(\mathbf{LHS}) \approx \begin{cases} 6k\varepsilon u^2, & \frac{1}{\varepsilon} \leq 6ku^2 \\ \frac{1}{6ku^2\varepsilon}, & \frac{1}{\varepsilon} > 6ku^2 \end{cases} \quad \text{assuming } \frac{1}{\varepsilon} \leq 6ku^2 \text{ then } \sim k\varepsilon$$

$$(D.25b) \quad \kappa_{active}(\mathbf{LHS}) \approx \frac{1}{2} (6k(u+1)^2 + \varepsilon) \left(\sqrt{(6k(u+1)^2 + \varepsilon)^2 + 4} + (6k(u+1)^2 + \varepsilon) \right) + 1 \sim (k + \varepsilon)^2$$

Another important point to highlight is that all the graphical representation of the \mathcal{L} shows a continuous field without flat areas (See Figures D.16, D.17, D.18, D.19). This already gives us an idea how the method already solves some problems presented on the previous section (D.7.2.2.2. Lagrange multiplier method) with the **LMM**, avoiding the need of choose between different active set strategies.

D.7.2.2.3.1 $\varepsilon = 0.5$:

Our first case with $\varepsilon = 0.5$ show a very similar convergence path to the one presented on [D.7.2.2.2.1.Active set 1](#), [Table D.11](#), but the functional from [Figure D.16](#) is smoother than the one from [Figure D.13](#).

Iteration	\mathbf{x}	λ	Error
1	$-\frac{1}{3}$	0	0.296296
2	-0.555556	0	0.0877915
3	-0.703704	0	0.0260123
4	-0.802469	0	0.00770735
5	-0.868313	0	0.00228366
6	-0.912209	0	0.0122272
7	-0.9	-0.000958925	0.000041075
8	-0.9	-0.001	4.33681^{-19}

Table D.11: Convergence on \mathcal{L}_{λ} with **ALM** for [\(D.22\)](#) with $\varepsilon = 0.5$

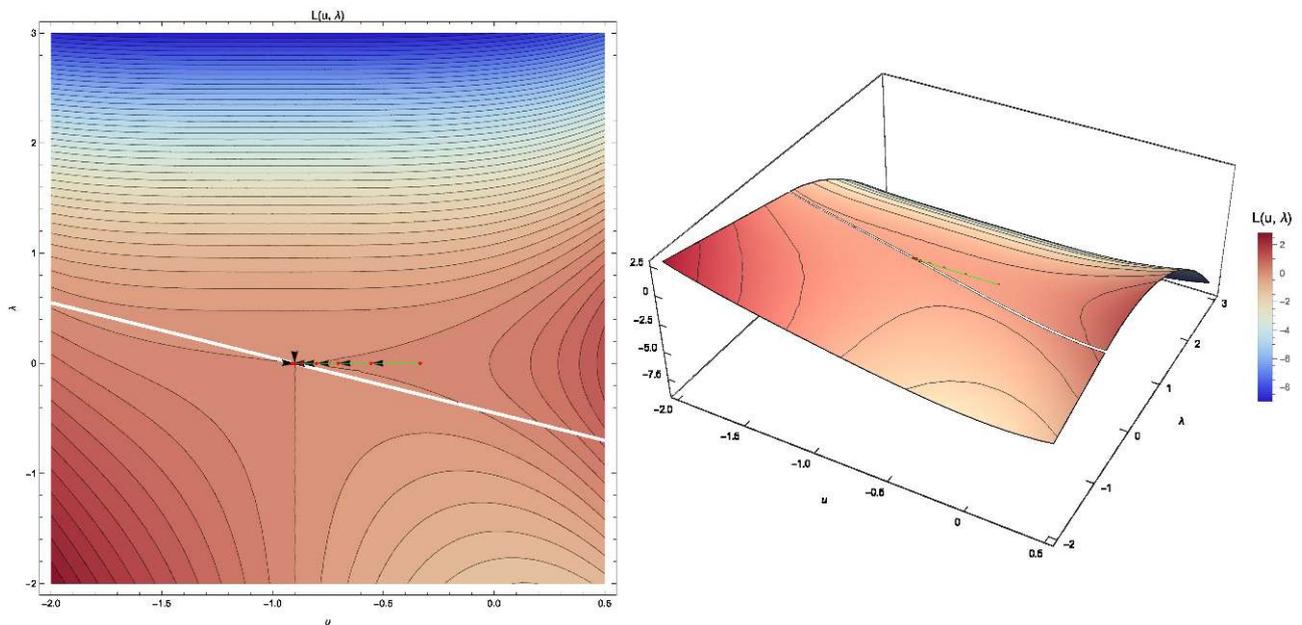


Figure D.16: Solution for a **NL SDOF** solved with **ALM**. $\varepsilon = 0.5$. Moved wall

D.7.2.2.3.2 $\varepsilon = 1.0$:

On this second case with $\varepsilon = 1.0$ the [Table D.12](#) presents practically the same path as [Table D.11](#), with the exception that once the solution enters in the inactive region the error changes slightly. The [Figure D.17](#) is even smoother than [Figure D.16](#).

Iteration	\mathbf{x}	λ	Error
1	$-\frac{1}{3}$	0	0.296296
2	-0.555556	0	0.0877915
3	-0.703704	0	0.0260123
4	-0.802469	0	0.00770735
5	-0.868313	0	0.00228366
6	-0.912209	0	0.0167938
7	-0.9	-0.000958925	0.000041075
8	-0.9	-0.001	1.0842^{-18}

Table D.12: Convergence on \mathcal{L}_{λ} with **ALM** for [\(D.22\)](#) with $\varepsilon = 1.0$

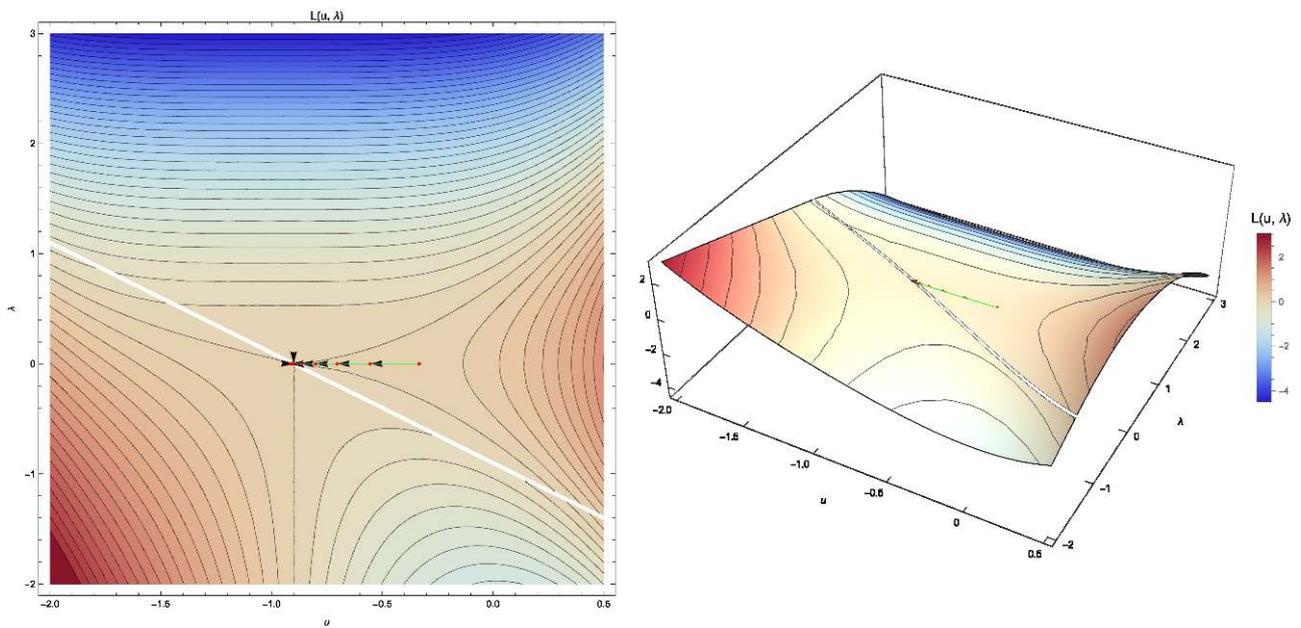


Figure D.17: Solution for a **NL SDOF** solved with **ALM**. $\varepsilon = 1.0$. Moved wall

D.7.2.2.3.3 $\varepsilon = 5.0$:

Iteration	x	λ	Error
1	$-\frac{1}{3}$	0	0.296296
2	-0.555556	0	0.0877915
3	-0.703704	0	0.0260123
4	-0.802469	0	0.00770735
5	-0.868313	0	0.00228366
6	-0.912209	0	0.061588
7	-0.9	-0.000958925	0.000041075
8	-0.9	-0.001	$2.60209 \cdot 10^{-18}$

Table D.13: Convergence on $\mathcal{L}_{\bar{\lambda}}$ with **ALM** for (D.22) with $\varepsilon = 5.0$

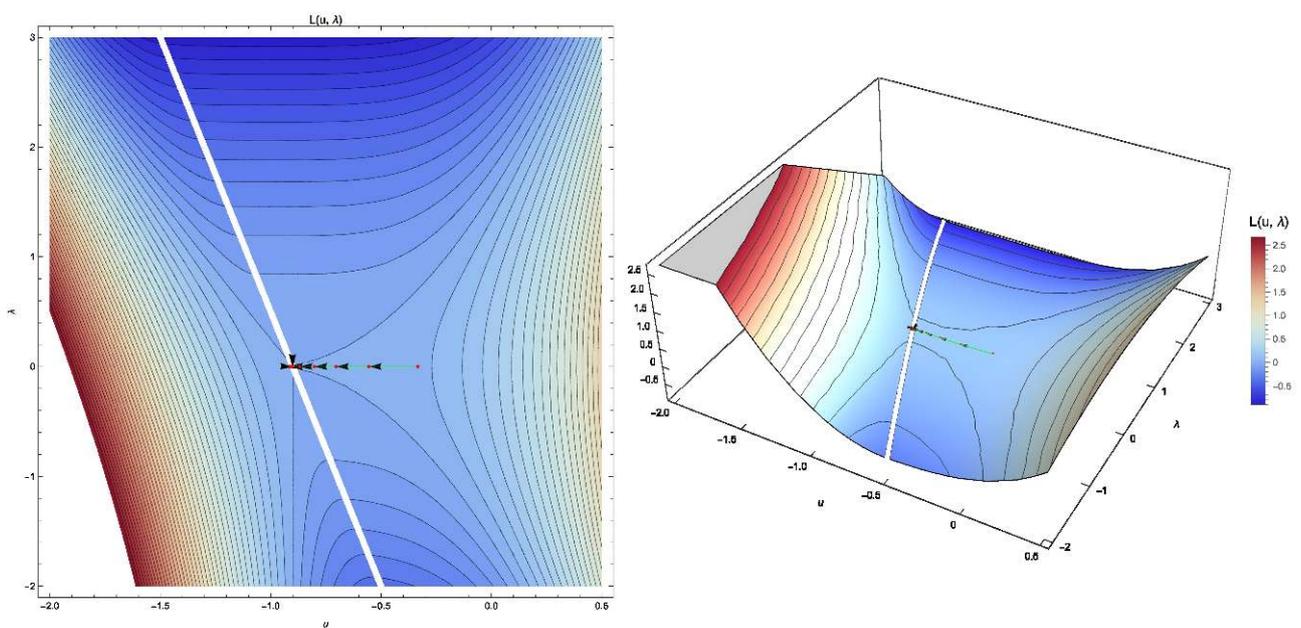


Figure D.18: Solution for a **NL SDOF** solved with **ALM**. $\varepsilon = 5.0$. Moved wall

For the case with $\varepsilon = 5.0$, we have again a very similar **NL** loop, Table D.13, converging again in 8 **NL** iterations. The only value that changes again is the error when crossing the constraint threshold. On the other side, the Figure D.13 shows a significant aspect on the \mathcal{L} representation.

D.7.2.2.3.4 $\varepsilon = 10.0$:

With $\varepsilon = 10.0$ we have again a very similar **NL** convergence, Table D.14, but the graphical representation of \mathcal{L} fathom the change of aspect on \mathcal{L} already seen in the previous case, see Figure D.19.

Iteration	\mathbf{x}	λ	Error
1	$-\frac{1}{3}$	0	0.296296
2	-0.555556	0	0.0877915
3	-0.703704	0	0.0260123
4	-0.802469	0	0.00770735
5	-0.868313	0	0.00228366
6	-0.912209	0	0.122021
7	-0.9	-0.000958925	0.000041075
8	-0.9	-0.001	$3.90313 \cdot 10^{-18}$

Table D.14: Convergence on \mathcal{L}_λ with **ALM** for (D.22) with $\varepsilon = 10.0$

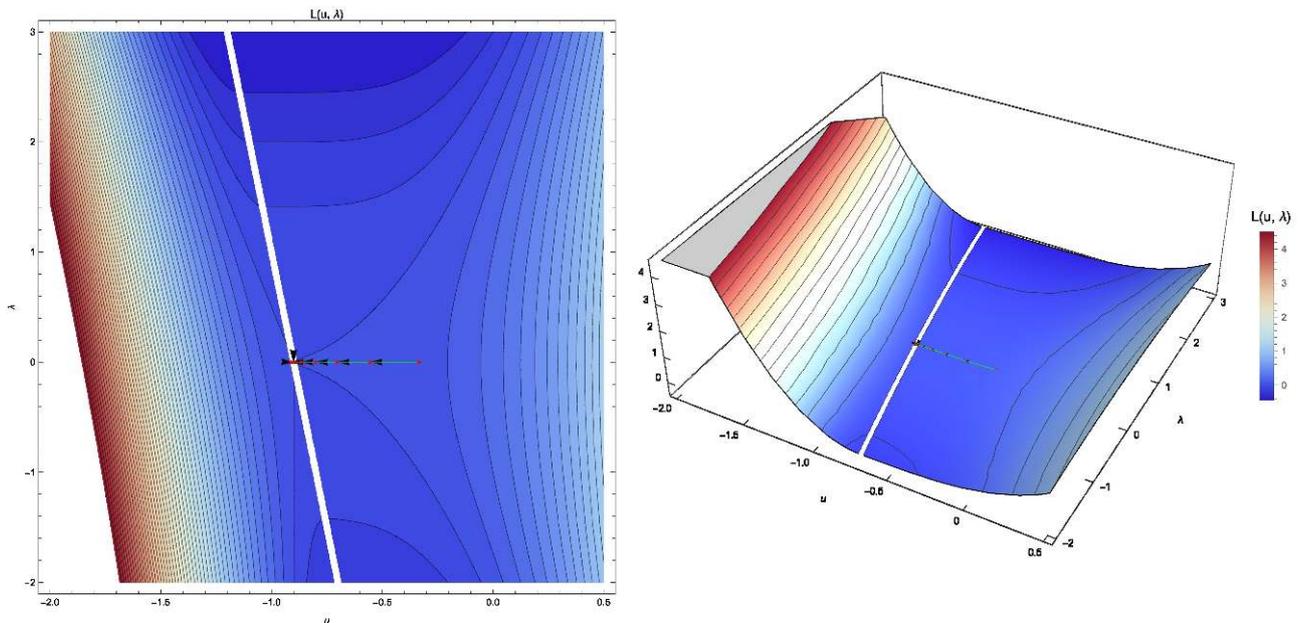


Figure D.19: Solution for a **NL SDOF** solved with **ALM**. $\varepsilon = 10.0$. Moved wall

D.7.3 Over-constrained optimisation problem

The next example[Onl], and last one on this section, concerns an optimisation problem over-constrained, that will be shown as an example of problem that can be solved with the **PM**, but not with the most advanced methodologies as the **LMM** or the **ALM**. The problem to solve of interest is shown in (D.26).

$$\begin{aligned}
 \text{Minimize} \quad & f(\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 7)^2 \\
 \text{subject to} \quad & g_1(\mathbf{x}) = -3x_1 - 2x_2 + 6 \leq 0 \\
 \text{(D.26)} \quad & g_2(\mathbf{x}) = -x_1 + x_2 - 3 \leq 0 \\
 & g_3(\mathbf{x}) = x_1 + x_2 - 7 \leq 0 \\
 & g_4(\mathbf{x}) = \frac{2}{3}x_1 - x_2 - \frac{4}{3} \leq 0
 \end{aligned}$$

This function and constrains can be seen visually on Figure D.20, here we can appreciate the constraints, as the limits on the functional representation.

