University of Glasgow

Glasgow Computational Engineering Centre

First Order Conservation Law Framework for Large Strain Explicit Contact Dynamics

Callum John Runcie

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

James Watt School of Engineering
College of Science and Engineering
University of Glasgow

April 2023

Abstract

This thesis presents a novel vertex-centred finite volume algorithm for explicit large strain solid contact dynamic problems where potential contact loci are known *a priori*. This methodology exploits the use of a system of first order conservation equations written in terms of the linear momentum and a triplet of geometric deformation measures, consisting of the deformation gradient tensor, its co-factor and its determinant, in combination with their associated Rankine-Hugoniot jump conditions. These jump conditions are used to derive several dynamic contact models ensuring the preservation of hyperbolic characteristic structure across solution discontinuities at the contact interface, which is a significant advantage over standard quasi-static contact models where the influence of inertial effects at the contact interface is completely neglected. By taking advantage of this conservative formalism, both kinematic (velocity) and kinetic (traction) contact-impact conditions are explicitly enforced at the fluxes through the use of the appropriate jump conditions. Specifically, the kinetic contact condition was enforced, in the traditional manner, through the linear momentum equation, while the kinematic contact condition was easily enforced through the geometric conservation equations without requiring a computationally demanding iterative scheme. Additionally, a Total Variation Diminishing shock capturing technique can be suitably incorporated in order to improve dramatically the performance of the algorithm at the vicinity of shocks, importantly no ad-hoc regularisation procedure is required to accurately capture shock phenomena. Moreover, to guarantee stability from the spatial discretisation standpoint, global entropy production is demonstrated through the satisfaction of semi-discrete version of the classical Coleman-Noll procedure expressed in terms of the time rate of the Hamiltonian energy of the system. Finally, a series of numerical examples is presented in order to assess the performance and applicability of the proposed algorithm suitably implemented across MATLAB and a purpose built OpenFOAM solver.

Contents

List of Tables

List of Figures

List of Algorithms

Listings

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors, Dr. Chun Hean Lee and Dr. Peter Grassl for their continuous support and guidance over the course of my research. In particular, Dr. Lee who has encouraged me towards academic success by imparting his knowledge and insight throughout our countless discussions.

I am deeply grateful to the University of Glasgow, College of Science and Engineering for granting me a College Scholarship without which, this research would not have been possible. I would also like to thank all members of the Glasgow Computational Engineering Centre for their supportive working environment and constructive feedback.

For their unwavering love and support, I would like to offer my profound gratitude to my girlfriend, Diana, my parents and family whose love, understanding and support have helped me throughout these past four years.

Finally, I would like to dedicate this work to my Grandmother, Avril, who sadly passed away during the final year of this work. Her constant love and support throughout my academic journey is sorely missed. This is for her.

Declaration

I certify that the thesis presented here for examination for PhD degree of the University of Glasgow is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it) and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

I acknowledge that if any issues are raised regarding good research practice based on review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Signature: .....C.Runcie.....
Date: .....April 2023.....

Research Output

Journal Publications

- **C. J. Runcie**, C. H. Lee, J. Haider, A. J. Gil, J. Bonet (2022): An acoustic Riemann solver for large strain computational contact dynamics. IJNME. 2022; 1-49. doi: 10.1002/nme.7085
- **C. J. Runcie**, C. H. Lee, A. J. Gil, J. Bonet (2023): A first order hyperbolic framework for non-linear large strain computational contact dynamics. IJNME. *To be Submitted*.

Conference Publications

- **C. J. Runcie**, C. H. Lee, A. J. Gil, J. Bonet (2023): "A Non-Linear Riemann Solver for Large Strain Contact Dynamics". UK Association for Computational Mechanics (UKACM) Conference 2023, University of Warwick, United Kingdom, 19-21 April 2023.
- **C. J. Runcie**, C. H. Lee, A. J. Gil, J. Bonet (2022): "An upwind vertex centred finite volume algorithm for large strain contact dynamics in OpenFOAM". In proceedings of the 10^{th} OpenFOAM Conference, Virtual Event, 8-10 November 2022.
- **C. J. Runcie**, C. H. Lee, A. J. Gil, J. Bonet (2021): "An explicit vertex centred finite volume algorithm for large strain fast contact dynamics". UK Association for Computational Mechanics (UKACM) Conference 2021, Loughborough, United Kingdom, 14-16 April 2021.
- **C. J. Runcie**, C. H. Lee, P. Grassl, A. J. Gil, J. Bonet (2020): " An Upwind Vertex Centred Finite Volume Algorithm for Computational Contact Dynamics". In proceedings of the UK Association for Computational Mechanics (UKACM) Conference 2020. DOI: 10.17028/rd.lboro.12102546.v1.

Abbreviations

ALE Arbitrary Lagrangian Eulerian

AMI Arbitrary Mesh Interface

CAD Computer Aided Design

CAE Computer Aided Engineering

CCFVM Cell Centred Finite Volume Method

CFD Computational Fluid Dynamics

CFL Courant-Friedrichs-Lewy

CSD Computational Solid Dynamics

FEM Finite Element Method

FVM Finite Volume Method

MEE Modified Entropic Elasticity

MG Mie-Grüneisen

NTN Node-To-Node

NTS Note-To-Segment or Note-To-Surface

RK2 Two Stage Runge-Kutta

STS Segment-To-Segment or Surface-To-Surface

RS Riemann Solver

TVD Time Variation Diminishing

VCFVM Vertex Centred Finite Volume Method

Nomenclature

Operators

$[\cdot] \cdot [\cdot]$	Dot/Scalar product	
$[\![\cdot]\!]$	Jump in variable $[\cdot]$	
max	Maximum of field	
min	Minimum of field	
$\mathrm{tr}[\cdot]$	Trace of tensor	
$\nabla_0 \cdot [\cdot]$, DIV$[\cdot]$	Material divergence	
$\nabla_0 \times [\cdot]$, CURL$[\cdot]$	Material curl	
$[\cdot] \boldsymbol{\times} [\cdot]$	Tensor cross product	
$[\cdot] \otimes [\cdot]$	Dyadic/Outer product	
$[\cdot] \times [\cdot]$	Vector cross product	
$\boldsymbol{\nabla}_0[\cdot]$, GRAD$[\cdot]$	Material gradient	
$\frac{\partial^2}{\partial [\cdot]^2}$	Second partial derivative with respect to $[\cdot]$	
$\frac{\partial}{\partial [\cdot]}$	First partial derivative with respect to $[\cdot]$	
$\frac{d}{d[\cdot]}$	First ordinary derivative with respect to $[\cdot]$	
det	Determinant	

Scalars

$\Delta\gamma$	Plastic multiplier	$[-]$
Γ	Surface	$[-]$
γ	Gap rate	$[\mathrm{m\,s^{-1}}]$

κ	Bulk modulus	[Pa]
λ	First Lamé coefficient	[Pa]
λ_e	Elastic stretch	[−]
$\mathcal{E}$	Internal energy	[J]
$\mathcal{H}$	Hamiltonian energy	[J]
U	Volumetric strain energy	[J]
μ	Shear modulus	[Pa]
$\Sigma_{[\cdot]}$	Work conjugate	[−]
E	Elastic modulus	[Pa]
E_T	Total energy	[J]
f_s	Slip criterion	[−]
H	Hardening parameter	[Pa]
k	Friction coefficient	[−]
p	pressure	[Pa]
s	Slope of shock wave to particle wave speed ratio	[−]
S	Stabilisation	[J]
W	Strain energy	[J]
$\dot{\Pi}_{\text{ext}}$	External work	[J]
$\dot{D}$	Dissipative energy	[J]
ν	Poisson's ratio	[−]
Ω	Enclosed volume	[m^3]
$\partial\Omega$	Boundary surface	[−]
Φ	Limit function	[−]
ρ_0	Reference density	[$\text{kg}\,\text{m}^{-3}$]
ξ, ζ	Material parameters	[Pa]
c	Acoustic wave speed	[$\text{m}\,\text{s}^{-1}$]

c_p	Acoustic pressure wave speed	$[\mathrm{m\,s^{-1}}]$
c_s	Acoustic shear wave speed	$[\mathrm{m\,s^{-1}}]$
dA	Differential reference area	$[\mathrm{m^2}]$
da	Differential spatial area	$[\mathrm{m^2}]$
dV	Differential reference volume	$[\mathrm{m^3}]$
dv	Differential spatial volume	$[\mathrm{m^3}]$
J	Jacobian of deformation	$[\mathrm{m^3}]$
t	Time	[s]
U	Shock wave speed	$[\mathrm{m\,s^{-1}}]$
U_p	Pressure shock wave speed	$[\mathrm{m\,s^{-1}}]$

Tensors

$\boldsymbol{\tau}$	Kirchoff stress	[Pa]
$\boldsymbol{\varepsilon}$	Spatial strain	$[-]$
$\boldsymbol{b}$	Left Cauchy-Green stress	[Pa]
$\boldsymbol{C}$	Right Cauchy-Green stress	[Pa]
$\boldsymbol{I}$	Identity tensor	$[-]$
$\boldsymbol{\sigma}$	Cauchy stress	[Pa]
$\boldsymbol{F}$	Deformation gradient	$[-]$
$\boldsymbol{G}$	Gradient operator	$[-]$
$\boldsymbol{H}$	Co-factor of deformation gradient	$[-]$
$\boldsymbol{P}$	First Piola-Kirchoff stress	[Pa]
$\boldsymbol{S}$	Stabilisation matrix	$[-]$
$\boldsymbol{S}$	Second Piola-Kirchoff stress	[Pa]