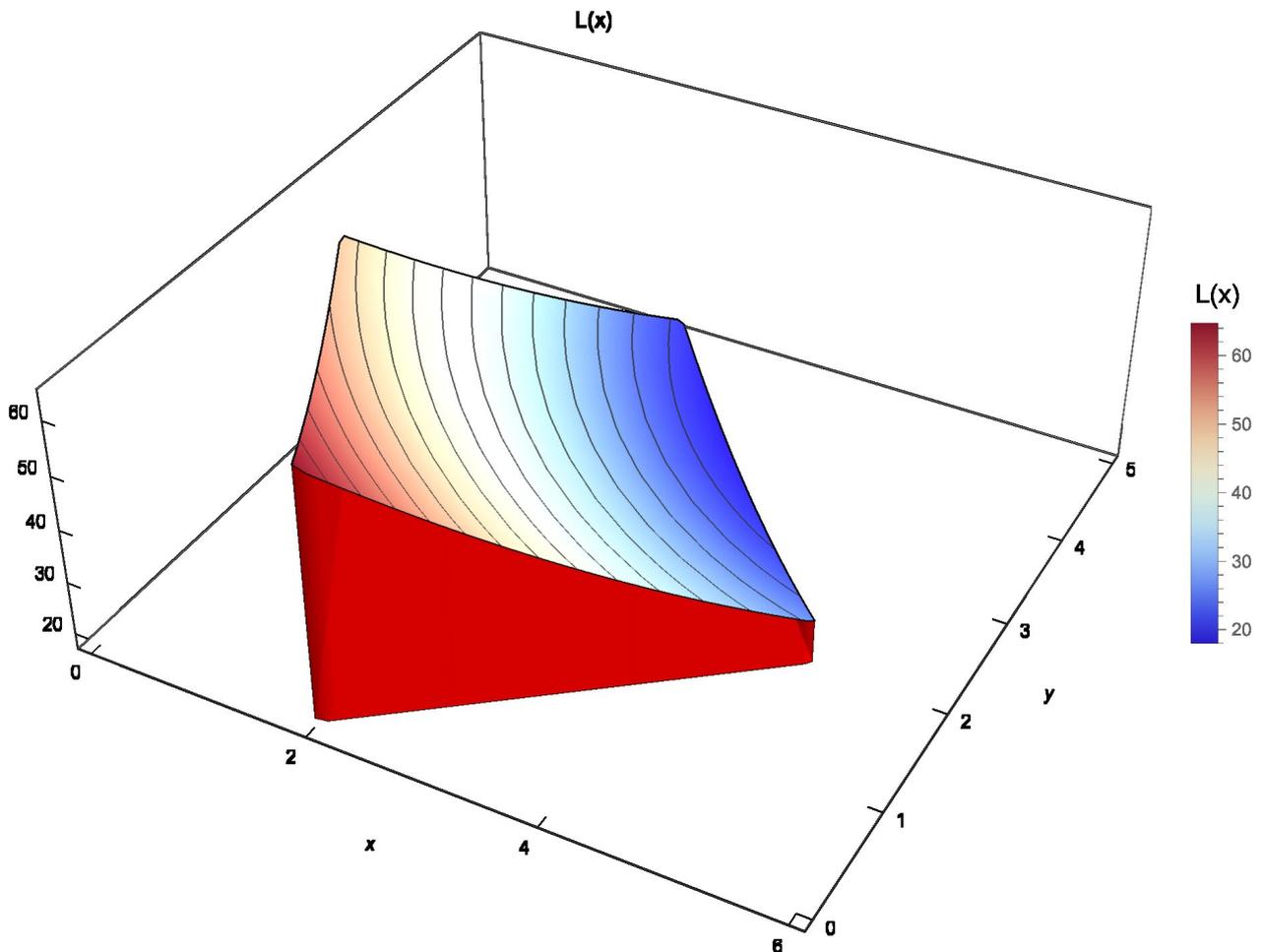


Figure D.20: Graphic representation of (D.26)

This equation becomes (D.27), where ε is the penalty parameter.

$$\begin{aligned}
 (D.27) \quad f_p(\varepsilon, \mathbf{x}) &= (x_1 - 6)^2 + (x_2 - 7)^2 \\
 &+ \varepsilon (\max \{0, -3x_1 - 2x_2 + 6\})^2 \\
 &+ \varepsilon (\max \{0, -x_1 + x_2 - 3\})^2 \\
 &+ \varepsilon (\max \{0, x_1 + x_2 - 7\})^2 \\
 &+ \varepsilon \left(\max \left\{ 0, \frac{2}{3}x_1 - x_2 - \frac{4}{3} \right\} \right)^2
 \end{aligned}$$

Considering (D.2) and solving applying a **NR** iterative resolution we can obtain the correct solution in 4 steps starting from $x = 0, y = 0$ with $\varepsilon = 10^6$, as can be seen on Table D.15.

Iteration	x	y	f_p
1	0	0	$2\sqrt{468000384000085}$
2	0	3	14.4222
3	6	7	1.69706×10^7
4	3	4	1.97134×10^{-9}

Table D.15: Convergence on f_p with penalty method for D.26

On the other hand, if we try to solve this problem using a **LMM**, we will find problems to solve due to the

overconstraint of the problem. If we formulate (D.26) considering for example the second active set strategy from (D.5e), with $x = 6$, $y = 7$, $\lambda_i = 0$ for $i = 1, 2, 3, 4$, the resulting **LHS** will be (D.28). This is a matrix is rank 2, being a 6×6 matrix, which means that the system of equations is not solvable.

$$(D.28) \quad \mathbf{LHS} = \begin{pmatrix} -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Same happens considering the first active set strategy, which is more robust. The underlying problem is the overconstrained problem, which we have already shown, is not a problem for a **PM**. This is a small example that shows that **LMM** despite being a powerful and generic technique it is not applicable for all cases.

Bibliography

Books

- [BookAkh88] *The calculus of variations*. Naum Ilich Akhiezer. CRC Press. 1988.
- [BookAnd01] *From Convexity to Nonconvexity*. Anders Klarbring (auth.) R. P. Gilbert P. D. Panagiotopoulos P. M. Pardalos (eds.) Lars-Erik Andersson. Springer US. 1st ed. 2001.
- [BookCA01] *Introduction to finite element methods*. Felippa C.A. 2001.
- [BookIK08] *Lagrange multiplier approach to variational problems and applications*. Kazufumi Ito and Karl Kunisch. Siam. 2008.
- [BookKO88] *Contact problems in elasticity: a study of variational inequalities and finite element methods*. Noboru Kikuchi and John Tinsley Oden. siam. 1988.
- [BookLei04] *Dynamics and Bifurcations of Non-Smooth Mechanical Systems*. Professor Dr. Henk Nijmeijer (auth.) Dr. Remco I. Leine. Springer-Verlag Berlin Heidelberg. 1st ed. 2004.
- [BookWri06] *Computational Contact Mechanics*. Peter Wriggers. Springer. 2nd. 2006.
- [BookYas13] *Numerical Methods in Contact Mechanics*. Vladislav A. Yastrebov. Wiley-ISTE. 1st ed. 2013.
- [BookZTF14] *The Finite Element Method for Solid and Structural Mechanics*. O. C. Zienkiewicz, Robert L. Taylor, and David Fox. Butterworth-Heinemann. 7th ed. 2014.

Articles

- [ArtAC91] “A mixed formulation for frictional contact problems prone to Newton like solution methods”. Pierre Alart and Alain Curnier. In: *Computer methods in applied mechanics and engineering*. No. 3, Vol. 92, 1991, pp. 353–375. Elsevier. DOI: [10.1016/0045-7825\(91\)90022-x](https://doi.org/10.1016/0045-7825(91)90022-x).
- [ArtAHU58] “Studies in linear and non-linear programming”. Kenneth J Arrow, Leonid Hurwicz, and Hirofumi Uzawa. In: . 1958, Cambridge Univ. Press.
- [ArtAla88] “Multiplicateurs “augmentés” et méthode de Newton généralisée pour contact avec frottement”. P Alart. In: *Report, Document LMA-DME-EPFL, Lausanne*. 1988,
- [ArtAla97] “Méthode de Newton généralisée en mécanique du contact”. Pierre Alart. In: *Journal de mathématiques pures et appliquées*. No. 1, Vol. 76, 1997, pp. 83–108. Elsevier.
- [ArtAS58] “Gradient methods for constrained maxima, with weakened assumptions”. KENNETH J Arrow and ROBERT M Solow. In: *Studies in linear and nonlinear programming*. 1958, pp. 166–176. Stanford University Press, Stanford, CA.
- [ArtBMP] “The adapted augmented Lagrangian method: a new method for the resolution of the mechanical frictional contact problem”. Philippe Busetta, Daniel Marceau, and Jean-Philippe Ponthot. In: *Computational Mechanics*. No. 2, , pp. 259–275. DOI: [10.1007/s00466-011-0644-z](https://doi.org/10.1007/s00466-011-0644-z).
- [ArtHes69] “Multiplier and gradient methods”. Magnus R Hestenes. In: *Journal of optimization theory and applications*. No. 5, Vol. 4, 1969, pp. 303–320. Springer. DOI: [10.1007/bf00927673](https://doi.org/10.1007/bf00927673).

- [ArtNit71] “Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind”. Joachim Nitsche. In: . No. 1, Vol. 36, 1971, pp. 9–15.
- [ArtPow69] “A method for nonlinear constraints in minimization problems”. Michael JD Powell. In: *Optimization*. 1969, pp. 283–298. Academic Press.
- [ArtQS99] “A survey of some nonsmooth equations and smoothing Newton methods”. Liqun Qi and Defeng Sun. In: . 1999, pp. 121–146. Springer. DOI: [10.1007/978-1-4613-3285-5_7](https://doi.org/10.1007/978-1-4613-3285-5_7).
- [ArtRoc73a] “The multiplier method of Hestenes and Powell applied to convex programming”. R Tyrrell Rockafellar. In: *Journal of Optimization Theory and applications*. No. 6, Vol. 12, 1973, pp. 555–562. Springer. DOI: [10.1007/bf00934777](https://doi.org/10.1007/bf00934777).
- [ArtRoc73b] “A dual approach to solving nonlinear programming problems by unconstrained optimization”. R Tyrrell Rockafellar. In: *Mathematical programming*. No. 1, Vol. 5, 1973, pp. 354–373. Springer. DOI: [10.1007/bf01580138](https://doi.org/10.1007/bf01580138).
- [ArtSWT85a] “A perturbed Lagrangian formulation for the finite element solution of contact problems”. Juan C Simo, Peter Wriggers, and Robert L Taylor. In: *Computer methods in applied mechanics and engineering*. No. 2, Vol. 50, 1985, pp. 163–180. Elsevier. DOI: [10.1016/0045-7825\(85\)90088-x](https://doi.org/10.1016/0045-7825(85)90088-x).
- [ArtSWT85b] “A perturbed Lagrangian formulation for the finite element solution of contact problems”. Juan C. Simo, Peter Wriggers, and Robert L. Taylor. In: *Computer Methods in Applied Mechanics and Engineering*. No. 2, Vol. 50, 1985, pp. 163–180. DOI: [10.1016/0045-7825\(85\)90088-x](https://doi.org/10.1016/0045-7825(85)90088-x).

Ph.D.'s thesis

- [PhDBus] “Modélisation et résolution du problème de contact mécanique et son application dans un contexte multiphysique”. Philippe Bussetta.
- [PhDHet14] “The Bipenalty Method for Explicit Structural Dynamics”. Jack Hetherington. 2014.
- [PhDYas11] “Computational contact mechanics: geometry, detection and numerical techniques”. Vladislav A. Yastrebov. 2011.

Online resources

- [Onl] *Algorithms for Constrained Optimization*. URL: https://www.me.utexas.edu/~jensen/ORMM/supplements/units/nlp_methods/const_opt.pdf.
- [Onl] *Dualisation of the boundary conditions*. URL: https://www.code-aster.org/V2/doc/v11/en/man_r/r3/r3.03.01.pdf.
- [Onl] *Wikipedia. Uzawa iteration*. URL: https://en.wikipedia.org/wiki/Uzawa_iteration.
- [OnlLag] *Wikipedia. Lagrange multiplier*. URL: http://en.wikipedia.org/wiki/Lagrange_multiplier.

Appendix E

Mortar mapper

“Projections are just bullshit.
They're just guesses.”

Jason Fried
(Software entrepreneur)

E.1 Introduction

This appendix introduces the mapper developed using the *Mortar* formulation [ArtDB06; ArtND08]. The goal of a map is the distribution of the variable of interest from the *origin mesh* to the *destination mesh*, see Figure E.1a. This method provides an exact decomposition of the target mesh (Figure E.1b). With this, the procedure of mapping has the advantage of conserving, in a weak sense at least, the important physical quantities such as mass, energy, etc. Additionally this method satisfies the equilibrium in a weak manner, being then consistent with the **FEM** formulation.

The mapping is used on this work in order to compute a consistent gap between meshes during the search process, see Algorithm 8, in order to discard *poor* contact candidates. *Poor* candidates can be defined as the contact candidates that will not activate during the whole **NL** iteration process. The inclusion of these candidates may be difficult the convergence of the **active-set** strategy, then increasing the number of **NL** iterations computed during the **NR** strategy.

Algorithm 8 Consistent gap computation

```

1: procedure CONSISTENT GAP COMPUTATION
2:   Reset auxiliar values for nodal coordinates on origin mesh  $x_{aux}$ 
3:   for all  $node \in OriginMesh_{nodes}$  do
4:      $x_{aux} = x_{node}$ 
5:   for all  $node \in DestinationMesh_{nodes}$  do
6:      $x_{aux} = \mathbf{0}$ 
7:   Map  $x_{aux}$  from  $OriginMesh_{nodes} \rightarrow DestinationMesh_{nodes}$ 
8:   for all  $node \in DestinationMesh_{nodes}$  do
9:     From  $node$  get the normal ( $\mathbf{n}$ )
10:     $g_{consistent} = -\mathbf{n} \cdot (x_{node} - x_{aux})$ 

```

The reason why the very same *Mortar* formulation has been taken into consideration are, first the developments are shared between the two frameworks; second the consistency between the results obtained between the system of equations and the search process; but primarily the **quality** of the results. On this last point, the results that can be

obtained with this method are between the best options available[ArtBBP]. Precisely on *Bussetta*[ArtBBP] the three types of data transfer procedures are discussed:

- **Element Transfer Method (ELM)**: The simplest one consists on map the value depending only on the location of the destination mesh on the origin mesh. Methods on this category we can find the *nearest element*[MasBuc17] or *nearest neighbour*[MasBuc17] mappers.
- Procedures based on the weak conservation of the data field. The presented method (sometimes called **ELM**) would we encompassed on this category. Other techniques defined on this category can be **Finite Volume Transfer Method (FVTM)**[ArtBBP], which consists on decide the meshes on **Finite Volumes (FV)** elements and the transfer is computed between **FV** meshes.
- These are the procedures based on reconstruction of the field in the neighbourhood of this point. On this category we can find the modern technique **Moving Least Square (MLS)**[ArtCho+05; ArtLIC07]. Additionally on [ArtYu+08; ArtHC90; ArtDHC92] we can find more examples of this kind of methodology. This concept is considered on the method presented on E.2.3. *Discontinuous meshes mapping*, where the nodes close to the discontinuity are taken into consideration to reconstruct the field.

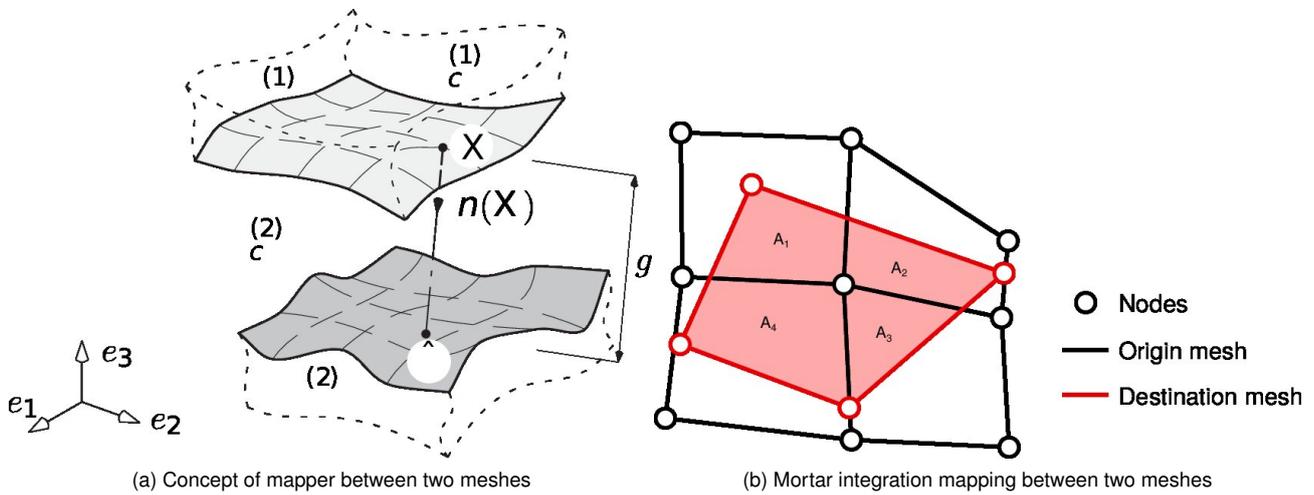


Figure E.1: Mapping concepts

The *Mortar* formulation employed, based on the concept of dual[ArtWoh02; ArtWoh11] **LM**, which allows to evaluate in a simplified manner the data transference between meshes. This method has been already presented on the corresponding contact sections (4.2.3.4. **DLMM**). Instead of solving a whole assembled system of the transfer operators, the dual **LM** allows to assemble it in a diagonal manner, which can be solved in a trivial manner. There are alternatives methodologies for using the *Mortar* formulation without solving directly the system of equations, such the **Jacobian-Free Newton Krylov (JFNK)** approach by *Hansen*[ArtHan11].

E.2 Theory

E.2.1 General mapping theory

Based on the concepts introduced on the work of *Ute*[MasIsr06] and *Jaiman*[ArtJai+06], we can start defining the consistent transfer operation between meshes (E.1).

$$(E.1a) \quad \mathbf{f} = \mathbf{M}_c \mathbf{u}$$

Where the matrices and vectors can be defined as:

$$(E.1b) \quad \begin{cases} \mathbf{M}_c = \int_{\Gamma_d} N_d^i N_d^j d\Gamma_d \\ \mathbf{f} = \int_{\Gamma_d} N_d^i N_o^j u_o^j d\Gamma_d \end{cases}$$

And $\mathbf{u} = \mathbf{u}_d^i$ the nodal values of the destination mesh. The index i represents the i^{th} NL iteration. We can define the **RHS** and **LHS** of our respective system of equations.

$$(E.1c) \quad \begin{cases} \mathbf{RHS} = \mathbf{f} - \mathbf{M}_c \mathbf{u}_i \\ \mathbf{LHS} = -\frac{\partial \mathbf{RHS}}{\partial \mathbf{u}} = \mathbf{M}_c \end{cases}$$

With the **RHS** and the **LHS** we can obtain the update of the update of \mathbf{u} , $\Delta \mathbf{u}$.

$$(E.1d) \quad \mathbf{u}_{i+1} = \mathbf{u}_i + \Delta \mathbf{u}$$

Where:

$$(E.1e) \quad \Delta \mathbf{u} = \mathbf{LHS}^{-1} \mathbf{RHS} = \mathbf{M}_c^{-1} (\mathbf{f} - \mathbf{M}_c \mathbf{u}_i) = \mathbf{M}_c^{-1} \mathbf{f} - \mathbf{u}_i$$

This methodology implies the whole system integration, and resolution of the whole system of equations. On [E.2.2.Dual Lagrange multiplier mapping](#) we will introduce how with the consideration of dual *Lagrange* multiplier the \mathbf{M}_c becomes diagonal, \mathbf{M}_D , with the respective advantages.

E.2.2 Dual Lagrange multiplier mapping

E.2.2.1 Introduction

As previously stated, using the *dual Lagrange* multiplier allows to solve the exact system of equations with the inversion of a **diagonal matrix**, trivial operation that can be parallelised. The theory introducing the concept of dual *Lagrange* multiplier was first introduced by *Wohlmuth*[[ArtWoh02](#)], who used as an alternative method for **DDM**. These nonstandard *Lagrange* multiplier spaces provided optimal discretisation schemes and a locally supported basis for the associated constrained mortar spaces.

E.2.2.2 Theory

Algorithm 9 Explicit contribution of the pairs during dual λ mortar mapping

```

1: procedure EXPLICIT CONTRIBUTION OF THE PAIRS
2:   while ( $rhs_{total} > tolerance_{abs}$  and  $ratio > tolerance_{rel}$ ) and  $i < iteration_{max}$  do
3:     Reset global residual norm  $rhs_{total}$ 
4:     Reset auxiliar values for nodal residual  $rhs_{node}$ 
5:     Reset weight areas  $w_{node}$ 
6:     for all  $elem \in DestinationMesh_{elements}$  do
7:       if  $Area_{mortar\ segmentation} > tolerance$  then
8:         Calculate D and M
9:         Calculate local residual (E.2a) and the local weight area (E.2b)
10:    for all  $node \in DestinationMesh_{nodes}$  do
11:      Compute  $u_d = u_d + \frac{rhs_{node}}{w_{node}}$ 
12:      Compute  $rhs_{total} = rhs_{total} + rhs_{node}^2$ 
13:      Compute the residual norm:  $rhs_{total} = \frac{rhs_{total}}{n_{dof}}$ 
14:      if  $i = 0$  then
15:         $rhs_{total0} = rhs_{total}$ 
16:      Compute  $ratio = \frac{rhs_{total}}{rhs_{total0}}$ 
17:       $i = i + 1$ 

```

We address the theory of the *dual Lagrange* multiplier previously presented, [4.3.3.4.1.Dual Lagrange multipliers](#), which means that we can focus on the required operations starting from the theory already presented in [E.2.1.General mapping theory](#), and modify some specific operations with our alternative λ .

On [Algorithm 9](#) we present the solution designed to obtain the explicit contributions of the pairs, for them compute the mapping. From [Algorithm 9](#) we need to define some missing concepts ([E.2](#)). Being known **D** and **M** in every pair of entities, where **D** is assumed to be diagonal as previously stated, then the explicit contribution of errors and weight

areas are (E.2a) and (E.2b) respectively. Where the sub-index *node* refers to the pair entities which share certain nodes.

$$(E.2a) \quad rhs_{node} = \sum_{elem=1}^{N_{node}} \mathbf{M}^{elem} \mathbf{u}_o^{elem} - \mathbf{D}^{elem} \mathbf{u}_d^{elem}$$

$$(E.2b) \quad w_{node} = \sum_{elem=1}^{N_{node}} \mathbf{D}^{elem}(i_{node}, i_{node})$$

E.2.3 Discontinuous meshes mapping

E.2.3.1 Introduction

This is an innovative contribution of this work, which allows to transfer data between two *discontinuous* meshes. We will call a *discontinuous* mesh to the mesh that is not **watertight**, which means that the nodes are not necessarily shared between the entities which conform the mesh. This means that following the metaphor of a container, in case of being filled with water, the water would leak across the gaps between the nodes of the mesh, reason why is called **watertight**.

This development can be used on *embedded* CFD simulations, where the condition of **watertight** mesh is not guaranteed, especially from meshes imported from STL files, which usually undergo these problematic.

E.2.3.2 Theory

Algorithm 10 Explicit contribution of the pairs during dual λ mortar discontinuous mapping

```

1: procedure EXPLICIT CONTRIBUTION OF THE PAIRS ON DISCONTINUOUS MAPPING
2:   Create an inverse database (pairing destination-origin entities and vice-versa)
3:   while ( $rhs_{total} > tolerance_{abs}$  and  $ratio > tolerance_{rel}$ ) and  $i < iteration_{max}$  do
4:     Reset global residual norm  $rhs_{total}$ 
5:     Reset auxiliar values for nodal residual  $rhs_{node}$ 
6:     Reset weight areas  $w_{node}$ 
7:     for all  $elem \in DestinationMesh_{elements}$  do
8:       if  $Area_{mortar\ segmentation} > tolerance$  then
9:         Calculate D and M
10:        Calculate local residual (E.2a) and the local weight area (E.2b)
11:        for all  $node \in elem$  do
12:          Local  $w_{local} = \mathbf{D}(node, node)$  contribution
13:          Local  $h$  equivalent to length of  $elem$ 
14:          for all  $pair_{inv} \in elem_{inverse\ paired}$  do
15:            for all  $node_{aux} \in pair_{inv}$  do
16:              Compute distance  $d$  between  $node$  and  $node_{aux}$ 
17:              Compute local weight area discontinuous contribution (E.3)
18:        for all  $node \in DestinationMesh_{nodes}$  do
19:          Compute  $u_d = u_d + \frac{rhs_{node}}{w_{node}^{discontinuous}}$ 
20:          Compute  $rhs_{total} = rhs_{total} + rhs_{node}^2$ 
21:          Compute the residual norm:  $rhs_{total} = \frac{rhs_{total}}{n_{dot}}$ 
22:          if  $i = 0$  then
23:             $rhs_{total0} = rhs_{total}$ 
24:            Compute  $ratio = \frac{rhs_{total}}{rhs_{total0}}$ 
25:             $i = i + 1$ 

```

This method preserves the properties presented on E.2.2.Dual Lagrange multiplier mapping, which means that the corresponding system of equations keeps the possibility to be solved in a simplified manner. The main

difference between the previous approach and the current one is the need to create an inverse mapping of the interface. This means to create a database which inverts the roles *origin-destination* meshes. Additionally a coefficient of contributions is considered to take into account the contribution of close nodes, this contribution is computed penalising the distance between the destination node and the one evaluated on the discontinuous interface. Algorithm 10 presents the modifications needed on the Algorithm 9 in order to compute the mapping of values for a discontinuous interface.

On (E.3) the additional contribution of the discontinuous interface is specified. Being $k \approx 1.0e - 4$, being k called discontinuous interface factor. This value has been estimated doing several evaluations in different cases with different levels of discontinuity on the interface, being this value the best compromise for the different cases studied.

$$(E.3) \quad w_{node}^{discontinuous} = w_{node} + \frac{w_{local}}{\left(1 + \frac{d}{kh}\right)^2}$$

E.3 Numerical examples

In all the cases presented below we map between two meshes with an *irregular double-curvature*, which makes the problem difficult to map for simpler methods as the *nearest neighbour mapping*. The functions (scalar and vector) we want to map between these two meshes is presented on (E.4). On the scalar case everything depends only on the z coordinate, meanwhile the vector case is equivalent to the direct imposition of the coordinates of the nodes.