Vectors

δ	Gap	[m]
$\boldsymbol{\mathcal{F}}$	Vector of flux variables	$[-]$

$\boldsymbol{\mathcal{S}}$	Vector of source variables	$[-]$
$\boldsymbol{\mathcal{U}}$	Vector of conserved variables	$[-]$
$\boldsymbol{\Sigma}_{[\cdot]}$	Work conjugate	$[-]$
$\boldsymbol{E}_I$	Cartesian basis	$[-]$
$\boldsymbol{p}$	Linear momentum	$[\mathrm{kg\,m\,s^{-1}}]$
$\boldsymbol{u}$	Displacement	[m]
$\boldsymbol{\phi}$	Mapping function	$[-]$
$\boldsymbol{b}$	Body force per unit mass	[N]
$\boldsymbol{C}$	Reference area vector	$[\mathrm{m^2}]$
$\boldsymbol{c}$	Spatial area vector	$[\mathrm{m^2}]$
$\boldsymbol{N}$	Material outward unit normal	[m]
$\boldsymbol{n}$	Spatial outward unit normal	[m]
$\boldsymbol{t}$	Traction	[Pa]
$\boldsymbol{T}_\alpha$	Tangential vector	$[-]$
$\boldsymbol{v}$	Velocity	$[\mathrm{m\,s^{-1}}]$
$\boldsymbol{X}$	Reference or Lagrangian position	[m]
$\boldsymbol{x}$	Spatial or Eulerian position	[m]

Superscripts

$[\cdot]^{Ave}$	Average of variable $[\cdot]$
$[\cdot]^{A}$	Variable associated with Body A
$[\cdot]^{B}$	Variable associated with a boundary or Body B
$[\cdot]^{C}$	Variable associated directly with a discontinuous interface
$[\cdot]^{I}$	Variable associated with an internal variable
$[\cdot]^{L}$	Variable associated with left hand side of a discontinuous interface
$[\cdot]^{max}$	Maximum of variable $[\cdot]$

$[\cdot]^{min}$	Minimum of variable $[\cdot]$
$[\cdot]^{R}$	Variable associated with right hand side of a discontinuous interface
$\hat{[\cdot]}$	Deviatoric component

Subscripts

$[\cdot]_0$	$[\cdot]$ variable in initial/reference configuration
$[\cdot]_e$	Elastic decomposition of $[\cdot]$
$[\cdot]_H$	Hugoniot state of $[\cdot]$
$[\cdot]_i$	$[\cdot]$ belonging to node i
$[\cdot]_j$	$[\cdot]$ belonging to node j
$[\cdot]_p$	Plastic decomposition of $[\cdot]$
$[\cdot]_x$	Spatial X-directional component of a vector
$[\cdot]_{ij}$	$[\cdot]$ belonging to edge connecting node i to node j
$[\cdot]_n$	Normal component of $[\cdot]$
$[\cdot]_t$	Tangential component of $[\cdot]$
$[\cdot]_{xX}$	XX-directional component of a two-point tensor

Part I

Preliminaries

Chapter 1

Introduction

1.1 Motivation

Modern engineering practices are consistently pushing the boundaries of innovation and technological development aided by the increase in computational power over the past few decades. This has given rise to an increasing reliance on Computer Aided Engineering (CAE) throughout the design and manufacturing process, becoming a staple of modern engineering practices, acting as a bridge between theoretical calculations and prototype testing. CAE is the encompassing term used to describe computer algorithms designed to solve the governing differential equations for an engineering problem, such as aerodynamics or stress analysis. CAE simulations save industry time and money by giving the design engineers an insight into the behaviour of the geometry, for the given problem, in a shorter time frame than expensive prototype testing, meaning faster design iterations and improvements for a lower cost than prototype testing alone. With this increased reliance on CAE, fast, robust and accurate computational algorithms are required to ensure a reliable and realistic solution to the governing equations, is obtained within the desired time frame for any given engineering problem. Computational engineering covers a wide array of engineering fields which depend on distinctive governing physics each with unique challenges in obtaining fast, robust and accurate results.

One such field is that of Computational Solid Dynamics (CSD), which is concerned with the evolution of solid materials over time, subjected to specified initial and boundary conditions resulting in body displacement and material deformation. Obtaining a solution for CSD problems becomes increasingly complex when considering materials that deform significantly (large strain) or by considering additional physics such as thermal effects. A particular area of interest within CSD, that significantly increases the com-

plexity of the problem, is that of contact or impact. Contact or impact scenarios play a crucial role in a number of engineering problems across a wide spectrum of industries driving the desire for fast, robust and accurate computational methodologies to aid in the innovation and development within these industries.

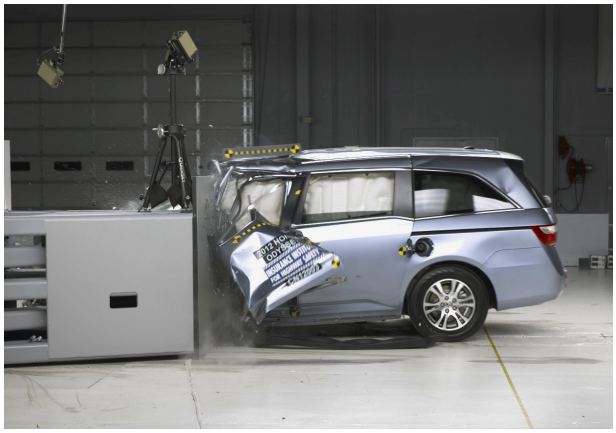
1.1.1 Practical Applications

Computational simulation of large strain contact dynamics can be applied to a wide range of scenarios and plays a significant role in numerous industries. Some example applications, shown in Figure 1.1 include; crash simulation [1, 2] in the automotive industry, bird strike simulation [3] in the aviation industry, hypervelocity impacts such as ballistic impacts or asteroid strikes in defence and space industries, surgical simulation [4, 5] in the biomedical industry, material manufacture such as shot peening [6] or cold-rolling [7] and more recently, the design of locomotive soft robots [8] which could be subjected to adverse terrains and environmental conditions.

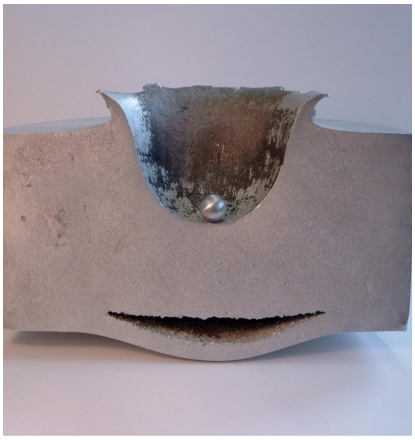
Each of these scenarios can benefit from computational simulation; whether it is during the design process, as slight changes of each iteration in the design can result in a significantly different deformation behaviour; in surgical training where practising on patients can be life threatening; or material manufacture where optimisation can be challenging. Investigating large deformations experimentally can be impractical or costly across all industries due to the complex materials or environmental conditions. Large strain computational contact dynamics proves challenging to accurately simulate due to the inherent characteristics of contact scenarios.

1.1.2 Characteristics of Contact Scenarios

Contact-impact describes the behaviour when two surfaces touch, whether that is two surfaces of the same body or of independent bodies. This produces a discontinuous contact interface between the two surfaces and an associated internal stress (shock) waves which emanate away from the contact interface [12, 13]. This contact region is subjected to a series of interface conditions that describe the interaction between the bodies in terms of the impenetrability (Kinematic), momentum balance of tractions (Kinetic) and persistence of the interface (Unitary). These conditions, that must be satisfied, combined with the propagation of shock waves proves challenging to accurately solve computationally due to various bottlenecks related to traditional computational methods implemented in commercial solvers.

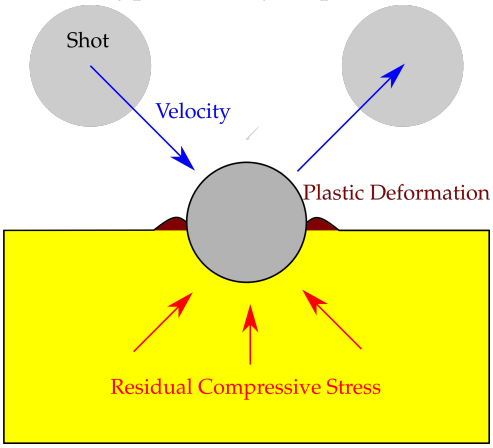


(a) Crash Testing [9]



(b) Hypervelocity Impacts [10]

(c) Practice Surgical Procedures [11]



(d) Shot-Peening Process

Figure 1.1: Example practical applications

1.2 Overview of Traditional Methodologies

The modelling of large strain contact dynamic problems proves challenging due to various aspects of the computation. In this section traditional methodologies will be explored with focus on methodologies implemented in commercial solvers.

Commercial solver methodologies for large strain contact dynamics can be broken into two aspects. The first aspect, is the underlying computational method used to approximate the governing equations for the solid dynamic behaviour of the problem. This aspect is not solely associated to contact scenarios but forms the backbone of the computation for contact mechanics, as a result any challenges or limitations to this methodology is carried into contact scenarios. The second aspect, is the identification and resolution of the contact region. This aspect forms the main focus of this research and can be further categorised into a series of stages, these include contact search algorithms, solving the discontinuous problem at contacting interfaces (contact resolution) and accurate modelling of shock wave propagation (shock capture).

The methodologies used to solve the contact stages vary dependant on the type of time integrator, either explicit or implicit, and are used to express the dynamic behaviour of the contact scenario. Explicit methods are traditionally fast and simple to implement however they are conditionally stable requiring a restriction on the time step to ensure stability. Implicit methods on the other hand, are generally computationally intensive but more accurate than explicit methods and unconditionally stable, therefore they have no restrictions imposed on the time step [14, 15]. In dynamic shock capture scenarios, such as impact, explicit methods are preferred to accurately capture the stress wave propagation due to their inherent need for small time steps [15]. Since the aim of this research is focused on fast and accurate computational contact dynamic methodologies, explicit methods will be the primary focus, however due to their accuracy and popularity, implicit methods will also be explored here. Each aspect and contact stage will be introduced in the following sections.