The scalar function:

$$(E.4a) \quad f(x, y, z) = z$$

The vector function:

$$(E.4b) \quad \mathbf{f}(x, y, z) = [x \quad y \quad z]$$

E.3.1 Non-matching meshes of triangles

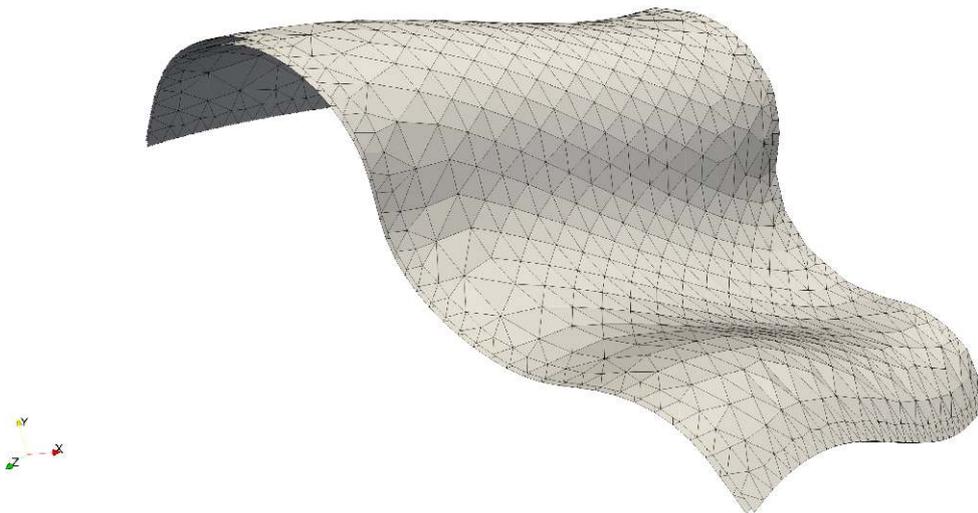
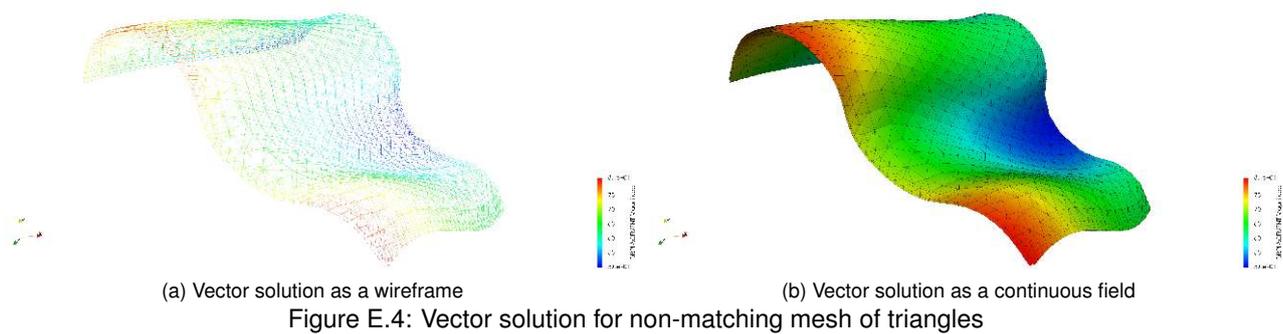
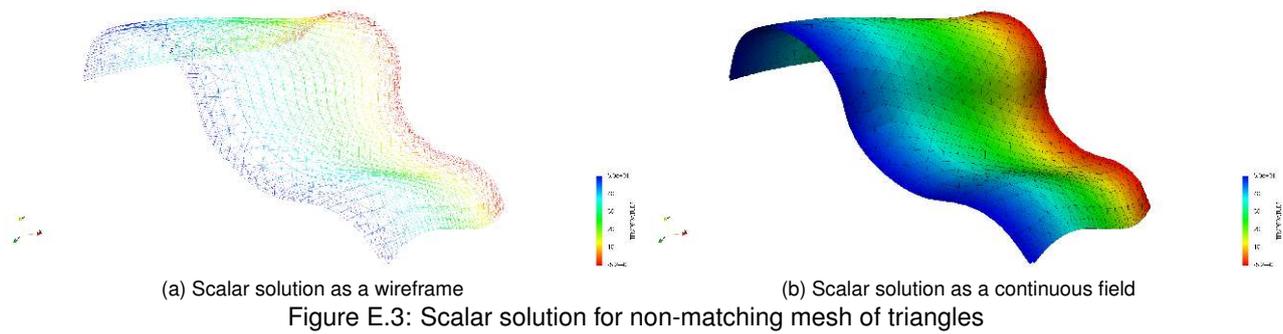


Figure E.2: Non-matching mesh of triangles

The first case from Figure E.2 present us the previously mentioned example with *irregular double-curvature* meshes. On this case, both meshes are conformed by triangular elements. The results obtained for a scalar value is shown in Figure E.9, meanwhile Figure E.10 shows the results for a vector value. Both solutions provide a very good approach of the values, this can be appreciated with particular detail on the wireframe results (Figures E.6a and E.7a), where it is easier to appreciate the resulting values on both meshes.



E.3.2 Non-matching meshes of triangles and quadrilaterals

The case from Figure E.5 instead of being fully meshed on triangles as Figure E.2, or even fully meshed with quadrilaterals, presents both types of geometries. The origin mesh consists on triangular entities, meanwhile the destination mesh is integrated by quadrilateral entities. This case shows how it is possible not only map values across *non-conformant* meshes, but also between different types of geometries, with the additional complexity.

The results obtained on scalar and vector fields (Figures E.9 and E.10) show again a good matching between the origin and destination values. In the same way as the previous example, the wireframe (Figures E.6a and E.7a) and continuous field solutions (Figures E.6b and E.7b) are presented.

E.3.3 Discontinuous meshes

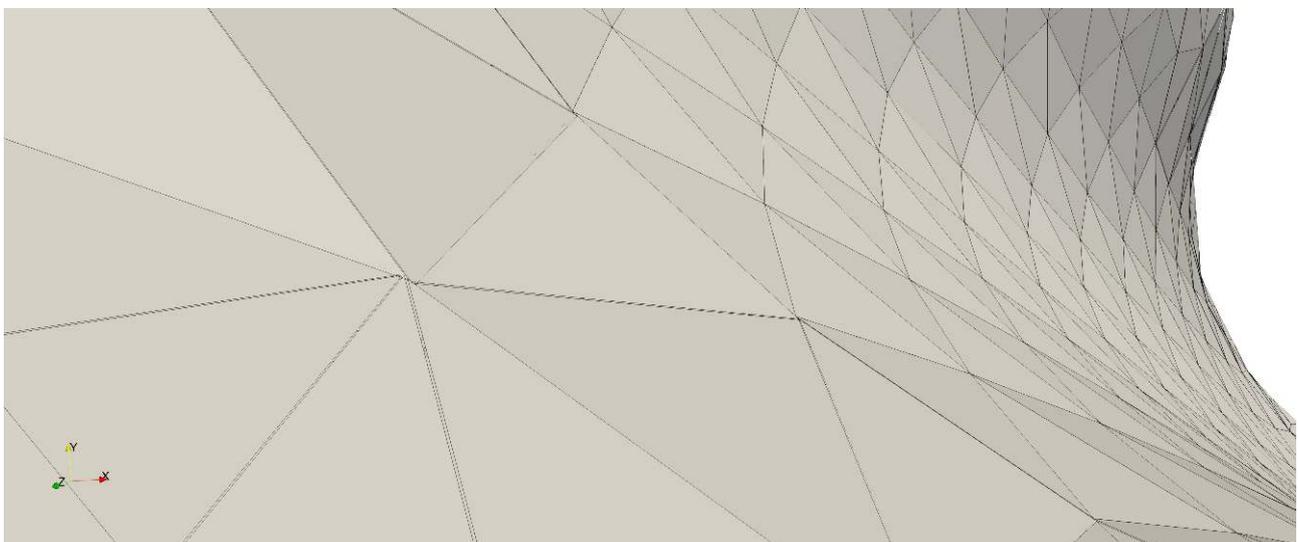


Figure E.8: Detail of the non-matching mesh discontinuous of triangles

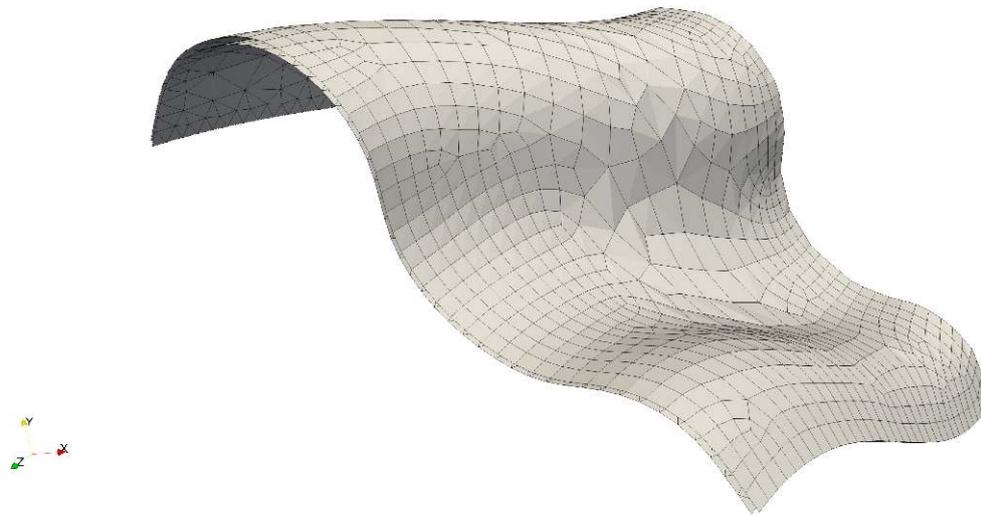
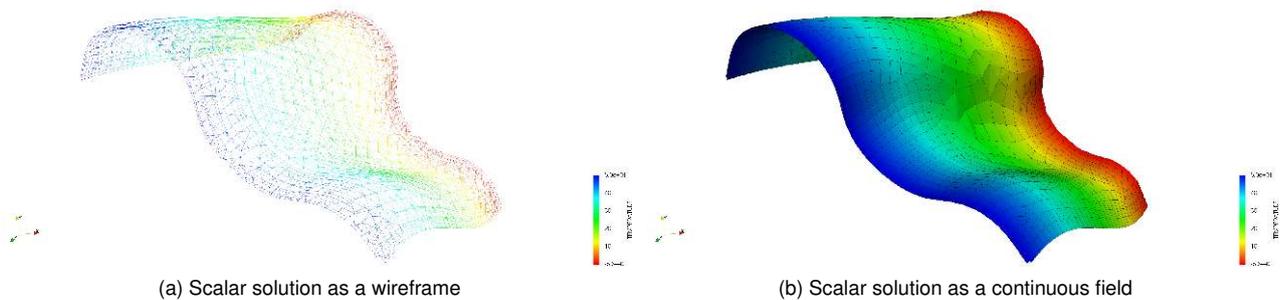


Figure E.5: Non-matching mesh of triangles and quadrilaterals



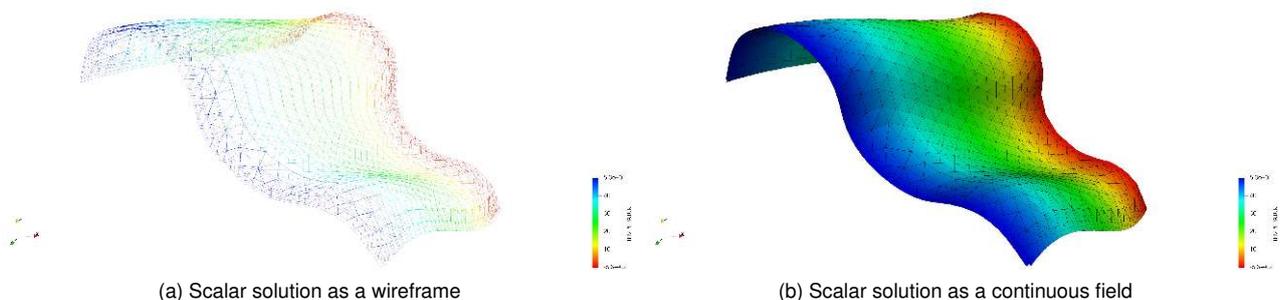
(a) Scalar solution as a wireframe

(b) Scalar solution as a continuous field

Figure E.6: Scalar solution for non-matching mesh of triangles and quadrilaterals

The following case may *a priori* look similar to Figure E.2, that's why Figure E.8 zooms into the detail of the mesh, so the discontinuity of the meshes is appreciated. Due to this the standard mapper cannot be applied, and the method presented on E.2.3. [Discontinuous meshes mapping](#) must be taken into consideration.

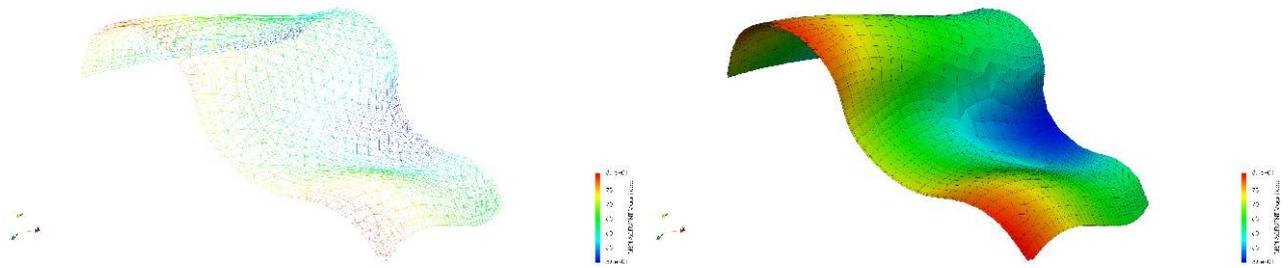
The results obtained are again being of good quality despite of the existence of this **discontinuity** on the mesh, both for scalar and vector fields (Figures E.9 and E.10 respectively).