1.2.1 Lagrangian Solid Dynamics

To approximate contact dynamic problems, an underlying computational method is required to discretise and solve the solid dynamic physics. In explicit CSD, commercial solvers such as Abaqus/Explicit [16] and LS-DYNA [17], utilise computational methods traditionally based on standard displacement based formulations most notably, the Finite Element Method (FEM) [12, 18, 19] for spatial discretisation and a Newmark based method [20, 21] for explicit time integration. These methodologies have emerged as the preferred CSD methods for commercial solvers in part due to compatibility with unstructured tetrahedral mesh generators [22, 23], providing fast and efficient discrete meshes. The combination of these methodologies however comes with a series of shortcomings when attempting to scale the procedure to complex physics and material models such as plasticity or large strain contact dynamics as follows:

1. Second order accuracy for displacement but a first order accuracy for stress [24, 25].
2. Spurious oscillations in the vicinity of shocks such as contact or impact [26, 27].
3. Difficulty with bending dominated scenarios [28, 29].
4. Hour-glassing and pressure checker boarding near the incompressible limit [30].
5. Shear and volumetric locking [31, 32].

Alleviating these shortcomings have been of particular interest in the CSD community in order to take advantage of the popularity of FEM and build on already developed commercial software. For example, higher order interpolation methods which improve

the accuracy of standard methods through higher order elements and basis functions compromising efficiency [12, 18], reduced integration techniques which improve computational efficiency while using higher order elements such as hexahedrals [12, 18], mean dilation methods to prevent locking [33], average nodal pressure elements which approximates the pressure at each node to alleviate pressure checker boarding [28, 29, 34]. A selection of these methodologies have been introduced into commercial software in order to address these shortcomings, improving the accuracy and efficiency of traditional solvers widely used in industry. These methods, however, only partially address each shortcoming as a result these shortcomings can manifest in scenarios with complex physics, geometries or material models, such as contact, due to the inability to discretise the domain accurately and efficiently for higher order elements.

1.2.2 Contact Mechanics

To accurately model contact mechanics within CSD solvers, additional procedures are required for the identification and resolution of contact regions for single or multiple bodies. There are three main areas to consider, presented here for application to traditional FEM solvers, in the order required for contact algorithms: first are search algorithms, required to identify bodies and regions that will potentially be in contact; second are contact resolution methods that approximate and enforce the contact interface conditions; lastly are shock capture methods that are required to accurately capture the stress waves generated by the contact discontinuity. Each of these areas for contact mechanics will be discussed in this section.

Search Algorithms

Search algorithms are used each time or load increment to determine first, whether a body is potentially going to be in contact with another body (spatial/global search), then which surfaces (elements and nodes) are possibly in contact requiring contact resolution (local search) [15]. For scenarios where large deformations occur this can prove computational expensive as treating all boundary surfaces as potential contact surfaces with other bodies, or with itself, require excessive computations every time increment [35]. By splitting the approach into global and local search, the algorithms for determining contact can be respectively improved by reducing unnecessary calculations on surfaces that are not in proximity with another surface. This plays a significant role in contact scenarios where the type of contact and contact region can be determined prior to simulation, significantly simplifying the problem and saving computational time. While search algorithms form an integral part of contact mechanics, detailed evaluation and discussion of popular methods are outside the scope of this research, with

a brief overview described in this section for completeness, only scenarios where the contact region are known *a priori* will be considered in later numerical examples.

Global search methods are designed to be fast and efficient, traditionally by representing bodies as a coarse description such as body based cells, adaptive grid or Octree methods [35–38] this reduces the number of computations required for the contact resolution algorithms. Search algorithms become increasingly complex when considering phenomena such as self contact or the fragmentation of bodies for example see methods presented by Benson and Hallquist [39] for single surface self contact and Kane et al. [40] for fragmentation of bodies. After the global search is complete and potentially contacting regions identified, the local search on each contact region must be conducted.

For small displacement contact interfaces the local search only needs to be conducted once, after a contact region has been identified either by the global search or set prior to the simulation since it can be assumed the mesh information will remain unchanged throughout the simulation. On the other hand for large displacement contact interfaces such as where sliding is present, the local search needs to be repeated each time step, if the geometry has moved, resulting in additional computational time. This proves to be an important area of research for search algorithms [15, 35, 36], with recent advances these search algorithms have become more efficient and widely used [41–43]. Once the contact regions have been identified, either through the global search or designated prior to the simulation, the contact region must be resolved each time step.

Contact Resolution

Contact resolution is the general term used to describe how the contact conditions are solved and enforced along the contact region, across multiple bodies or in self contact. Contact resolution proves challenging to solve computationally with numerous different methodologies to address the contact interface problem. Contact resolution can be split into two important stages, first is to conduct the local search and determine whether the potential contact pairs are in contact, and if so to solve and enforce the contact constraints based on the contact region discretisation. The choice of contact resolution method varies depending on the contact scenario since each have different accuracy, stability and robustness properties [15].

The type of contact resolution method can be characterised into three types depending on the contact discretisation. Each method first requires a local search method principally based on the identification of the nearest neighbour [15] to be conducted prior to solving the contact constraints. The three core discretisations are known as, Node-To-Node (NTN), Note-To-Segment or Note-To-Surface (NTS) and Segment-To-Segment or

Surface-To-Surface (STS). NTN contact discretisation such as that proposed by Francavilla and Zienkiewicz [44], is a method where the closest neighbouring nodes on a contact pair are identified for evaluation and enforcement of the contact constraints. This method is only traditionally used in one dimensional or small displacement scenarios where the contact interface has identical matching meshes. NTS such as proposed by Hughes *et al.* [45] are two and three dimensional equivalent discretisation methods where the slave node of a contact pair identifies the closest boundary element (segment) of the master. This method enforces the contact constraints at the slave nodes thus preventing penetration of these nodes, however this does not apply to the master nodes and is known as a one (single) pass NTS method. The one pass NTS method is the most commonly used contact discretisation approach in commercial solvers due to its versatility with large displacement problems. This method can also be used to enforce the constraints at the master nodes and is known as two (double) pass method [46–48] however this can over restrict the contact region [49]. Lastly the STS contact discretisation, such as that proposed by Zavarise and Wriggers [50], can be used in two and three dimensions where the slave boundary surface identifies the nearest master boundary surfaces. This method is traditionally used for contact resolution methods that enforce the contact constraints at the slave integration point of the contact surface but can lead to penetration of the boundary nodes.

With each of these contact discretisations, a method of solving the contact constraints is required. The most popular solution methods implemented in commercial solvers are the penalty method and Lagrange multiplier methods each with unique advantages and disadvantages for solving the contact constraints and can be applied to any contact discretisation depending on the required accuracy and stability.

The penalty method consists of imposing the impenetrability condition through a penalty normal traction along the contact surface which allows for some interpenetration [12] based on a user defined parameter. Therefore the impenetrability conditions is only approximately satisfied and does not enforce the continuity of velocities. Due to this the penalty method can generate excessive interpenetration which can be unpredictable and potentially generate stiff, ill-conditioned systems that require, to maintain stability, very small time steps [12, 51]. Even with these disadvantages, the penalty method is the preferred constraint solution method for explicit solvers due to its simplicity and scheme compatibility for shock capture scenarios [52].

The Lagrange multiplier method, on the other hand does not allow for interpenetration by exactly enforcing the impenetrability condition through additional solved variables (Lagrange multipliers). The addition of these variables, significantly increases contact accuracy, however introducing an additional contact variable increases computational

time and can generated a system which is not positive-definite with a zero diagonal term [19, 53]. Lagrange multipliers generally require computationally expensive implicit solvers [54] or are adapted to explicit time schemes by utilise localised iterative algorithms to solve the boundary constraints, such as the kinematic predictor-corrector schemes discussed in [55–57]. Lagrange multipliers are generally preferred in quasi-static or implicit scenarios where capturing shock wave propagation is not a priority [15].

Accuracy and efficiency of contact resolution methods, rely heavily on the underlying computational method, type of contact discretisation and solution method of the contact constraints. Therefore, obtaining an accurate and realistic explicit contact response can prove challenging with each method requiring a compromise in either accuracy or efficiency depending on the contact scenario.

Shock Capture

When a discontinuity forms, such as a contact interface, an associated shock wave propagates from the contact interface throughout the domain. In CSD, the shock wave created from contact propagates throughout the solid as a pressure wave (normal) and a shear wave (tangential) [13]. Shock capture proves challenging in CSD due to the use of traditional explicit linear finite elements which can generate spurious oscillations in the solution if not taken into consideration by the numerical method. To counter this issue, many commercial solvers implement a method known as artificial bulk viscosity first developed by von-Neumann and Richtmyer [58]. This ad-hoc regularisation effectively smooths the solution near the shock by introducing an artificially created viscosity resulting in a stable solution but sacrifices accuracy near the shock. This can be an issue when dealing with complex material models or physics that require accurate prediction of the stress wave propagation in order to generate the correct material response. As a result explicit methods are preferred to accurately capture shock wave propagation with a suitably small time step [15].

1.2.3 Summary of Commercial Contact Methodology

To summarise the above sections, commercial solvers for explicit large strain contact dynamic applications typically utilised standard displacement based FEMs, requiring higher order elements (such as hexahedrals) to discretise a solid domain. The contact region is then modelled using a standard NTS contact discretisation with the penalty method or a predictor-corrector Lagrange multiplier method to solve and enforce the contact conditions. By utilising this combination of methodologies, the commercial solvers can generate realistic solutions to contact scenarios as a compromise of accu-

racy, due to the contact resolution methods, and efficiency, due to the higher order elements, this however limits commercial solvers contact capabilities. The explicit contact dynamic challenges can be summarised as follows:

1. **Complex geometries**; can not be sufficiently discretised using hexahedral elements requiring unstructured tetrahedrals leading to locking or pressure checkerboarding.

2. **Complex material models**; require significant computational cost due to the solver requiring higher order of elements therefore more stress evaluations.

3. **Shock capture**; proves challenging due to the spurious oscillations generated in the vicinity of contact regions stemming from the underlying computational method.

4. **Contact Accuracy and Efficiency**; the penalty method cannot accurately model contact without very small time steps but are very efficient, while Lagrange multipliers provide greater accuracy but require more computational intensive iterative schemes.

5. **Contact constraint enforcement**; the impenetrability condition can only be enforced at the slave node using the NTS approach when using penalty or predictor-corrector methods.

With the increasing reliance on CAE in industry, complex geometries and material models are now common place, therefore a computational method that can efficiently and accurately model contact dynamics of complex geometries or material models with accurate and efficient enforcement of the contact constraints for unstructured tetrahedral meshes would be highly beneficial to industry. As a result, significant research focus has been directed towards these challenges and current state of the art contact methodologies will be presented in the next section.

1.3 State of the Art Contact Methodologies

The modelling of large strain contact dynamic problems proves challenging due to various aspects of the computation, the current state of the art methodologies will be described and discussed in this section in order to address the limitations of commercial solvers.

Similarly to traditional methodologies, current state of the art methods can be broken into two categories; 1) the underlying computational method, 2) the identification and resolution of the contact region. Since this research is focused on contact methodolo-

gies, the state of the art computational methods will not be discussed however see references [59–61] for examples of advanced mixed formulation for the FEM applied to contact scenarios.

Commercial contact methods, as discussed in Section 1.2.2, have limitations primarily in relation to accuracy in terms of the penalty method and efficiency in terms of the predictor-corrector Lagrange multiplier method. Numerous extensions and variations to these methods have been proposed over the years, in order to extend these methods to complex problems as well as improve the robustness and accuracy of these methods. The current state-of-the-art contact methodologies are known as the mortar method, Nitsche method and the less common Riemann solver approach, these methods are yet to be adopted by commercial software with an introduction discussed here.

1.3.1 Mortar Method

The mortar method was first proposed for static contact applications in [62–64] and later for quasi-static large deformations by Puso *et al.* [65], it is a particular approach of STS contact discretisation with constraint enforcement through the use of Lagrange multipliers at the integration points. This is achieved through the projection of intersecting elements of the contact surface and sub-dividing the integration scheme to accurately integrate the contact contributions [66]. This method has shown popularity due to its robustness and accuracy for enforcing the contact constraints for non-matching meshes as an improvement over standard NTS approaches used in commercial solvers [49, 65]. However, the popularity and research focus of the mortar method has been directed towards quasi-static [67–79] and implicit [75, 80–84] contact applications due to the underpinning Lagrange multiplier formulation. More recently, the mortar method has been applied using an explicit time scheme by Otto *et al.* [85], this application of the mortar method results in exact enforcement of the impenetrability condition in an explicit scheme however it requires a coupled contact layer, increasing the computational complexity, and an ad-hoc regularisation to smooth the contact transitions reducing the accuracy of the solution at contact initialisation and separation.

1.3.2 Nitsche Method

While the Nitsche method [86, 87] was first proposed for quasi-static contact applications in [88] and extended to quasi-static friction and large deformations in [66, 89–92]. In this methodology the contact constraints are consistently applied through a traction vector computed from the bulk stress tensor with a suitably large penalty parameter [93]. This approach allows the contact constraints to be applied without any additional variables such as the Lagrange multiplier approach and can be described as a particu-

lar approach of NTS where the contact constraints are applied at the integration points. This method was later applied to dynamic applications in [94, 95] for various implicit time schemes and in [96] for explicit time schemes. It can be observed in both time schemes that the Nitsche method is accurate for displacements but generates oscillations in the stress history during contact, likely due to the underlying FEM, without the use of regularisation techniques or any additionally solved variables. This suggests that the Nitsche method is more computationally efficient than mortar/Lagrange multipliers but improving the stress accuracy with this method is an open area of research.

1.3.3 Riemann Solver

Lastly, is the less common Riemann solver approach, traditionally used in applications related to discontinuous interfaces such as shock waves in Computational Fluid Dynamics (CFD) or numerical schemes (for example cell-centred finite volume method) for full discussion in this context see references [97, 98]. The Riemann solver approach derives the numerical fluxes at a discontinuous surface based on the Rankine-Hugoniot jump conditions. This method results in explicit expressions for the contacting fluxes between discontinuous surfaces that can be applied directly into a numerical scheme. Recently, this approach has been applied to contact dynamics by Abedi and Haber [54], and later improved upon in [99], utilising spacetime discontinous Galerkin discretisation. By utilising this method the impenetrability condition is strongly enforced but requires ad-hoc regularisation to smooth contact transitions. This methodology, while promising, has not received much attention due to the requirement for the computational method to be formulated in the context of fluxes, i.e. linear momentum or velocity, while traditional solid dynamic methods are formulated in terms of displacements. As a result this method cannot easily be adopted by commercial solvers however could prove beneficial in contact dynamic applications with an appropriate computational scheme.

1.3.4 Summary of State of the Art Methodologies

In terms of explicit contact dynamics, solving the contact conditions accurately and efficiently proves challenging as the impenetrability condition cannot be strongly enforced with standard displacement based finite elements alone. As a result obtaining accurate results is an open research topic with continuous advancements in contact resolution methods such as mortar or Nitsche methods and emerging methods such as Riemann solvers. These methodologies are yet to be adopted by commercial solvers leading to advancement of Open-source solvers created by the research community with alternate methodologies to address the shortcomings of traditional methods, one such method

will be the primary focus of this research and introduced in the next section.

1.4 Proposed Methodology

More recently, a novel methodology has emerged for CSD known as the first order framework for large strain solid dynamics, first proposed by Lee *et al.* [100] and extended in recent years for a variety of discretisation techniques and complex physics, see references [101–118]. This methodology has been formulated to address the shortcomings of traditional displacement based methods, as described in Section 1.2.1, and is achieved through an alternate first order continuum formulation. This formulation is based on the balance principle of linear momentum and additional triplet of kinematic conservation equations resulting in a formulation of conservation equations similar to the conservation formulation in fluid dynamics. By utilising this continuum formulation, the methodology provides the flexibility to utilise CFD solution and discretisation methodologies directly for CSD problems, in order to alleviate the shortcomings related to traditional methods.

In this research, the first order framework will be harnessed as the underlying computational method and discretised using the well known CFD discretisation, Vertex Centred Finite Volume Method (VCFVM). This computational formulation will be extended to explicit large strain contact dynamics by applying traditional CFD shock capture methodologies to resolve the contact regions, namely a generalised Riemann solver. By utilising this combination of methodologies a series of novelties are achieved.

1.4.1 Novelties of Proposed Method

The novelty of this research is to apply and extend the first order framework to contact-impact problems through the use of VCFVM discretisation[1] with an appropriate Riemann solver to solve and enforce for the contact constraints. This aims to have several significant benefits over standard displacement based methods implemented in commercial solvers:

1. Strong enforcement of both contact conditions; namely impenetrability condition and traction condition through the additional kinematic conservation equations and expressed explicitly by application of a Riemann solver to the contact interface.

2. Faster computational time; high order elements such as hexahedral or shell el-

[1]While the use of finite volume methods in solid mechanics is not novel, it is uncommon compared with traditional finite elements for a comprehensive review see Cardiff and Demirdžić [119].

ements will not be required at the interface or throughout the domain by using the VCFVM thus allowing for traditional unstructured tetrahedral elements. This significantly reduces the number of stress evaluations required as the number of integration points is reduced hence benefits more complex constitutive models such as plastic or visco-elastic material models.

3. Increased shock capture accuracy; through using an appropriate Riemann solver for the contact interface and throughout the domain. This does not require ad-hoc regularisation, however an appropriate limiter is required.

4. Non-linear Riemann solver for solid dynamics; introduction of a novel Riemann solver for non-linear material models focused on modelling large impact velocities.

5. Increased overall accuracy; second order accuracy for both velocity and stress is achieved across the contact region by utilising the first order framework for the underlying CSD method. This methodology also provides, a smooth, locking free solution without pressure instabilities and ensures the satisfaction of the Coleman-Noll procedure at a semi-discrete level.

1.5 Thesis Scope and Outline

The primary objective of this thesis is to develop a robust contact dynamic solver through the extension of the first order framework to explicit large strain contact dynamic scenarios. It is important at this stage to emphasise that the theory and results presented throughout this research consider contact and impact scenarios where fracture does not occur and contact surfaces are known *a priori* therefore global search algorithms[2] are not required. This thesis is structured as follows:

- **Chapter 2** outlines the underlying theory first for smooth (continuous) non-linear continuum mechanics used to describe the motion and deformation of a single body, which forms the backbone of this thesis. The first order framework for large strain solid dynamics is introduced including relevant polyconvex constitutive models. The theory is then extended to non-smooth problems and example derivations of the general jump condition for non-smooth problems and simple material wave speeds are demonstrated and discussed. This theory is then applied to Hugoniot shock wave speeds from experimental observations.

- **Chapter 3** introduces the theory of continuum contact mechanics and discusses the key contact conditions namely the impenetrability, traction and unitary con-

[2]Refer to Han *et al.* [38] for examples of global search algorithms.

tact condition. Resolution of the contact-impact interface conditions for contact-stick, frictionless contact and coulomb frictional contact are then explored at a continuum level through the derivation of two distinct solid dynamic Riemann solvers, acoustic and non-linear, based on the jump conditions. With these derivations, the one-dimensional exact solution is derived for the linear elastic impact of two identical bars. The Hamiltonian energy is then introduced in order to assess the entropy production of the numerical scheme.