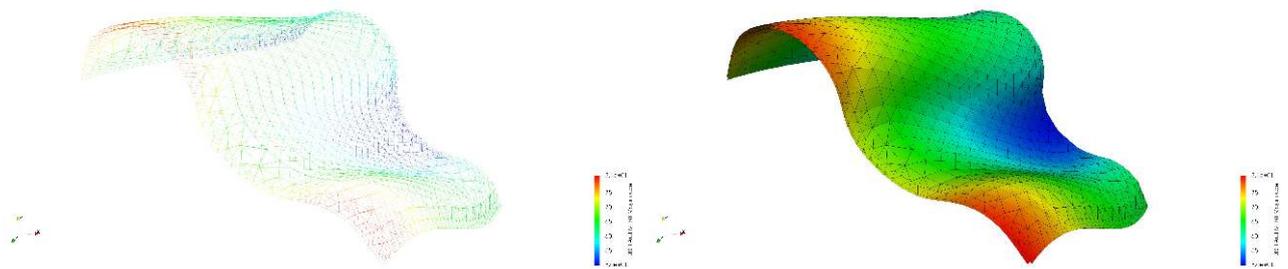
(a) Scalar solution as a wireframe

(b) Scalar solution as a continuous field

Figure E.9: Scalar solution for non-matching discontinuous mesh of triangles



(a) Vector solution as a wireframe
 (b) Vector solution as a continuous field
 Figure E.7: Vector solution for non-matching mesh of triangles and quadrilaterals



(a) Vector solution as a wireframe
 (b) Vector solution as a continuous field
 Figure E.10: Vector solution for non-matching discontinuous mesh of triangles

Bibliography

Articles

- [ArtBBP] “Efficient 3D data transfer operators based on numerical integration”. Philippe Bussetta, Romain Boman, and Jean-Philippe Ponthot. In: *International Journal for Numerical Methods in Engineering*. No. 3-4, , pp. 892–929. Wiley Online Library. DOI: [10.1002/nme.4821](https://doi.org/10.1002/nme.4821).
- [ArtCho+05] “An improved interface element with variable nodes for non-matching finite element meshes”. Young-Sam Cho, Sukky Jun, Seyoung Im, and Hyun-Gyu Kim. In: *Computer Methods in Applied Mechanics and Engineering*. No. 27-29, Vol. 194, 2005, pp. 3022–3046. Elsevier. DOI: [10.1016/j.cma.2004.08.002](https://doi.org/10.1016/j.cma.2004.08.002).
- [ArtDB06] “Information transfer between incompatible finite element meshes: application to coupled thermo-viscoelasticity”. David Dureisseix and Henri Bavestrello. In: *Computer Methods in Applied Mechanics and Engineering*. No. 44-47, Vol. 195, 2006, pp. 6523–6541. Elsevier. DOI: [10.1016/j.cma.2006.02.003](https://doi.org/10.1016/j.cma.2006.02.003).
- [ArtDHC92] “Automatic adaptive remeshing for numerical simulations of metalforming”. M Dyduch, AM Habraken, and Serge Cescotto. In: *Computer Methods in Applied Mechanics and Engineering*. No. 1-3, Vol. 101, 1992, pp. 283–298. Elsevier. DOI: [10.1016/0045-7825\(92\)90026-g](https://doi.org/10.1016/0045-7825(92)90026-g).
- [ArtHan11] “A Jacobian-free Newton Krylov method for mortar-discretized thermomechanical contact problems”. Glen Hansen. In: *Journal of Computational Physics*. No. 17, Vol. 230, 2011, pp. 6546–6562. Elsevier. DOI: [10.1016/j.jcp.2011.04.038](https://doi.org/10.1016/j.jcp.2011.04.038).
- [ArtHC90] “An automatic remeshing technique for finite element simulation of forming processes”. AM Habraken and Serge Cescotto. In: *International Journal for Numerical Methods in Engineering*. No. 8, Vol. 30, 1990, pp. 1503–1525. Wiley Online Library. DOI: [10.1002/nme.1620300811](https://doi.org/10.1002/nme.1620300811).
- [ArtJai+06] “Conservative load transfer along curved fluid–solid interface with non-matching meshes”. Rajeev K Jaiman, Xiangmin Jiao, Philippe H Geubelle, and Eric Loth. In: *Journal of Computational Physics*. No. 1, Vol. 218, 2006, pp. 372–397. Elsevier. DOI: [10.1016/j.jcp.2006.02.016](https://doi.org/10.1016/j.jcp.2006.02.016).
- [ArtLIC07] “Variable-node elements for non-matching meshes by means of MLS (moving least-square) scheme”. Jae Hyuk Lim, Seyoung Im, and Young-Sam Cho. In: *International Journal for Numerical Methods in Engineering*. No. 7, Vol. 72, 2007, pp. 835–857. Wiley Online Library. DOI: [10.1002/nme.1988](https://doi.org/10.1002/nme.1988).
- [ArtND08] “A computational strategy for thermo-poroelastic structures with a time–space interface coupling”. David Néron and David Dureisseix. In: *International Journal for Numerical Methods in Engineering*. No. 9, Vol. 75, 2008, pp. 1053–1084. Wiley Online Library. DOI: [10.1002/nme.2283](https://doi.org/10.1002/nme.2283).
- [ArtWoh02] “A Comparison of Dual Lagrange Multiplier Spaces for Mortar Finite Element Discretizations”. Barbara I. Wohlmuth. In: *ESAIM: Mathematical Modelling and Numerical Analysis*. No. 6, Vol. 36, 2002, pp. 995–1012. EDP Sciences. DOI: [10.1051/m2an:2003002](https://doi.org/10.1051/m2an:2003002).
- [ArtWoh11] “Variationally consistent discretization schemes and numerical algorithms for contact problems”. Barbara Wohlmuth. In: *Acta Numerica*. Vol. 20, 2011, pp. 569–734. DOI: [10.1017/s0962492911000079](https://doi.org/10.1017/s0962492911000079).

- [ArtYu+08] “Three-dimensional RITSS large displacement finite element method for penetration of foundations into soil”. L Yu, J Liu, XJ Kong, and Y Hu. In: *Computers and Geotechnics*. No. 3, Vol. 35, 2008, pp. 372–382. Elsevier. DOI: [10.1016/j.compgeo.2007.08.007](https://doi.org/10.1016/j.compgeo.2007.08.007).

Master's thesis.

- [MasBuc17] *Development and Implementation of a Parallel Framework for Non-Matching Grid Mapping*. Philipp Bucher. 2017.
- [MasIsr06] *Implementation of an Algorithm for Data Transfer on the Fluid-Structure Interface between Non-Matching Meshes in Kratos and an Algorithm for the Generation*. Ute Israel. 2006.

List of Tables

2.1	Transformations of stresses. Source[BookBel+14]	40
3.1	Isoparametric coordinates of the six nodes in the patch of Figure 3.1b	54
3.2	Derivatives of the shape functions from (3.14)	58
3.3	Isotropic-kinematic hardening law constants	69
4.1	Parameters considered for ALM parameters calibration	102
4.2	Results of numerical experiment for ALM parameters	104
4.3	Analogies between friction and plasticity	114
4.4	Parameters considered for simple patch test	136
4.5	Parameters considered for <i>Taylor</i> [ArtTP] patch test	137
4.6	Parameters considered for simple patch test	138
4.7	Parameters considered for 2D sphere-plane <i>Hertz</i> benchmark	139
4.8	Parameters considered for 2D cylinder-cylinder <i>Hertz</i> benchmark	141
4.9	Parameters considered for 3D <i>Hertz</i> plane-sphere problem	143
4.10	Parameters considered for 3D <i>Hertz</i> sphere-sphere problem	144
4.11	Parameters considered for teeth model	145
4.12	Parameters considered for energy conservation test	146
4.13	Parameters considered for double arc benchmark	148
4.14	Parameters considered for arc pressing blocks	151
4.15	Parameters considered for hyperelastic tubes	152
4.16	Parameters considered for contacting cylinders	153
4.17	Parameters considered for press fit	156
4.18	Parameters considered for circular ironing	157
4.19	Parameters considered for shallow ironing	159
4.20	2D lines pairs geometrical configuration	162
4.21	2D lines perturbation amplitudes	162
4.22	3D triangles pairs geometrical configuration	169
4.23	3D triangles perturbation amplitudes	169
4.24	3D quadrilaterals pairs geometrical configuration	169
4.25	3D quadrilaterals perturbation amplitudes	169
5.1	Parameters considered for the cube minimal example	203
5.2	Parameters considered for the tensile test	204
5.3	Parameters considered for gear example	205
6.1	Parameters considered for cantilever example	232
6.2	Parameters considered for the <i>Hessian</i> remesh punch test	235
6.3	Parameters considered for the <i>Hessian</i> remesh <i>Hertz</i> example	243
6.4	Parameters considered for the <i>Hessian</i> remesh contacting cylinders example	246
A.1	Parameters considered for the mesh tying example	292
D.1	Comparison of different methods presented	322

D.2	Convergence on f_p with PM for (D.13)	324
D.3	Convergence on \mathcal{L}_λ with LMM for (D.13)	324
D.4	First active set strategy convergence on \mathcal{L}_λ with LMM for (D.16)	326
D.5	Second active set strategy convergence on \mathcal{L}_λ with LMM for (D.16)	326
D.6	Third active set strategy convergence on \mathcal{L}_λ with LMM for (D.16)	327
D.7	Convergence with PM for (D.21)	328
D.8	First active set strategy convergence on \mathcal{L}_λ with LMM for (D.16) with moved wall	328
D.9	Second active set strategy convergence on \mathcal{L}_λ with LMM for (D.16) with moved wall	330
D.10	Third active set strategy convergence on \mathcal{L}_λ with LMM for (D.16) with moved wall	330
D.11	Convergence on $\mathcal{L}_{\tilde{\lambda}}$ with ALM for (D.22) with $\varepsilon = 0.5$	332
D.12	Convergence on $\mathcal{L}_{\tilde{\lambda}}$ with ALM for (D.22) with $\varepsilon = 1.0$	332
D.13	Convergence on $\mathcal{L}_{\tilde{\lambda}}$ with ALM for (D.22) with $\varepsilon = 5.0$	333
D.14	Convergence on $\mathcal{L}_{\tilde{\lambda}}$ with ALM for (D.22) with $\varepsilon = 10.0$	334
D.15	Convergence on f_p with penalty method for D.26	335

List of Figures

- 1.1 Examples of numerical simulation of forming. Source[OnlShe] 23
- 1.2 Parameters influencing sheet metal formability. Inspired in[BookBan10] 24
- 1.3 Structure of an expert system for the analysis of sheet metal formability. Inspired in [BookBan10] 25
- 1.4 *Springback* 28
- 1.5 Wrinkling phenomena 28

- 2.1 Minimal example of **FEA** for a *Laplacian* problem. Resolution using *Mathematica*[OnlWol] 35
- 2.2 Deformation of a continuum body 37
- 2.3 Polar decomposition of **F** 38
- 2.4 Graphic representation of the arc-length method 41
- 2.5 Graphic representation of *Rayleigh* damping matrix 45

- 3.1 Patch of elements for **BST** and **EBST** 52
- 3.2 Patch performed in the element considering the neighbour elements 58
- 3.3 Nodes considered in the computation of the transverse shear strains 60
- 3.4 Patch test geometry 63
- 3.5 Solution for the displacement 63
- 3.6 Solution for the stress 64
- 3.7 Solution for the displacement 64
- 3.8 Solution for the stress 65
- 3.9 Cantilever with point load 65
- 3.10 Cantilever geometries 65
- 3.11 Solution 66
- 3.12 Cantilever subjected to end bending moment 67
- 3.13 Solution for a cantilever subjected to an end bending moment 67
- 3.14 Frequencies behaviour test 67
- 3.15 *Cook's* membrane geometry 68
- 3.16 *Cook's* membrane convergence solution 68
- 3.17 *Cook's* membrane deformed shape solution 69
- 3.18 Geometry of the scoordelis cylindrical roof 70
- 3.19 Solution for the scoordelis cylindrical roof 70
- 3.20 Geometry of a semi-spherical shell with a hole, both original and deformed geometry 70
- 3.21 Geometry of a semi-spherical shell with a hole solution 71
- 3.22 The open-end cylindrical shell subjected to radial pulling forces. 72
- 3.23 Solution of the open-ended cylindrical shell 72
- 3.24 The semi-cylindrical shell subjected to an end pinching force 72
- 3.25 Solution of the semi-cylindrical shell 73
- 3.26 Slit annular plate 73
- 3.27 Slit annular plate solution 74
- 3.28 Geometry of the hinged cylindrical panel 74
- 3.29 Solution of the cylindrical panel 75
- 3.30 Geometry of the cone shell 75
- 3.31 Cone-shell test 76

3.32 Square thin film under in-plane shear	76
3.33 Solution of the wrinkling simulation	77
3.34 Solution of the FSI -Vein test	78
4.1 Representation of moving blocks during ancient Egypt. Source[BookPop10]	86
4.2 Original <i>Hertz</i> paper in 1882. Source[ArtHer82]	86
4.3 Basic definition of the contact problem	88
4.4 Contact states. Based on[PhDYas11]	89
4.5 NTN . Inspired[PhDYas11]	90
4.6 NTS . Inspired[PhDYas11]	90
4.7 CDM . Inspired[PhDYas11]	91
4.8 STS . Inspired[PhDYas11]	91
4.9 3D segmentation between two triangles	92
4.10 Friction cone for variants of the <i>Coulomb</i> law. Inspired [ArtRao16]	96
4.11 Definition of the contact problem	97
4.12 KKT conditions of non-penetration.	98
4.13 Condition number study for the ALM	102
4.14 Condition number study graphic representation	103
4.15 Dual shape functions for the 2D linear line	105
4.16 Dual shape functions for the 3D linear triangle	106
4.17 Dual shape functions for the 3D bilinear quadrilateral	107
4.18 Example of convergence check in the frictionless contact	112
4.19 Nodal NCP function, or <i>Lagrangian</i> (\mathcal{L}) contribution for the LM in the ALM	112
4.20 <i>Coulomb's</i> schematic depiction of frictional contact conditions in tangential direction	113
4.21 Graphical representation of <i>Coulombs</i> frictional conditions for 3D contact problem. Inspired [PhDYas11]114	114
4.22 Augmented Lagrangian function for the frictional contact problem. Corresponding to (4.61b)	118
4.23 Example of convergence check in the frictional contact	123
4.24 NCP for the frictional contact problem.	123
4.25 Conventional penetration estimations. Symmetric: a segment intersection; b volume intersection. Asymmetric: c,d segment in volume; e,f node in volume; g,h node under surface. Source[PhDYas11] 124	124
4.26 Types of bounding volumes. a) The radius search (BS). b) AABB . c) OBB . d) k-DOP (convex hull). Inspired on [ArtYL08]	125
4.27 BS approach	126
4.28 OBB . Source[ArtRom11]	126
4.29 OBB from surfaces build comparison	127
4.30 Separating axis theorem	127
4.31 SAT concept. Source[OnlCho]	127
4.32 Space subdivision approaches	128
4.33 Tree structures	129
4.34 Consistent gap example. Circular surface vs plane	130
4.35 Consistent gap example. Double curvature surfaces	131
4.36 Self-contact normals discrimination	131
4.37 Self-contact detection simplest case, two parallel planes	133
4.38 Self-contact detection for simple tubular case	134
4.39 Self-contact detection for tubular case	134
4.40 S-shape profile mesh (tetrahedra)	135
4.41 S-shape profile resulting self-contact simulation	135
4.42 Geometries of the simplest patch test	135
4.43 Solution for the frictionless simplest patch test	136
4.44 Geometry and solution for the frictional simplest patch test	136
4.45 Solution for the <i>Taylor</i> patch test in 2D	137
4.46 Solution for the <i>Taylor</i> patch test in 3D	137
4.47 Friction problem from <i>Dong</i> [ArtDon99]	138
4.48 Solution for pure friction problem	138
4.49 Setup for the 2D sphere-plane <i>Hertz</i> benchmark	139