- **Chapter 4** describes and applies the numerical method used to discritise the first order framework. For the spatial numerical scheme, vertex centred finite volume method is explored while a two-stage time variation diminishing Runge-Kutta time integrator will be discussed to explicitly evolve time. The entropy production of the semi-discrete system will be examined to assess the stability of the numerical scheme.

- **Chapter 5** presents the computational implementation of the proposed method within open source software "OpenFOAM" [120]. The software will be introduced followed by crucial algorithms of the purpose built code implemented as part of this research.

- **Chapter 6** presents the one-dimensional problems used to benchmark and assess the numerical results from the proposed method using MATLAB. This includes, the impact of: two identical linear elastic bars, two dissimilar linear elastic bars, two identical Hugoniot bars and two dissimilar Hugoniot bars.

- **Chapter 7** presents two and three dimensional problems addressing crucial aspects of contact problems; spurious oscillations, pressure checker-boarding and non-matching contact interfaces. These aspects are investigated through a variety of problems and compared with standard displacement based finite element software, Abaqus/Explicit [16]. First, in two dimensions the impact of two compressible rings are investigated followed by the impact of two nearly incompressible rectangular bars. Then in three dimensions, the traditional benchmark plasticity problem known as the Taylor bar impact is compared with published results, followed by the impact of a torus with a rigid surface which is investigated for elastic and plastic material models followed by investigation of the accuracy of non-matching contact interfaces through a problem consisting of two nearly incompressible bars similar to that of the two dimensional problem. Lastly, a simplified car crash is presented to demonstrate a practical example.

- **Chapter 8** summarises and concludes this research with proposed avenues of future work.

Part II

Lagrangian Contact Dynamics

Chapter 2

Continuum Equations for Solid Dynamics

2.1 Preliminaries

In order to develop the computational method as part of this research, this chapter introduces the fundamental theory for non-linear continuum mechanics used to describe the motion and deformation of a continuum. This theory is required to describe the underlying large strain solid dynamics, before extension to contact dynamics. The chapter begins by outlining the kinematics and problem variables required to describe the motion and deformation of a continuum in Section 2.2. Section 2.3 goes on to use these descriptions to formulate the governing equations for solid dynamics namely the balance principle for linear momentum and additional kinematic conservation equations, that will form the basis of the first order conservation framework described in Section 2.4. In order to complete the system of equations for smooth solid dynamics, the constitutive models considered in this research are described in Section 2.5. The extension of the theory to consider non-smooth problems is then discussed in Section 2.6 where the derivation of the general jump conditions, simple wave speeds and Hugoniot shock wave speeds for non-linear elasticity are demonstrated to complete the solid dynamic system for discontinuous problems.

2.2 Kinematics

Within the field of large strain contact mechanics the deformation and motion of a material are typically expressed using the theory of non-linear continuum mechanics, which will be described in this section following the notation from Bonet *et al.* [21, 33].

2.2.1 Problem Variables

To describe the deformation of a continuum, first the kinematics must be examined. Consider the motion and deformation of an enclosed domain Ω_0 as shown in Figure 2.1.

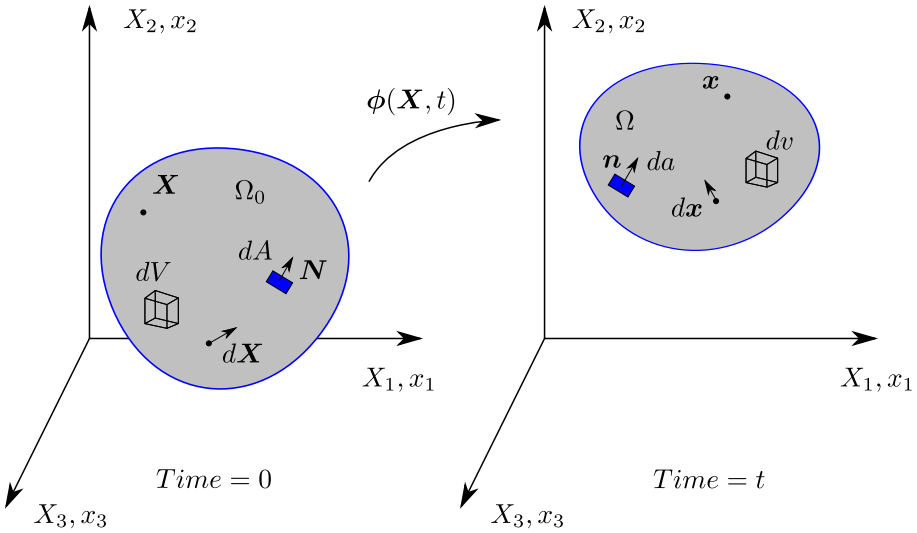


Figure 2.1: An enclosed domain subjected to deformation mapping $\boldsymbol{\phi}(\boldsymbol{X}, t)$ from the reference configuration (Left) to current configuration (Right)

Figure 2.1 describes the motion and deformation of a continuum particle from material (reference) or Lagrangian position $\boldsymbol{X}$ to spatial or Eulerian position $\boldsymbol{x}$ through mapping $\boldsymbol{\phi}$ which is a function of reference position $\boldsymbol{X}$ and time t such that $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t)$. This continuum model assumes particles are bijective, mapping, $\boldsymbol{\phi}$ is invertible and sufficiently smooth (i.e. the derivatives can be calculated via a Taylor expansion) and higher order terms are neglected. This allows mapping $\boldsymbol{\phi}$ to describe the change in a fibre of differential length within the domain from the reference configuration to spatial configuration through a tensor known as the deformation gradient $\boldsymbol{F}$. The deformation gradient, also known as the fibre map, is a two-point tensor such that $d\boldsymbol{x} = \boldsymbol{F} d\boldsymbol{X}$ and can be defined as:

$$\boldsymbol{F} = \boldsymbol{\nabla}_0 \boldsymbol{\phi}\left(\boldsymbol{X}, t\right), \tag{2.1}$$

where $[\boldsymbol{\nabla}_0]_I$ is the material gradient defined as $\frac{\partial}{\partial X_I}$. From this definition of a fibre map, a volume map can then be defined. The volume map, also known as the Jacobian, relates a differential volume within the domain from the initial configuration dV to spatial configuration dv such that $dv = JdV$ and is defined as:

$$J = \det \boldsymbol{F}; \tag{2.2}$$

J must always be greater than 0. Next an area map, also known as co-factor, can be

defined which relates a differential area within the domain from the reference configuration dA, with material normal $\boldsymbol{N}$, to spatial configuration da, with spatial normal $\boldsymbol{n}$, through the co-factor/adjoint tensor $\boldsymbol{H}$. The area map is traditionally related to the fibre map as:

$$\boldsymbol{H} = J\boldsymbol{F}^{-T}. \tag{2.3}$$

Alternatively, the area map can be expressed through the use of the tensor cross product, represented by a bold cross product symbol $\boldsymbol{\times}$, as discussed in [106]. This results in the equivalent definition for the area map as:

$$\boldsymbol{H} = \frac{1}{2}\boldsymbol{F} \boldsymbol{\times} \boldsymbol{F}, \tag{2.4}$$

where the tensor cross product between two two-point tensors is defined as $(\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B})_{iI} = \mathcal{E}_{ijk}\mathcal{E}_{IJK}A_{jJ}B_{kK}$, for a complete list of properties refer to Bonet *et al.* [121]. Lastly, the velocity $\boldsymbol{v}$ can be defined in terms of the deformation mapping as the partial derivative with respect to time t:

$$\boldsymbol{v} = \frac{\partial \boldsymbol{\phi}(\boldsymbol{X}, t)}{\partial t}. \tag{2.5}$$

From this description of the motion and deformation of a continuum the governing equations can then be described for solid dynamics.

2.3 Governing Equations

Solid dynamic problems are governed by a set of physical laws known as the balance principle of linear momentum and conservation of mass which are traditionally used to formulate the basis of CSD methodologies. In this section, these governing equations will be described in the context of Lagrangian solid dynamics and additional unconventional kinematic conservation equations introduced in order to express the complete the first order conservation framework for large strain solid dynamics.

Remark 2.1. *The conservation of mass simply ensures that the mass of a system remains constant. For Lagrangian formulations this conservation equation does not need to be considered as an unknown variable as it is inherently fulfilled [106].*

2.3.1 Linear Momentum Balance Principle

The balance principle of linear momentum ensures the linear momentum of the system is conserved in the presence of any additional body forces and is expressed in the

reference configuration, in global form per unit of undeformed volume as:

$$\frac{d}{dt}\int_{\Omega_0}(\rho_o \boldsymbol{v})\,dV = \int_{\Omega_0}\rho_0 \boldsymbol{b}_0 dV + \int_{\partial\Omega_0}\boldsymbol{t}_0 dA, \tag{2.6}$$

where $\rho_0 \boldsymbol{v} = \boldsymbol{p}$ is the linear momentum per unit undeformed volume, $\boldsymbol{b}_0$ is the body force per unit mass, $\boldsymbol{t}_0 = \boldsymbol{P}\boldsymbol{N}$ is the traction vector with $\boldsymbol{P}$ defined as the first Piola-Kirchhoff stress tensor and $\boldsymbol{N}$ defined as the material outward unit normal. The local equilibrium equation can be expressed by applying the divergence theorem to the right hand side of Equation (2.6) to give:

$$\frac{\partial(\rho_0 \boldsymbol{v})}{\partial t} = \mathrm{DIV}\,\boldsymbol{P} + \rho_0 \boldsymbol{b}_0, \tag{2.7}$$

where DIV is the divergence operator carried out in the reference configuration.

2.3.2 Kinematic Conservation Equations

In order to formulate the first order framework for large strain solid dynamics initially described by Lee *et al.* in [100], the conservation of the deformation gradient is also required and can be simply extended to include the conservation of the area map and conservation of the Jacobian as described in [101] and [103] to create a set of kinematic conservation equations[1]. These kinematic conservation equations can be expressed in local form as:

$$\frac{\partial \boldsymbol{F}}{\partial t} = \mathrm{DIV}\,(\boldsymbol{v}\otimes\boldsymbol{I})\,; \tag{2.8}$$

$$\frac{\partial \boldsymbol{H}}{\partial t} = \mathrm{CURL}(\boldsymbol{v}\times\boldsymbol{F}); \tag{2.9}$$

$$\frac{\partial J}{\partial t} = \mathrm{DIV}\left(\boldsymbol{H}^T\boldsymbol{v}\right); \tag{2.10}$$

where CURL is the curl operator carried out in the reference configuration. The first kinematic conservation Equation (2.8) is the conservation of the deformation gradient $\boldsymbol{F}$. This conservation equation can be considered a generalisation of the continuity equation in fluid dynamics and is considered part of the framework to increase flexibility. By doing so, the appearance of locking in low order simulations can be alleviated [100]. The series of kinematic conservation equations can be extended to include the conservation of the area map $\boldsymbol{H}$ as shown in Equation (2.9). This definition can be derived by utilising the definition of the area map from Equation (2.4) by utilising the tensor cross product, as described in [106], resulting in a compact conservation equation. The last natural extension to the kinematic conservation equations is to consider

[1] If necessary, refer to [106] for the derivation of these conservation equations.

the conservation of the volume map J in Equation (2.10). The addition of this conservation equation further increases the flexibility of the framework particularly in nearly incompressible regime by alleviating pressure checker boarding [103, 105].

By introducing these kinematic conservation equations, in order to ensure the existence of a single-valued mapping ϕ, additional conditions must be satisfied known as the compatibility conditions[2]. These compatibility conditions apply to the evolution of the deformation gradient $\boldsymbol{F}$ and area map $\boldsymbol{H}$ defined as:

$$\text{CURL}\, \boldsymbol{F} = \boldsymbol{0}; \quad \text{DIV}\, \boldsymbol{H} = \boldsymbol{0}. \tag{2.11}$$

If compatibility conditions are not satisfied spurious oscillations can occur leading to an inaccurate solution or ultimately complete breakdown of the numerical scheme, therefore, these conditions must be carefully considered when applying the computational method. For the particular case of linear elasticity these conditions becomes the Saint-Venant compatibility conditions [122]. With these kinematic conservation Equations (2.8) - (2.10) and the balance principle for linear momentum in Equation (2.7), the first order framework for large strain solid dynamics can be formulated.

Remark 2.2. *The conservation of energy, also known as the first law of thermodynamics, plays an important role in temperature dependant problems ensuring that the energy of the system is conserved. In this research, only isothermal processes, where the temperature remains constant, are considered thus the conservation of energy is not required as part of the first order framework. It is useful however, to examine the conservation of energy for various aspects throughout this research including derivations and most importantly examining the consistency of results. The conservation of energy for an isothermal process is defined in local form as:*

$$\frac{\partial E_T}{\partial t} - \text{DIV}\left(\boldsymbol{P}^T \boldsymbol{v}\right) = \rho_0 \boldsymbol{b}_0 \cdot \boldsymbol{v}, \tag{2.12}$$

where E_T is the total energy, DIV *is the material divergence, $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor, $\boldsymbol{v}$ is the velocity field and $\boldsymbol{b}_0$ is the body force per unit mass. For the complete form of the conservation of energy, refer to Appendix A.*

---

[2]Compatibility conditions are also known as involutions.

2.4 First Order Framework for Large Strain Solid Dynamics

The balance principle of linear momentum and kinematic conservation equations can be combined into a set of conservation equations, in order to formulate the first order framework for large stain solid dynamics. In local form this can be expressed in compact form as:

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \sum_{I=1}^{3} \frac{\partial \boldsymbol{\mathcal{F}}_I}{\partial X_I} = \boldsymbol{\mathcal{S}}, \tag{2.13}$$

where $\boldsymbol{\mathcal{U}}$ is the vector of conserved variables, $\boldsymbol{\mathcal{F}}_I$ is the flux vector in the I-th material direction and $\boldsymbol{\mathcal{S}}$ is the vector of source terms as follows:

$$\boldsymbol{\mathcal{U}} = \begin{bmatrix} \rho_0 \boldsymbol{v} \\ \boldsymbol{F} \\ \boldsymbol{H} \\ J \end{bmatrix}; \quad \boldsymbol{\mathcal{F}}_I = - \begin{bmatrix} \boldsymbol{P}\boldsymbol{E}_I \\ \boldsymbol{v} \otimes \boldsymbol{E}_I \\ \boldsymbol{F} \times (\boldsymbol{v} \otimes \boldsymbol{E}_I) \\ \boldsymbol{H} : (\boldsymbol{v} \otimes \boldsymbol{E}_I) \end{bmatrix}; \quad \boldsymbol{S} = \begin{bmatrix} \rho_0 \boldsymbol{b} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ 0 \end{bmatrix}, \tag{2.14}$$

with Cartesian basis defined as:

$$\boldsymbol{E}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{E}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \boldsymbol{E}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{2.15}$$

Therefore, corresponding flux vector associated with material outward unit normal $\boldsymbol{N}$ is defined as:

$$\boldsymbol{\mathcal{F}}_{\boldsymbol{N}} = \boldsymbol{\mathcal{F}}_I N_I = - \begin{bmatrix} \boldsymbol{P}\boldsymbol{N} \\ \boldsymbol{v} \otimes \boldsymbol{N} \\ \boldsymbol{F} \times (\boldsymbol{v} \otimes \boldsymbol{N}) \\ \boldsymbol{H} : (\boldsymbol{v} \otimes \boldsymbol{N}) \end{bmatrix}. \tag{2.16}$$

The above hyperbolic system[3], combined with a well-posed constitutive model ensures the existence of real wave speeds at any state of deformation. This property is highly beneficial for the application to contact problems as it will ensure the robustness of the computational methodology. The framework presented in this section is valid for problems that have a smooth solution i.e. there are no shock waves (discontinuities) present. For the framework to be valid for non-smooth problems the framework must satisfy conditions known as jump conditions, these will be explored in Section 2.6.

[3]For proof of hyperbolicity refer to [106]

Remark 2.3. *In comparison, the standard displacement based formulation is derived by considering only the linear momentum balance principle from Equation* (2.7) *and reformulating in terms of displacement to give:*

$$\rho_0 \frac{\partial \boldsymbol{v}}{\partial t} - \text{DIV}\, \boldsymbol{P}\left(\boldsymbol{\nabla}_0 \boldsymbol{\phi}\right) = \rho_0 \boldsymbol{b}; \quad \textit{where} \quad \boldsymbol{v} = \frac{\partial \boldsymbol{\phi}}{\partial t}, \tag{2.17}$$

where ρ_0 is material density, $\boldsymbol{P}$ is the first Piola-Kirchhoff stress as a function of the gradient of the deformation mapping function $\boldsymbol{\phi}$ and $\boldsymbol{v}$ is the velocity field. In this second order formulation the only unknown variable is displacement, as a result solving this method using standard linear finite elements results in a second order solution for displacement but a first order solution in the derived stresses and strains [24, 25] and requires integration in time to evaluate the velocity. On the other hand, the first order framework circumvents these shortcomings by solving the conserved variables instead of displacement, which preserves the order of accuracy for the derived stresses and strains and solves velocity directly. Further benefits of the first order framework are contained within the application of the constitutive model that relates stress to strain.

2.5 Constitutive Models

With the governing equations now defined for the first order framework in Sections 2.3 and 2.4, the next stage is to define an appropriate constitutive (material) model that describes the relation between the stress and strain of the material. For large strain constitutive models, used in this work they are typically defined as an energy functional from which the stresses and strains are derived. Three models will be discussed in this section as they are required for the numerical examples presented in Part IV; nearly incompressible Mooney-Rivlin, nearly incompressible neo-Hookean and rate-independent von-Mises plasticity.