4.50	Solution compared for different mesh sizes for the 2D <i>Hertz</i> plane-sphere contact	140
4.51	Error compared for different mesh sizes for the 2D <i>Hertz</i> plane-sphere contact	140
4.52	Setup for the two cylinders <i>Hertz</i> benchmark	141
4.53	Solution for the two cylinders frictionless <i>Hertz</i> benchmark	141
4.54	Solution for the two cylinders frictional <i>Hertz</i> benchmark	142
4.55	Mesh considered for 3D <i>Hertz</i> plane-sphere	142
4.56	Solution for the 3D <i>Hertz</i> plane-sphere contact	143
4.57	Solution compared for different mesh sizes for the 3D <i>Hertz</i> plane-sphere contact	143
4.58	Error compared for different mesh sizes for the 3D <i>Hertz</i> plane-sphere contact	143
4.59	Reference solution for two spheres <i>Hertz</i> contact. Source[ArtZhu12]	144
4.60	Solution for two hemispheres <i>Hertz</i> contact	144
4.61	Teeth layers model	145
4.62	Solution for teeth layers model	145
4.63	Energy conservation test. Cylinder inside ring. The example can be found in here	146
4.64	Evolution of displacement and energy compared with analytical solution	147
4.65	Solution of the problem at certain times. The time is specified in each subfigure	147
4.66	Double arc benchmark	148
4.67	Double arc meshes	148
4.68	Displacement solution for frictionless in double arc benchmark	149
4.69	Compared solution for frictionless in double arc benchmark	149
4.70	Displacement solution for frictional in double arc benchmark	150
4.71	Compared solution for frictional in double arc benchmark	150
4.72	Arc pressing block	151
4.73	Solution for different stiffness between the arc and the block	151
4.74	Mesh for hyperelastic tubes	152
4.75	Displacement solution for hyperelastic tubes	152
4.76	Slice solution for hyperelastic tubes	153
4.77	Contacting cylinders mesh	153
4.78	Solution for contacting cylinders with horizontal movement	154
4.79	Frictionless reaction solution in the lower cylinder for the horizontal movement	154
4.80	Solution for contacting cylinders with vertical movement	154
4.81	Reaction solution in the lower cylinder for the vertical movement	155
4.82	Press fit problem	155
4.83	Solution for 2D press fit compared with reference	156
4.84	Press fit 2D solution	156
4.85	Press fit 3D solution	157
4.86	Circular ironing test	158
4.87	Circular ironing test solution	158
4.88	Comparison between the frictionless solutions of the shallow and the circular ironing	159
4.89	Shallow ironing test	159
4.90	Shallow ironing test solution, comparing frictional and frictionless cases	160
4.91	Reaction solution for shallow ironing test	160
4.92	Line validation geometries	162
4.93	Convergence plot for the <i>Jacobian</i> derivatives for the 2D linear line	163
4.94	Shape functions for the 2D linear line	163
4.95	Integration segment for a linear line	164
4.96	Convergence plot for the shape function derivatives for the 2D linear line	165
4.97	Convergence plot for the dual shape function derivatives for the 2D linear line	166
4.98	Normal average for a 2D line. Proposed by <i>Taylor and Papadopoulos</i> [ArtPT92]	167
4.99	Convergence plot for the normal derivatives for the 2D linear line	168
4.100	Intersection and clipping procedure during <i>mortar</i> segmentation	170
4.101	Detail on intersection on <i>mortar</i> segmentation	170
4.102	Convergence plot for the <i>Jacobian</i> derivatives for the 3D linear triangle	172
4.103	Convergence plot for the <i>Jacobian</i> derivatives for the 3D bilinear quadrilateral	172
4.104	Shape functions for the 3D linear triangle	173

4.105	Shape functions for the 3D bilinear quadrilateral	174
4.106	Convergence plot for the shape function derivatives for the 3D linear triangle	175
4.107	Convergence plot for the shape function derivatives for the 3D bilinear quadrilateral	176
4.108	Convergence plot for the dual shape function derivatives for the 3D linear triangle	176
4.109	Convergence plot for the dual shape function derivatives for the 3D bilinear quadrilateral	177
4.110	Normal average for a 3D geometry	177
4.111	Convergence plot for the normal vector derivatives for the 3D linear triangle	178
4.112	Convergence plot for the normal vector derivatives for the 3D bilinear quadrilateral	179
5.1	<i>Liberty ships</i> suddenly broke in half while moored at the dock	190
5.2	Types of hardening. Left: Isotropic hardening. Right: Kinematic hardening	192
5.3	Multiplicative decomposition of the deformation gradient \mathbf{F}	193
5.4	<i>Elastic-plastic</i> CL class structure	199
5.5	Consistent constitutive tensor perturbation method convergence. The code which tests the convergence rate can be found here	202
5.6	Simple cube geometry. Only one 8-node <i>hexahedron</i>	202
5.7	<i>Von Mises</i> stresses compared with the reference solution. The test can be found here	203
5.8	Mesh comparisons. It can be found here	203
5.9	Resulting plastic dissipation	204
5.10	Resulting displacement for the mesh of hexahedra	204
5.11	Gears example mesh	205
5.12	Detail solution for the displacement, VM stress and plastic dissipation	205
5.13	Overall solution for the displacement and the VM stress	206
6.1	Adaptive mesh techniques applied on a contact patch test	211
6.2	Concept of h-refinement and p-refinement	213
6.3	Metric analogies. Images from [ArtFA05]	218
6.4	Effects of the metric on a tetrahedra	219
6.5	Initial mesh	222
6.6	Solution for the example	222
6.7	Scalar and its gradient	223
6.8	Stanford's Bunny example	224
6.9	Resulting mesh after remeshing	224
6.10	Construction of a recovered gradient in 1D. Inspired [BookAO11]	226
6.11	Optimal sampling points in 1D linear elements. Source [BookZZT13]	227
6.12	SPR example for linear triangles. Source [BookZZT13]	227
6.13	SPR example	229
6.14	Transfer operators. Source [ArtGJ06]	230
6.15	The beam problem of reference for the internal interpolation	230
6.16	The beam problem of reference for the internal interpolation	231
6.17	Cantilever mesh and geometry	232
6.18	Resulting VM stress before and after recover	232
6.19	<i>Hertz</i> geometry considered. Example here	233
6.20	Solution obtained	233
6.21	Simplest patch test	235
6.22	Punch test geometry considered. Solution $t = 0.5s$. Example here	235
6.23	Punch test solution for VM stress and contact stress <i>Hessian</i> at $t = 1.0s$	236
6.24	Punch test solution for <i>strain energy</i> , VM stress and contact stress <i>Hessian</i> at $t = 1.0s$	236
6.26	Punch test solution for <i>strain energy</i> , VM stress and contact stress <i>Hessian</i> at $t = 2.0s$	236
6.25	Punch test solution for VM stress and contact stress <i>Hessian</i> at $t = 2.0s$	237
6.27	SPR contact interface example for linear triangles. Inspired [ArtWS98 ; BookZZT13]	237
6.28	Error obtained	239
6.29	Mesh evolution	239
6.30	Error solution	239
6.31	Displacement solution	240

6.32	Workflow for <i>Level set/Hessian</i> metric remesh	240
6.33	Workflow for SPR metric remesh	241
6.34	Distance function	242
6.35	Mesh before and after remesh for coarse mesh. Example	242
6.36	Coarse mesh distance function	243
6.37	Final configuration of the beam example	243
6.38	<i>Hertz</i> solution step 1	243
6.39	<i>Hertz</i> solution step 4	244
6.40	<i>Hertz</i> solution step 7	244
6.41	<i>Hertz</i> solution step 10	244
6.42	<i>Hertz</i> solution step 1	245
6.43	<i>Hertz</i> solution step 4	245
6.44	<i>Hertz</i> solution step 7	245
6.45	<i>Hertz</i> solution step 10	245
6.46	Initial mesh in contacting cylinders	246
6.47	Contacting cylinders solution at $t = 0.35s$	246
6.48	Contacting cylinders perspective solution at $t = 0.35s$	247
6.49	Contacting cylinders solution at $t = 0.70s$	247
6.50	Contacting cylinders perspective solution at $t = 0.70s$	247
6.51	Contacting cylinders solution at $t = 1.05s$	248
6.52	Contacting cylinders perspective solution at $t = 1.05s$	248
6.53	Contacting cylinders solution at $t = 1.4s$	248
6.54	Contacting cylinders perspective solution at $t = 1.4s$	248
6.55	Mesh before and after remesh cavity case	249
6.56	Embedded cylinder geometry	249
6.57	Mesh before and after remesh embedded case	250
6.58	Base mesh before and after remesh boxes with isosurface	250
6.59	Mesh before and after remesh boxes with isosurface	250
6.60	STL geometry for the <i>Lamborghini</i>	251
6.61	Original mesh for the <i>Lamborghini</i>	251
6.62	Resulting mesh for the <i>Lamborghini</i>	251
6.63	Geometry and initial mesh in 2D channel. The example can be found here	252
6.64	Solution in 2D channel	252
6.65	Initial mesh in 3D channel	253
6.66	Solution in 3D channel	253
7.1	Cylinder punch case	261
7.2	Mesh considered in the cylinder punch case	262
7.3	Solution for cylinder punch test	262
7.4	Spherical punch case	263
7.5	3D solution for the case $\mu_1 = 0.04$ and $\mu_2 = 0.04$	263
7.6	Solution for spherical punch test	264
A.1	Exact segmentation method for integration. Figure inspired on <i>Popp</i> [PhDPop12]	286
A.2	Collocation method for integration. Figure inspired on <i>Popp</i> [PhDPop12]	286
A.3	<i>Taylor</i> patch test[ArtTP]	287
A.4	Displacement on <i>Taylor</i> test	287
A.5	Compared solution for the augmented contact pressure	288
A.6	Convergence of the solution for different number of GP	288
A.7	Mesh tying example	292
A.8	Solution for mesh tying example	292
B.1	Kratos Multiphysics logo. Image from Kratos web	295
B.2	Source[ArtDRO10], updated with the recent classes added	296
B.3	Model data structure	297
B.4	Concept of submodelpart for BC	297

B.5	Mmg logo. Image from Mmg web	299
B.6	Mmg idealised geometry. Source[ArtDDF14]	299
B.7	Mmg-Kratos integration	300
B.8	Concept of colours	300
C.1	AD concept	306
C.2	NR representation for a SDOF	307
C.3	AD workflow in a UML diagram	310
D.1	Analogy between <i>penalty</i> method and springs. Image from [BookYas13]	314
D.2	Representation of \mathcal{F} . Image from [ArtBMP]	315
D.3	Lagrangian function for the contact problem	317
D.4	Augmented Lagrangian function for the contact problem	320
D.5	Patch tests solutions obtained with <i>Master-Slave</i> elimination method	321
D.6	Simplified contact problem within spring and a wall	323
D.7	Solution for a SDOF solved with PM , comparing different values of ε	323
D.8	Solution for a SDOF solved with LMM	324
D.9	Solution for a NL SDOF solved with LMM . Active set 1	325
D.10	Solution for a NL SDOF solved with LMM . Active set 2	326
D.11	Solution for a NL SDOF solved with LMM . Active set 3	327
D.12	Solution for a NL SDOF problem solved with PM , comparing different values of ε . Moved wall	328
D.13	Solution for a NL SDOF solved with LMM . Active set 1. Moved wall	329
D.14	Solution for a NL SDOF solved with LMM . Active set 2. Moved wall	329
D.15	Solution for a NL SDOF solved with LMM . Active set 3. Moved wall	330
D.16	Solution for a NL SDOF solved with ALM . $\varepsilon = 0.5$. Moved wall	332
D.17	Solution for a NL SDOF solved with ALM . $\varepsilon = 1.0$. Moved wall	333
D.18	Solution for a NL SDOF solved with ALM . $\varepsilon = 5.0$. Moved wall	333
D.19	Solution for a NL SDOF solved with ALM . $\varepsilon = 10.0$. Moved wall	334
D.20	Graphic representation of (D.26)	335
E.1	Mapping concepts	340
E.2	Non-matching mesh of triangles	343
E.3	Scalar solution for non-matching mesh of triangles	344
E.4	Vector solution for non-matching mesh of triangles	344
E.8	Detail of the non-matching mesh discontinuous of triangles	344
E.5	Non-matching mesh of triangles and quadrilaterals	345
E.6	Scalar solution for non-matching mesh of triangles and quadrilaterals	345
E.9	Scalar solution for non-matching discontinuous mesh of triangles	345
E.7	Vector solution for non-matching mesh of triangles and quadrilaterals	346
E.10	Vector solution for non-matching discontinuous mesh of triangles	346

List of Algorithms

1	<i>Line-search</i>	41
2	Algorithm for the frictionless contact problem	111
3	Algorithm for the frictional contact problem	122
4	Self-contact detection algorithm	132
5	Adaptive refinement process	212
6	Compute of nodal average value	232
7	Adaptation of normal penalty coefficient[ArtBMP]	320
8	Consistent gap computation	339
9	Explicit contribution of the pairs during dual λ mortar mapping	341
10	Explicit contribution of the pairs during dual λ mortar discontinuous mapping	342

Glossaries

Acronyms

- AABB** Axis-Aligned Bounding Box.
- AALM** Adapted Augmented Lagrangian Method.
- AD** Automatic Differentiation.
- ADLM** Augmented Dual Lagrange Multiplier.
- AGAUR** Agència de Gestió d'Ajuts Universitaris i de Recerca.
- ALM** Augmented Lagrangian Method.
- AMG** Algebraic MultiGrid.
- ANS** Assumed Normal Strain.
- API** Application Programming Interface.
- APM** Adapted Penalty Method.
- BC** Boundary Conditions.
- BDF** Backward Differentiation Formula.
- BS** Bounding Spheres.
- BSD** Berkeley Software Distribution.
- BST** Basic Shell Triangle.
- BVH** Bounding Volume Hierarchies.
- BVP** Boundary Value Problem.
- CAD** Computer-Aided Design.
- CAE** Computer Aided Engineering.
- CAM** Computer Aided Manufacturing.
- CAS** Computer Algebra Systems.
- CBR** Continuum-Based Resultant.
- CCM** Computational Contact Mechanics.

CD Collision Detection.

CDM Contact Domain Method.

CDT Constrained Delaunay Triangulation.

CFD Computational Fluid Dynamics.

CG Conjugate Gradient.

CIMNE International Center for Numerical Methods in Engineering.

CL Constitutive Law.

COF Coefficient Of Friction.

CPPM Closest Point Projection Method.

CPT Closest Point Transfer.

CSD Computational Structural Dynamics.

DDM Domain Decomposition Method.

DEM Discrete Element Method.

DLMM Double Lagrange Multiplier Method.

DMT Derjaguin, Muller and Toporov.

DOF Degree Of Freedom.

DOP Discretised Orientation Polytopes.

DS Degenerated Shell.

EAS Enhanced Assumed Strain.

EBST Enhanced Basic Shell Triangle.

ELM Element Transfer Method.

ETH Eidgenössische Technische Hochschule.

FAD Forward Automatic Differentiation.

FD Finite Differences.

FE Finite Element.

FEA Finite Element Analysis.

FEM Finite Element Method.

FETI Finite Element Tearing and Interconnect.

FOSS Free and Open Source Software.

FS Finite Strain.

FSI Fluid Structure Interaction.

FV Finite Volumes.

FVM Finite Volume Method.

FVTM Finite Volume Transfer Method.

GFEM Generalized Finite Element Method.

GMRES Generalized Minimal RESidual.