2.5.1 Nearly Incompressible Mooney Rivlin

The first material model considered, is the nearly incompressible Mooney Rivlin model. The Mooney Rivlin model is a hyperelastic constitutive model and is used here to introduce the concept of polyconvex constitutive models [106, 108, 117]. Polyconvex constitutive models are material models that satisfy the polyconvexity mathematical criteria [123] and are based on a strain energy functional that are a convex function of the components of the deformation gradient, area map and Jacobian [124]. The Mooney Rivlin model can be expressed as a multi-variable polyconvex strain energy functional

$W(\boldsymbol{F}, \boldsymbol{H}, J)$ as the decomposition of the deviatoric $\hat{W}(\boldsymbol{F}, \boldsymbol{H}, J)$ and volumetric contributions $\mathrm{U}(J)$ as [106, 108, 124]:

$$W(\boldsymbol{F}, \boldsymbol{H}, J) = \hat{W}(\boldsymbol{F}, \boldsymbol{H}, J) + \mathrm{U}(J); \tag{2.18}$$

where:

$$\hat{W}(\boldsymbol{F}, \boldsymbol{H}, J) = \zeta J^{-2/3}(\boldsymbol{F} : \boldsymbol{F}) + \xi J^{-2}(\boldsymbol{H} : \boldsymbol{H})^{3/2} - 3\left(\zeta + \sqrt{3}\xi\right); \quad \mathrm{U}(J) = \tfrac{\kappa}{2}(J-1)^2;$$

where ζ, ξ and κ are positive material parameters with κ as the bulk modulus and ζ, and ξ related to the shear modulus as $2\zeta + 3\sqrt{3}\xi = \mu$ in order to recover the Lamé coefficients for the small strain regime. From this strain energy functional the first Piola-Kirchhoff stress tensor can be expressed in terms of work conjugate stresses ($\boldsymbol{\Sigma}_{\boldsymbol{F}}$, $\boldsymbol{\Sigma}_{\boldsymbol{H}}$, Σ_J) with respect to ($\boldsymbol{F}$, $\boldsymbol{H}$, J) as:

$$\boldsymbol{P} = \boldsymbol{\Sigma}_{\boldsymbol{F}} + \boldsymbol{\Sigma}_{\boldsymbol{H}} \times \boldsymbol{F} + \Sigma_J \boldsymbol{H}; \quad \text{with} \quad \Sigma_J := \hat{\Sigma}_J + p; \tag{2.19}$$

where:

$$\boldsymbol{\Sigma}_{\boldsymbol{F}} := \tfrac{\partial \hat{W}}{\partial \boldsymbol{F}} = 2\zeta J^{-2/3}\boldsymbol{F}; \quad \boldsymbol{\Sigma}_{\boldsymbol{H}} := \tfrac{\partial \hat{W}}{\partial \boldsymbol{H}} = 3\xi J^{-2}(\boldsymbol{H} : \boldsymbol{H})^{1/2}\boldsymbol{H};$$

and:

$$\hat{\Sigma}_J := \tfrac{\partial \hat{W}}{\partial J} = -\tfrac{2}{3}\zeta J^{-5/3}(\boldsymbol{F} : \boldsymbol{F}) - 2\xi J^{-3}(\boldsymbol{H} : \boldsymbol{H})^{3/2}; \quad p := \tfrac{d\mathrm{U}}{dJ} = \kappa(J-1).$$

One crucial aspect of constitutive equations is frame indifference (objectivity) where the deformation measures must be independent of the reference frame. By formulating the strain energy functional in terms of $\{\boldsymbol{F}, \boldsymbol{H}, J\}$, objectivity is not necessarily met since the deformation gradient is not objective. Objectivity must therefore be proved through an analogous expression by reformulating the dependency of strain energy from Equation (2.18) with respect to $\boldsymbol{F}$ and $\boldsymbol{H}$ in terms of the right Cauchy-Green tensor $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ and its co-factor $\boldsymbol{G} = \boldsymbol{H}^T\boldsymbol{H}$ as demonstrated by Bonet *et al.* in [121] resulting in an analogous polyconvex expression for the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$. For the purpose of this research, the polyconvex strain energy functional expressed in terms of $\{\boldsymbol{F}, \boldsymbol{H}, J\}$ is more convenient in order to obtain a polyconvex expression for the first Piola-Kirchhoff stress tensor required as part of the first order framework. For the case of isotropic elasticity, the expression for the strain energy in Equation (2.18) can also be expressed through the re-definition of the standard invariants which is more suitable in the case of an isotropic polyconvex energy functional, described as $I_1 = \boldsymbol{F} : \boldsymbol{F}$, $I_2 = \boldsymbol{H} : \boldsymbol{H}$ and $I_3 = J^2$. This leads to a representation of the strain energy as $W(\boldsymbol{F}, \boldsymbol{H}, J) = w(I_1, I_2, I_3)$, for further details refer to Bonet *et al.* [121].

2.5.2 Nearly Incompressible Neo-Hookean

The second model considered is that of nearly incompressible neo-Hookean constitutive model. This is a hyperelastic model that is valid for large deformations and reduces to linear elasticity for small deformations. The polyconvex multi-variable energy functional can be defined based on the energy functional in Equation (2.18) by using values of $\zeta = \frac{\mu}{2}$ and $\xi = 0$ as:

$$W(\boldsymbol{F}, J) = \frac{\mu}{2}\left[J^{-2/3}\left(\boldsymbol{F} : \boldsymbol{F}\right) - 3\right] + \frac{\kappa}{2}\left(J-1\right)^2, \tag{2.20}$$

where μ is the shear modulus and κ is the bulk modulus. This energy functional is then used to obtain the first Piola-Kirchhoff stress from Equation (2.19) in terms of the deviatoric work conjugates as:

$$\boldsymbol{\Sigma}_{\boldsymbol{F}} = \mu J^{-2/3}\boldsymbol{F}; \quad \boldsymbol{\Sigma}_{\boldsymbol{H}} = \boldsymbol{0}; \quad \hat{\Sigma}_J = -\tfrac{\mu}{3}J^{-5/3}\left(\boldsymbol{F} : \boldsymbol{F}\right); \tag{2.21}$$

and volumetric pressure as:

$$p(J) = \kappa\left(J-1\right). \tag{2.22}$$

Capturing material behaviour accurately can prove challenging even for simple hyperelastic materials as shown by these two constitutive models and becomes increasingly complex when considering more complex models and physics such as visco-elasticity, thermo-elasticity[4] or thermo-plasticity. As the current research is focused on large strain isothermal contact, it is important to extend the current work to include elasto-plasticity and will be introduced in the following section for the specific case of rate-independent von-Mises plasticity material model.

Remark 2.4. *The simplest constitutive model is that of linear elasticity and is valid only for small deformations with infinitesimal strain for isotropic materials [33]. The energy functional for linear elasticity can be expressed in terms of spatial strain $\boldsymbol{\varepsilon}$ as:*

$$\psi(\boldsymbol{\varepsilon}) = \mu\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} + \frac{\lambda}{2}\left(\mathrm{tr}(\boldsymbol{\varepsilon})\right)^2, \tag{2.23}$$

where μ is the shear modulus, λ is Lamé's first parameter and are related to the bulk modulus κ as $\kappa = \lambda + \frac{2}{3}\mu$. The linear elastic Cauchy stress tensor is derived from the energy functional by taking the first derivative with respect to the small strain tensor as:

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}) = \frac{\partial\psi(\boldsymbol{\varepsilon})}{\partial\boldsymbol{\varepsilon}} = 2\mu\boldsymbol{\varepsilon} + \lambda\mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I}. \tag{2.24}$$

[4]See Appendix A for further details on thermo-elastic constitutive models.

In the small strain regime, the Cauchy and first Piola-Kirchhoff stress tensors are identical and can be expressed in terms of the deformation gradient tensor by making use of the definition of spatial strain in terms of the deformation gradient as:

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\boldsymbol{F} + \boldsymbol{F}^T - 2\boldsymbol{I}\right), \tag{2.25}$$

by simple substitution into Equation (2.24) *and manipulation, noting the relation between* κ, μ *and* λ *the definition of the Cauchy and first Piola-Kirchhoff stress tensors can be obtained as:*

$$\boldsymbol{P}(\boldsymbol{F}) = \mu\left[\boldsymbol{F} + \boldsymbol{F}^T - \frac{2}{3}\mathrm{tr}(\boldsymbol{F})\boldsymbol{I}\right] + \kappa\left(\mathrm{tr}(\boldsymbol{F}) - 3\right)\boldsymbol{I}. \tag{2.26}$$

The two hyperelastic models in Section 2.5 degenerate to this simple model for the small strain regime. Since this research is focused on large strain contact dynamics, the small strain regime will only be considered for benchmark scenarios through this degenerative property of hyperelastic constitutive models.

2.5.3 Rate-Independent von-Mises Plasticity

A more complex constitutive model is that of rate-independent von-Mises plasticity with isotropic hardening, which is the simplest plasticity model and is used to define materials that deform plastically where the stress is independent of the strain rate. This constitutive model will only briefly be described here for application in numerical examples, for further detail refer to [33]. In order to utilise this constitutive model the multiplicative decomposition of the deformation gradient is required and defined as:

$$\boldsymbol{F} = \boldsymbol{F}_e\boldsymbol{F}_p. \tag{2.27}$$

where $\boldsymbol{F}_e$ and $\boldsymbol{F}_p$ are the deformation gradient tensors associated with the elastic and plastic contributions respectively. This leads to the definitions of the elastic left Cauchy-Green strain tensor $\boldsymbol{b}_e = \boldsymbol{F}\boldsymbol{C}_p^{-1}\boldsymbol{F}^T$ with right plastic Cauchy-Green strain tensor $\boldsymbol{C}_p = \boldsymbol{F}_p^T\boldsymbol{F}_p$. The basic algorithm for solving this type of constitutive model is given in Algorithm 2.1, where the formulation of the stain energy functional is expressed in terms of the principle stretches and Jacobian as:

$$W(\lambda_{e,1}, \lambda_{e,2}, \lambda_{e,3}, J) = \hat{W}(\lambda_{e,1}, \lambda_{e,2}, \lambda_{e,3}) + \mathrm{U}(J), \tag{2.28}$$

where:

$$\hat{W} = \mu \left[\sum_{i=1}^{3} \left(\ln(\lambda_{e,i}) \right)^2 \right] + \frac{1}{3}\mu \left(\ln J \right)^2 - \frac{2}{3}\mu \ln J \left(\sum_{i=1}^{3} \left(\ln(\lambda_{e,i}) \right) \right), \tag{2.29}$$

and:

$$\mathrm{U}(J) = \frac{1}{2}\kappa \left(\ln J \right)^2. \tag{2.30}$$

Algorithm 2.1: Rate-Independent von-Mises Plasticity

Input : $\boldsymbol{F}^{n+1}$, $[\boldsymbol{C}_{\boldsymbol{p}}^{-1}]^{\boldsymbol{n}}$, ε_p^n

Output: $\boldsymbol{P}^{n+1}$

Compute Jacobian: $J^{n+1} = \det\left(\boldsymbol{F}^{n+1}\right)$;