GNM Generalized Newton-Raphson method.

GP Gauss Point.

GUI Graphic User Interface.

HOO Hybrid Object-Oriented.

HPC High Performance Computing.

HSM Hertz-Signorini-Moreau.

IBVP Initial Boundary Value Problem.

ICT Information and Communications Technology.

IGA Iso-Geometric Analysis.

IO Input-Output.

JFNK Jacobian-Free Newton Krylov.

JKR Johnson, Kendall and Roberts.

k-DOP Discrete Orientation Polytopes.

KKT Karush-Kuhn-Tucker.

LBB Ladykaja-Babuska-Brezzi.

LGPL GNU Lesser General Public License.

LHS Left-Hand Side.

LM Lagrange Multiplier.

LMM Lagrange Multiplier Method.

LST Least-Square projection Transfer.

MDD Maximal Detection Distance.

MDOF Multiple Degrees Of Freedom.

MEF Método de los Elementos Finitos.

MFC MultiFreedom Constraint.

MLDA Multi-Layer Design Approach.

MLS Moving Least Square.

MPC MultiPoint Constraint.

MPI Message Passing Interface.

MPM Material Point Method.

NASTRAN NASA STRucture ANalysis.

NC Numerical Controlled.

NCP Non-Linear Complementary Function.

NL Non-Linear.

NR Newton-Raphson.

NTN Node-To-Node.

NTS Node-To-Segment.

NURB Non-Uniform Rational B-spline.

OBB Oriented Bounding Box.

ODE Ordinary Differential Equation.

OO Object Oriented.

OOP Object Oriented Programming.

OSS Open Source Software.

OSU Ohio State University.

PDASS Primal-Dual Active Set Strategies.

PDE Partial Differential Equation.

PK1 First *Piola-Kirchhoff*.

PK2 Second *Piola-Kirchhoff*.

PM Penalty Method.

PVW Principle of Virtual Work.

REP Recovery by Equilibrium of Patches.

RHS Right-Hand Side.

SAC Symbolic and Algebraic Computational Systems.

SAT Separating Axis Theorem.

SDOF Single Degree Of Freedom.

SFT Shape Function projection Transfer.

SPR Super convergent Patch Recovery.

SS Small Strain.

STAS Segment-To-Analytical-Surface.

STL Standard Triangle Language.

STS Segment-To-Segment.

SVD Singular Value Decomposition.

TL Total Lagrangian.

UK United Kingdom.

UL Updated Lagrangian.

UML Unified Modeling Language.

U.S. United States.

VDI Verein Deutscher Ingenieure.

VM *Von Mises*.

WWI First World War.

WWII Second World War.

Mathematical symbols

C^0 The class C^0 consists of all continuous functions.

C^1 The class C^1 consists of all differentiable functions whose derivative is continuous; such functions are called continuously differentiable.

C^k When the function f is said to be of k order differentiability.

Δ Symbol for directional derivative, corresponding with the partial derivative respect the **DOF**.

H It is a square matrix of second-order partial derivatives of a scalar-valued function or scalar field.

J For the *Jacobian*. When the matrix is a square matrix, both the matrix and its determinant are referred to as the *Jacobian*. Except explicitly stated the latter is the case.

κ The condition number of a function measures how much the output value of the function can change for a small change in the input argument. A problem with a low condition number is said to be **well-conditioned**, while a problem with a high condition number is said to be **ill-conditioned**.

$\langle \rangle$ *Macaulay* brackets, these are a notation used to describe the ramp function.

L_2 The *Euclidean* norm. As $p = 2$ in L_p . $L_2 = \sqrt{\sum_{i=1}^n \|x_i\|^2}$.

L_∞ The infinity norm. As p approaches ∞ the L_p approaches the infinity norm or maximum norm. $L_\infty = \max_i \|x_i\|$.

L_p The p norm. For $p = 1$ we get the *taxicab* norm, for $p = 2$ we get the *Euclidean* norm. $L_p = \left(\sum_{i=1}^n \|x_i\|^p\right)^{\frac{1}{p}}$.

∇ The vector derivative of a scalar field f .

∇^2 Laplace operator or Laplacian is a differential operator given by the divergence of the gradient of a function on Euclidean space.

π Ratio of the circumference of a circle to its diameter.

Second order PDE Second order linear **PDE** are classified as either elliptic, hyperbolic, or parabolic. Any second order linear **PDE** in two variables can be written in the form: $Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu + G = 0$.

elliptic PDE **Second order PDE** is elliptic when: $B^2 - AC < 0$.

hyperbolic PDE **Second order PDE** is hyperbolic when: $B^2 - AC > 0$.

Taylor expansion *Taylor* series is a representation of a function as an infinite sum of terms that are calculated from the values of the function's derivatives at a single point. $f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$

Rotation-free shell and solid-shell elements

α The additional internal degree of freedom considered in the stabilisation.

k_α The stiffness contribution by the α **DOF** in the **SPRISM** element.

A_M The area of the central triangle of the **EBST** element.

C_{13} The mixed component in axial-normal direction of the left *Cauchy* tensor in the **SPRISM** element.

C_{23} The mixed component in shear-normal direction of the left *Cauchy* tensor in the **SPRISM** element.

C_3 The third row of the left *Cauchy* tensor in the **SPRISM** element.

C_{33} The normal direction component of the left *Cauchy* tensor in the **SPRISM** element.

$C_{\eta 3}$ The transverse shear component in η direction of the left *Cauchy* tensor in the **SPRISM** element.

$C_{\xi 3}$ The transverse shear component in ξ direction of the left *Cauchy* tensor in the **SPRISM** element.

I The node I index in the **SPRISM** element.

$N^I(\xi)$ The shape functions in the node I in the **SPRISM** element.

\mathbf{u}^I The displacements in the node I in the **SPRISM** element.

\mathbf{x}^I The current coordinates (or deformed configuration) in the node I in the **SPRISM** element.

\mathbf{X}^I The original coordinates (or undeformed configuration) in the node I in the **SPRISM** element.

$\hat{\kappa}$ The constant curvature field in the **EBST** element.

κ The curvature vector in the **EBST** element.

$\bar{\mathbf{C}}$ The improved left *Cauchy* tensor in the **SPRISM** element.

\bar{C}_{33} The normal direction component of the improved left *Cauchy* tensor in the **SPRISM** element.

$\bar{\mathbf{F}}$ The improved deformation gradient tensor (**F**) in the **SPRISM** element.

$\bar{\mathbf{R}}$ The improved rotation tensor (**R**) in the **SPRISM** element.

$\bar{\mathbf{U}}$ The improved right stretch tensor (**U**) in the **SPRISM** element.

λ_α The eigenvalues of **U** in the **EBST** element.

\mathbf{r}_α The eigenvectors of **U** in the **EBST** element.

\mathbf{f}_1 The deformation gradient components in the first in-plane direction (ξ) in the **SPRISM** element.

\mathbf{f}_2 The deformation gradient components in the second in-plane direction (η) in the **SPRISM** element.

\mathbf{f}_3 The deformation gradient components in the normal direction (ζ) in the **SPRISM** element.

\mathbf{f}_t The deformation gradient components in natural coordinate derivative in the **SPRISM** element.

h_0 The initial thickness of the shell.

L_i^e The linear shape functions of each node in the **EBST** element.

φ The coordinates of the middle surface in the **EBST** element.

\mathbf{t}_3 The normal in the middle surface in the **EBST** element.

ζ The distance from the point to the middle surface in the undeformed configuration in the **EBST** element.

n_G The number of integration points along the direction ζ in the **SPRISM** element.

w_i^e The nodal deflection of each node in the **EBST** element.

H This is an auxiliary relation operator between the α **DOF** and the displacement **DOF** derived from the formulation in the **SPRISM** element.

r_α The residual contribution by the α **DOF** in the **SPRISM** element.

$\bar{\mathbf{B}}_m$ The membrane tangent matrix contribution in the **SPRISM** element.

$\bar{\mathbf{B}}^f$ The modified tangent matrix in the **SPRISM** element.

$\bar{\mathbf{B}}_3$ The normal direction tangent matrix contribution in the **SPRISM** element.

$\bar{\mathbf{B}}_s$ The shear tangent matrix contribution in the **SPRISM** element.

λ The parameter that relates the thickness at the present and initial configuration in the **EBST** element.

y_3 The spatial local coordinate in the transverse direction in the **SPRISM** element.

Constraint enforcement and optimization

$\bar{\lambda}_n$ The augmented contact stress vector, used on the **ALM** for contact problems. Corresponds with $\mathbf{n}\bar{\lambda}_n$.

\mathcal{L} This is the symbol used to represent the *Lagrangian* functional. Different *Lagrangian* can be obtained depending of the formulation considered.

$\mathcal{L}_{\bar{\lambda}}$ This is the *Lagrangian* resulting of the **ALM** method.

\mathcal{L}_λ This is the *Lagrangian* resulting of the **LMM** method.

λ *Lagrange* multiplier symbols. The Greek letter λ is equivalent to the Latin letter l , representing then the name of the mathematician that gives the name to the method.

$\bar{\lambda}$ On a **ALM** formulation, we call augmented *Lagrange* multiplier the value obtained after adding to our λ the contribution of the constraint multiplied by a ε value $\lambda + \varepsilon g(x)$.

λ_n *Lagrange* multiplier used on normal contact. It corresponds with the normal contact pressure.

$\bar{\lambda}_n$ The augmented contact stress, used on the **ALM** for contact problems, computed as $\bar{\lambda}_n = \lambda_n + \varepsilon g(x)$, if the scale factor k is taken into consideration then can be evaluated as $\bar{\lambda}_n = k\lambda_n + \varepsilon g(x)$.

λ_τ *Lagrange* multiplier used on frictional contact. It corresponds with the frictional contact pressure.

λ *Lagrange* multiplier vector field.

g Constraint, subjects to certain restrictions a function during optimisation problems.

f The symbol considered to represent the function to optimise on optimisation problems.

f_p This is the function augmented with the **PM**.

ε The penalty symbol, used to represent a value considered in order to impose a certain restriction a optimisation problem resolution. The value must be relatively big to the system of equation considered in order to take effect over the problem resolution. An infinite value would provide a exact fulfill of the constraint, but at the same time numerically impossible to be solved.

ε_n The penalty ε on the normal direction.

ε_τ The penalty ε on the tangent direction.

Finite element framework

α_f The damping introduced on the elastic forces in the α -**generalised** scheme.

α_m The damping introduced on the inertial forces in the α -**generalised** scheme.

α The parameter considered in the conceptual definition of the explicit and implicit schemes on the equilibrium equation, being explicit when $\alpha = 0$ and implicit when $\alpha = 1$.

B The strain matrix, which gives the *strain-displacement* equation $\varepsilon = \mathbf{B}\mathbf{u}$.

b The left *Cauchy-Green* tensor.

Γ Represents the boundary in a generic solid.

Γ_0 Represents the boundary in a generic solid in the initial configuration.

C The right *Cauchy-Green* tensor.

$\mathcal{O}(h)^s$ Represents an order s approximation in an integration scheme.

c The damping coefficient.

c_{cr} The critical damping coefficient. Critical damping occurs when the damping coefficient is equal to the undamped resonant frequency of the oscillator.

ξ Critical-damping ratio, usually expressed in % of c_{cr} .

δ Stiffness-proportional damping coefficient.

η Mass-proportional damping coefficient.

F The deformation gradient is the derivative of each component of the deformed \mathbf{x} vector with respect to each component of the reference \mathbf{X} vector.

R It is the rotation tensor.

U It is the right stretch tensor.

V It is the left stretch tensor.

$\delta\mathbf{u}$ The infinitesimal incremental of displacements.

\mathcal{C}^0 It consists of all continuous functions.

\mathcal{C}^1 It consists of all differentiable functions whose derivative is continuous; such functions are called continuously differentiable.

$\hat{\cdot}$ Represents a discretisation of the continuous form.

w_{gp} The integration weight in a certain **GP**.

C The damping matrix.

K The stiffness matrix.

\mathbf{K}^{geo} The geometric stiffness matrix.

\mathbf{K}^{mat} The material stiffness matrix.

M The mass matrix.

P Stress divergence or stress force term.

ξ The first local coordinate for the surface geometries, also corresponds with the local derivative on the line.

η The second local coordinate for the surface and volume geometries.

ζ The third local coordinate for the volume geometries.

$\nabla_{\mathbf{x}}\mathbf{u}$ The material displacement gradient tensor.

n_m^c Number of nodes in the given master condition.

n_s^c Number of nodes in the given slave condition.

\mathcal{J} The *Jacobian* operator during the **NR** procedure. It corresponds with the **LHS** of the system of equations.

ϕ An arbitrary motion for a given solid.

\mathbf{u} The increment of position or displacement.

$\dot{\mathbf{u}}$ The velocity of a certain material point.

$\ddot{\mathbf{u}}$ The acceleration of a certain material point.

X The initial position of each material particle.

x The current position of each material particle.

ϕ^* It represents the **pull-back** operation.

ϕ_* It represents the **push-forward** operation.

\mathfrak{R}^3 Represents the three-dimensional Euclidean space.

ρ The density of a given material.

f The resulting vector of forces.

N The shape function is the function which interpolates the solution between the discrete values obtained at the mesh nodes in the **FEM**.

\mathbf{N}_b The shape functions corresponding to the elements considered.

N_i The shape function value on the node i in a certain **GP**.

Ω Represents a generic solid for our problem.

Ω_0 Represent the solid in the initial configuration.

- Ω_e Represent the discrete elements e which conform the domain.
- Ω^i Represent the i solid of the contact problem.
- Ω_0^i Represent the i solid of the contact problem in the initial configuration.
- m The parameter used to define a certain strain measure.
- $\mathbf{E}^{(m)}$ The strain tensor for a given strain measure in function of the parameter m .
- σ *Cauchy* stress.
- $\hat{\sigma}$ Corotational *Cauchy* stress.
- τ **First Piola-Kirchhoff (PK1)** stress.
- \mathbf{P} Nominal stress (transpose of **PK1** stress).
- \mathbf{S} **PK2** stress.
- t Represents the time on our problem.
- Δt Represents the time increment in a given step.

Other terms

- git** Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. See [link](#).
- C++** It is a general-purpose programming language created as an extension of the C programming language, but **Object Oriented Programming (OOP)** designed.
- Kratos** *Kratos Multiphysics* (A.K.A *Kratos*) is a framework for building parallel multidisciplinary simulation software. Modularity, extensibility and **HPC** are the main objectives. Kratos has **BSD** license and is written in **C++** with extensive **Python** interface. See [B.2.Kratos Multiphysics](#).
- Mmg** Mmg is an open source software for simplicial remeshing. See [B.3.Mmg library](#).
- OpenMP** *Open Multi-Processing* is an **API** that supports multi-platform shared memory multiprocessing programming.
- Trilinos** An **OO** software framework for the solution of large-scale, complex multi-physics engineering and scientific problems.

Mortar formulation

- \mathbf{A}_e The coefficient matrix for dual λ shape functions.
- \mathbf{D}_e Auxiliar matrix necessary in order to compute \mathbf{A}_e . The D is used to represent that represents the slave side.
- \mathbf{M}_e Auxiliar matrix necessary in order to compute \mathbf{A}_e . The M is used to represent that represents the master side.
- ξ_a^1 On the mortar integrated line corresponds with the local coordinate of the first segment point on the slave side.
- ξ_b^1 On the mortar integrated line corresponds with the local coordinate of the second segment point on the slave side.
- ξ_a^2 On the mortar integrated line corresponds with the local coordinate of the first segment point on the master side.
- ξ_b^2 On the mortar integrated line corresponds with the local coordinate of the second segment point on the master side.

J_{clip} The J corresponding to the integration triangle during 3D mortar segmentation.

mortar Mortar finite element methods (segment-to-segment) allow for a variationally consistent treatment of contact conditions despite the fact that the underlying contact surface meshes are non-matching or/and non-conforming.

D Mortar operator corresponding with the *slave* side. When using **DLMM** the matrix is diagonal. It is a matrix of $n \times n$ size.

M Mortar operator corresponding with the *master* side. It is a matrix of $m \times n$ size.

\mathbf{n}_{plane} Normal which defines the auxiliary plane considered during the mortar segmentation procedure.

\bar{N} The shape functions in the integration triangles during the 3D surface-mortar segmentation.

\mathbf{x}_{clip} The resulting point coordinates of the clipping algorithm between the segments from the master and slave side.

$\hat{\mathbf{x}}_1^1$ The first node coordinates on the slave side during a clipping procedure.

$\hat{\mathbf{x}}_2^1$ The second node coordinates on the slave side during a clipping procedure.

$\hat{\mathbf{x}}_1^2$ The first node coordinates on the master side during a clipping procedure.

$\hat{\mathbf{x}}_2^2$ The second node coordinates on the master side during a clipping procedure.

Contact mechanics

\mathbf{B}_{co} The discrete mortar unilateral contact operator.

bilateral Contact between two or more deformable solids.

unilateral Contact between a deformable and a rigid solid.

Tribology The science that covers the interfacial behaviour related to frictional response. Tribology covers topics like adhesion, friction, wear, lubrication, thermal contact or electric contact.

\mathcal{W} This space represents the dual space of the trace space of V .

β The velocity-traction ratio in friction theory.

μ The friction coefficient, also denominated **Coefficient Of Friction (COF)**, is a dimensionless scalar value which describes the ratio of the force of friction between two bodies and the force pressing them together.

g The *Tresca* law constant threshold parameter.

\mathcal{F} Represents the friction threshold, corresponding with μN in the case of the *Coulomb* frictional law.

Γ_c The contact boundary conditions in the initial configuration.

γ_c The contact boundary conditions in the current configuration.

Γ_c^1 The contact boundary conditions in the initial configuration in the domain first domain.

Γ_c^2 The contact boundary conditions in the initial configuration in the domain second domain.

Γ_c^i The contact boundary conditions in the initial configuration in the domain i .

Γ_σ The *Neumann* boundary conditions in the initial configuration.

Γ_u The *Dirichlet* boundary conditions in the initial configuration.

g_n The normal gap considered during the contact computations. Can be defined as the closest distance to a certain object in the normal direction.

\mathbf{t}_{co} It represents the traction on the interface.

\mathbf{t}_{co}^n It represents the contact normal stress on the interface.

\mathbf{t}_{co}^τ It represents the contact tangent stress on the interface.

χ Represents the contact interface mapping. This means that the values from the domain 1 are projected into the domain 2.

χ_h Represents the discrete contact interface mapping. Discrete version of χ .

\mathcal{M} The *Lagrange* multiplier solution space.

\mathcal{M} The vector *Lagrange* multiplier solution space.

\mathcal{M}_h The discrete vector *Lagrange* multiplier solution space.

l_n The normal contribution for the **ALM** frictional contact.

ξ_g^2 The local coordinates corresponding to the **GP** on the master side.

ξ_g^1 The local coordinates corresponding to the **GP** on the slave side.

\mathbf{n} Represents the unitary normal vector for a given point.

τ Represents the complementary direction to the normal vector \mathbf{n} .

τ_1 Represents the first unitary tangent vector for a given point.

τ_2 Represents the second unitary tangent vector for a given point.

l_τ The frictional (tangent) contribution for the **ALM** frictional contact.

\mathbf{n}_{area} Represents the area normal vector for a given point. The area normal corresponds with the vector perpendicular to the surface at a given point.

$\bar{\mathbf{n}}$ Represents the averaged normal vector between adjacent entities.

\mathbf{n} The normal of a neighbour entity to a given node.

k The scale factor used in order to define the contact \mathcal{L} contribution. Helps to improve the condition number (κ) of the system of equations.

\mathbf{t}_c^1 The slave side contact traction.

$\mathbf{c}(t)$ Corresponds with the observer frame, which is frame indifferent.

$\mathbf{Q}(t)$ An orthogonal tensor considered in the slip frame indifferent formulation.

$\mathbf{c}(t)$ It is the relative rigid body translation between the original spatial frame and observer $\mathbf{c}(t)$.

$\mathbf{v}_{\tau,rel}$ It represents the tangential relative velocity of the contacting bodies.

$\bar{\mathbf{u}}_\tau$ It represents the nodal slip increment.

$\tilde{\mathbf{v}}_\tau$ It represents the tangential relative velocity.

\mathcal{U}^i The solution spaces defined on the contact problem.

\mathcal{V}^i The weighting spaces defined on the contact problem.

- \mathcal{A} This symbol is used in order to represent the slave active **DOF** in the algebraic representations of the contact problem.
- \mathcal{I} This symbol is used in order to represent the slave inactive **DOF** in the algebraic representations of the contact problem.
- \mathcal{M} This symbol is used in order to represent the master **DOF** in the algebraic representations of the contact problem.
- \mathcal{N} This symbol is used in order to represent the **DOF** which are not involved in the contact problem.
- \mathcal{S} This symbol is used in order to represent the slave **DOF** in the algebraic representations of the contact problem.
- $s/$ This symbol is used in order to represent the slave active slip **DOF** in the algebraic representations of the contact problem.
- st This symbol is used in order to represent the slave active stick **DOF** in the algebraic representations of the contact problem.
- $\hat{\mathbf{u}}^2$ The displacements of the second (master) domain projected on the first (slave) domain on contact simulations.
- $\delta\mathcal{L}_{co}$ The unilateral contact contribution to the energy potential.
- \bar{g}_n It represents the nodal weighted gap.
- $\bar{\mathbf{v}}_{\tau_j}$ It represents the weighted relative velocity.
- $\delta\mathcal{L}_{int,ext}$ The internal and external contributions to the energy potential.
- $\delta\mathcal{L}_{kin}$ The kinematic contribution to the energy potential.
- $\delta\mathcal{L}_{\lambda}$ The unilateral contact constraint contribution to the energy potential.
- $\delta\mathcal{L}_{mnt}$ The mesh tying interface contribution to the energy potential.
- \mathbf{x}^1 The coordinates on the first (slave) domain on contact simulations.
- \mathbf{x}^2 The coordinates on the second (master) domain on contact simulations.
- $\hat{\mathbf{x}}^2$ The coordinates of the second (master) domain projected on the first (slave) domain on contact simulations.
- \mathbf{x}_n The normal components of the coordinates of all nodes.
- $\mathbf{x}_{n\mathcal{M}}$ The normal components of the coordinates of the master nodes.
- $\mathbf{x}_{n\mathcal{S}}$ The normal components of the coordinates of the slave nodes.

Adaptive remeshing

- $R_{\lambda_{rel}}$ This anisotropic relative radius is considered in order to enforce a higher level of anisotropy on the *Hessian* metric tensor.
- ρ The anisotropic ratio of the mesh, which indicates the anisotropic shape factor of the element, with 1 being isotropic mesh and 0 a collapsed geometry.
- anisotropic** Anisotropic meshes are those that employ extremely narrow mesh elements.
- \mathbf{e}_1 The first component of the common basis in a metric (\mathcal{M}).
- \mathbf{e}_2 The second component of the common basis in a metric (\mathcal{M}).
- \mathbf{e}_3 The third component of the common basis in a metric (\mathcal{M}).

- f Scalar function used on the remeshing chapter to introduce concepts.
- h The element size on the mesh.
- h_{max} The maximum element size on the mesh.
- h_{min} The minimum element size on the mesh.
- h_{new} The new element size on the mesh.
- ε_K This interpolation error defined in the *Hessian* metric remeshing.
- ε The maximum level of error tolerated on the mesh elements.
- $\Pi_h \mathbf{u}$ The linear interpolation of \mathbf{u} in the mesh \mathcal{T}_h in considering the **FE** shape functions (**N**).
- K This identifies the element number K on the mesh \mathcal{T}_h .
- \mathcal{M} The metric which defines the size of the elements in a mesh in all the directions.
- \mathcal{M}_1 The first metric considered in a metric intersection.
- \mathcal{M}_2 The second metric considered in a metric intersection.
- \mathcal{M}_{aniso} The metric considered in anisotropic mesh.
- $\tilde{\mathcal{M}}$ This metric defines a region that is minimal in volume.
- c A constant independent of the current mesh \mathcal{T}_h .
- c_d A constant ratio of a mesh constant.
- $e(x)$ The vertices of a given geometry.
- E_K It is the set of edges of K .
- ϵ The error threshold for the *Hessian* metric.
- \mathcal{M}_{iso} The metric considered in isotropic mesh.
- $\mathbf{\Lambda}$ Eigenvalues matrix of \mathcal{M} .
- \mathcal{N} The common basis between two different metrics (\mathcal{M}).
- \mathcal{P} Eigenvectors matrix of \mathcal{N} .
- \mathcal{R} Eigenvectors matrix of \mathcal{M} .
- \mathcal{T}_h It represents a certain mesh.
- f_{norm} The normalisation factor considered during the *Hessian* metric computation to adequate to the magnitude order of the problem.
- P The reference point considered in the construction of a metric.
- \mathbf{a}_i The set of unknown parameters during the **SPR**.
- \mathbf{e}_{var} The error measure for a certain variable var .
- $\hat{\mathbf{e}}_{var}$ The error estimation for a certain variable var .
- $\|\mathbf{e}\|_E^2$ The energy norm considered as an error measure.

$\|e\|_K$ The actual error in the element K .

η The relative energy norm error.

$\eta_{threshold}$ The relative energy norm error limit, or threshold, or sometimes also denominated permissible value.

$\tilde{\sigma}^*$ The nodal recovered stress values.

\mathbf{p} The polynomial terms used to recover the stresses during the **SPR**.

σ^* The overall recovered stress values.

ξ_K The refinement ratio for each element when considering an adaptive remeshing.

$\|e\|_{threshold}$ The threshold error considered on the adaptive remeshing.

$\|e\|_{threshold,K}$ The permissible error in the element K .

$\|\mathbf{u}\|_E$ The total energy of deformation.

\mathbf{u} The exact solution of the **PDE** problem.

\mathbf{e}_u The **FE** error, the difference between the exact solution \mathbf{u} and the **FE** solution \mathbf{u}_h .

$\tilde{\mathbf{e}}_u$ The interpolation error, the difference between the exact solution \mathbf{u} and the interpolated solution $\Pi_h \mathbf{u}$.

\mathbf{u}_h The **FE**-solution.

\bar{x} It represents the coordinates respect the interior vertex node of the patch $x - x_c$.

Plasticity and constitutive relations

associated plastic flow rule An associated flow rule can be interpreted as saying that the plastic strain increment vector is normal to the *yield surface* (Φ). This is denominated the normality rule.

non-associated plastic flow rule A non-associated flow rule is the normality rule is not fulfilled.

Ω_0 The initial configuration.

Ω_p The plastic configuration.

\mathbb{C} The constitutive tensor.

$\mathbb{C}^{\sigma T}$ Material tangent moduli relating *Truesdell* rate of *Cauchy* stress tensor to the rate of deformation tensor.

\mathbf{F}^e The elastic deformation gradient.

\mathbf{F}^p The plastic deformation gradient.

\mathbf{R}^p The plastic rotation tensor.

\mathbf{R}^e The elastic rotation tensor.

$\mathbf{U}^e, \mathbf{V}^e$ The elastic stretch tensors (right and left).

$\mathbf{U}^p, \mathbf{V}^p$ The plastic stretch tensor (right and left).

ε_{kl}^e The elastic components of the strain tensor.

\mathbf{C}^e The elastic right *Cauchy-Green* tensor. Calculated as $\mathbf{C}^e := (\mathbf{F}^e)^T \cdot \mathbf{F}^e$.

\mathbb{C}^{ep} The elasto-plastic tangent operator.

- $\bar{\Psi}$ The free-energy potential.
- α The (strain-like) internal variables of the constitutive model.
- J_2 The **VM** criterion. Denominated that way as corresponds with the second deviatoric stress invariant. It also represents the distortion strain energy.
- η The kinematic plastic hardening internal variable.
- \mathbf{T} The *Mandel* stress tensor. The *Mandel* stress tensor is a convenient stress measure for finite plasticity.
- G_f Fracture energy, the energy that has to be dissipated to open a fracture in a unitary area of the material.
- ν The *Poisson* ratio.
- γ_c The yield stress.
- E The *Young* modulus of the material.
- p The hydrostatic stress.
- \mathbf{M}_p The plastic flow functions.
- \mathbf{M}_h The general hardening flow function.
- \mathcal{K} The plastic hardening function, which depends on the mechanical process.
- A The plastic hardening parameter.
- $h_{\kappa}(\boldsymbol{\sigma}, \mathbf{q})$ The scalar function of plastic hardening.
- $\mathbf{h}_{\kappa}(\boldsymbol{\sigma}, \mathbf{q})$ The tensor function of plastic hardening.
- κ^p The internal variable of plastic hardening.
- $\dot{\gamma}$ The plastic multiplier. Also denominated plastic consistency factor, it is a non-negative scalar which is determined by the *Prager* consistency condition.
- Γ The plastic potential.
- \mathbf{L}^p The plastic velocity gradient.
- \mathbf{q} The (stress-like) internal variables of the constitutive model, usually the hardening variables.
- $\bar{\mathbf{S}}$ The stresses in the intermediate configuration.
- Ψ The stored energy potential for an hyperelastic material.
- w The stored energy potential for an hyperelastic material expressed in terms of the *Green-Lagrange* strain.
- $\boldsymbol{\varepsilon}$ The strain tensor in *Voigt* notation.
- $\boldsymbol{\sigma}$ The stress tensor in *Voigt* notation.
- Π The total energy in the system in the final configuration.
- Π^* The total energy in the system in the initial configuration.
- Φ The yield surface or discontinuity surface. Also denominated potential plastic surface.