Compute Pressure: $p^{n+1} = \kappa\left(\frac{\ln J^{n+1}}{J^{n+1}}\right)$;

Compute trial elastic left cauchy strain tensor: $\boldsymbol{b}_{e,trial}^{n+1} = \boldsymbol{F}^{n+1}\left[\boldsymbol{C}_p^{-1}\right]^n [\boldsymbol{F}]^{n+1}$;

Spectral decomposition of $\boldsymbol{b}_{e,trial}^{n+1}$:

$\lambda_{e,i}^{trial}, \boldsymbol{n}_i^{trial} \leftarrow \boldsymbol{b}_{e,trial}^{n+1} = \sum_{i=1}^{3}\left(\lambda_{e,i}^{trial}\right)^2\left(\boldsymbol{n}_i^{trial} \otimes \boldsymbol{n}_i^{trial}\right)$;

Obtain trial deviatoric Kirchhoff stress tensor:

$\hat{\boldsymbol{\tau}}^{trial} = \sum_{i=1}^{3}\hat{\boldsymbol{\tau}}_{ii}^{trial}\left(\boldsymbol{n}_i^{trial} \otimes \boldsymbol{n}_i^{trial}\right)$; $\quad \hat{\tau}_{ii}^{trial} = 2\mu\ln\left(\lambda_e^{trial,i}\right) - \frac{2}{3}\mu\ln\left(\lambda_e^{trial,i}\right)$;

Obtain yield criterion: $f\left(\hat{\boldsymbol{\tau}}^{trial}, \varepsilon_p^n\right) = \sqrt{\frac{3}{2}\left(\hat{\boldsymbol{\tau}}^{trial} : \hat{\boldsymbol{\tau}}^{trial}\right)} - \left(\tau_y^0 + H\varepsilon_p^n\right)$;

Compute Return-Mapping:

if $f\left(\hat{\boldsymbol{\tau}}^{trial}, \varepsilon_p^n\right) > 0$ **then**

  Direction Vector: $\boldsymbol{v}_i^{n+1} = \frac{\hat{\tau}_{ii}^{trial}}{\sqrt{\frac{2}{3}\left(\hat{\boldsymbol{\tau}}^{trial} : \hat{\boldsymbol{\tau}}^{trial}\right)}}$;

  Plastic Multiplier: $\Delta\gamma = \frac{f\left(\hat{\boldsymbol{\tau}}^{trial}, \varepsilon_p^n\right)}{3\mu + H}$;

else

  $\boldsymbol{v}_i^{n+1} = \Delta\gamma = 0$;

end

Compute elastic stretch: $\lambda_{e,i}^{n+1} = \exp\left(\ln(\lambda_{e,i}^{trial} - \Delta\gamma\boldsymbol{v}_i^{n+1})\right)$;

Assign spatial normals: $\boldsymbol{n}_i^{n+1} = \boldsymbol{n}_i^{trial}$;

Compute Kirchhoff stress tensor: $\boldsymbol{\tau}^{n+1} = \sum_{i=1}^{3}\tau_{ii}\left(\boldsymbol{n}_i^{n+1} \otimes \boldsymbol{n}_i^{n+1}\right)$

$\tau_{ii} = \hat{\tau}_{ii} + J^{n+1}p^{n+1}$; $\quad \hat{\tau}_{ii} = \left(1 - \frac{2\mu\Delta\gamma}{\sqrt{\frac{2}{3}\left(\boldsymbol{\tau}^{trial\prime} : \boldsymbol{\tau}^{trial\prime}\right)}}\right)\hat{\tau}_{ii}^{trial}$;

Compute first Piola-Kirchhoff stress tensor: $\boldsymbol{P}^{n+1} = \boldsymbol{\tau}^{n+1}\left[\boldsymbol{F}^{-T}\right]^{n+1}$;

Update elastic left Cauchy strain tensor: $\boldsymbol{b}_e^{n+1} = \sum_{i=1}^{3}\left(\lambda_{e,i}^{n+1}\right)^2\left(\boldsymbol{n}_i^{n+1} \otimes \boldsymbol{n}_i^{n+1}\right)$;

Update plastic Cauchy Green tensor: $\left[\boldsymbol{C}_p^{-1}\right]^{n+1} = \left[\boldsymbol{F}^{-1}\right]^{n+1}\boldsymbol{b}_e^{n+1}\left[\boldsymbol{F}^{-T}\right]^{n+1}$;

Update plastic strain: $\varepsilon_p^{n+1} = \varepsilon_p^n + \Delta\gamma$;

2.6 Non-Smooth Problems

As mentioned previously, the first order framework for solid dynamics discussed in Section 2.4 is only valid for problems which have a smooth solution. For the framework to be applied to non-smooth problems such as contact, the conservation equations must satisfy the jump conditions of the flux variables. In this section, the jump conditions will be derived and applied to the first order conservation framework for large strain solid dynamics, based on the derivation by Bonet *et al.* [21] and included in detail here for completeness as required for the application to contact mechanics. This is followed by the derivation of two types of material wave speeds for a simple constitutive model followed by the Hugoniot shock model.

2.6.1 Derivation of General Jump Condition in Lagrangian Dynamics

To derive the general jump condition in Lagrangian dynamics, first consider the possibility of a discontinuity in the conserved variables, $\boldsymbol{\mathcal{U}}$, and flux variables, $\boldsymbol{\mathcal{F}}$, across a moving surface, $\Gamma(t)$, travelling at velocity U in the direction of $\boldsymbol{N}$ which is perpendicular to the surface, as depicted in Figure 2.2,

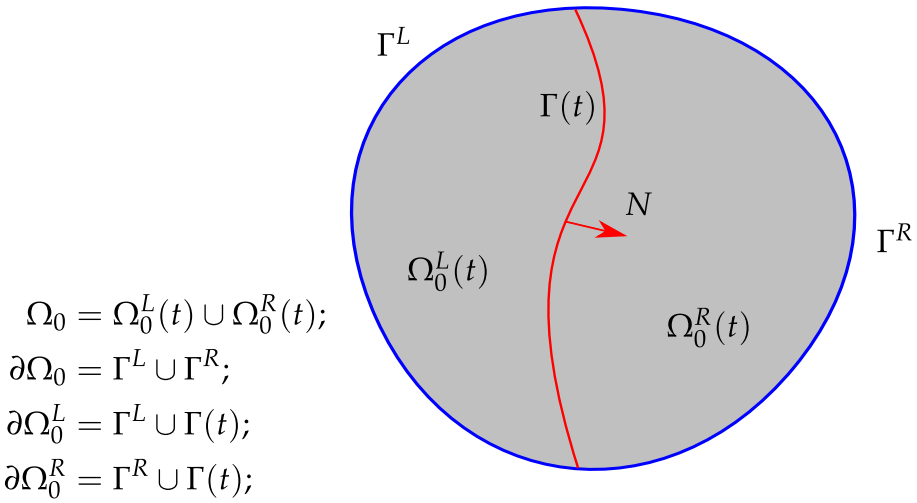


Figure 2.2: Discontinuity across moving surface, $\Gamma(t)$

where Ω_0 is the total volume and is equal to the union of the time dependant volumes to the left and to the right of the discontinuity, defined as $\Omega_0^L(t)$ and $\Omega_0^R(t)$. The boundary of the domain $\partial\Omega_0$, is defined as the union of boundary of the left and right volumes as Γ^L and Γ^R respectively. The boundary of the left volume Ω_0^L, is defined as the union of the left boundary Γ^L, and the time-dependant discontinuous surface $\Gamma(t)$. Similarly, the boundary of the right volume Ω_0^R, is defined as the union of the right boundary Γ^R, and the discontinuous surface $\Gamma(t)$. First, by considering the global balance principle

defined, for this scenario, as:

$$\frac{d}{dt}\int_{\Omega_0} \boldsymbol{\mathcal{U}}\, dV + \int_{\partial\Omega_0} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\, dA = \int_{\Omega_0} \boldsymbol{\mathcal{S}}\, dV; \quad \text{with} \qquad \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} = \sum_{I=1}^{3} \boldsymbol{\mathcal{F}}_I N_I. \tag{2.31}$$

The time rate of unknown variable can be expanded by considering the relation to the time-dependant left and right volumes by using the Reynolds transport Theorem [12] as:

$$\begin{aligned}
\frac{d}{dt}\int_{\Omega_0} \boldsymbol{\mathcal{U}}\, dV &= \frac{d}{dt}\int_{\Omega_0^L(t)} \boldsymbol{\mathcal{U}}\, dV + \frac{d}{dt}\int_{\Omega_0^R(t)} \boldsymbol{\mathcal{U}}\, dV; \\
&= \int_{\Omega_0^L} \frac{\partial \boldsymbol{\mathcal{U}}}{\partial t}\, dV + \int_{\Gamma(t)} U \boldsymbol{\mathcal{U}}^L\, dA + \int_{\Omega_0^R} \frac{\partial \boldsymbol{\mathcal{U}}}{\partial t}\, dV - \int_{\Gamma(t)} U \boldsymbol{\mathcal{U}}^R\, dA; \\
&= \int_{\Omega_0} \frac{\partial \boldsymbol{\mathcal{U}}}{\partial t}\, dV - \int_{\Gamma(t)} U \llbracket \boldsymbol{\mathcal{U}} \rrbracket\, dA.
\end{aligned} \tag{2.32}$$

This results in recovering the time derivative of the conserved variable as the first term and then an additional second term consisting of the jump in the conserved variable $\llbracket \boldsymbol{\mathcal{U}} \rrbracket = \left(\boldsymbol{\mathcal{U}}^R - \boldsymbol{\mathcal{U}}^L \right)$ across the discontinuous surface. At this point, it is important to note that the negative term in the equation above is due to the opposite directions of the outward normals of the left and right volumes, i.e. $\boldsymbol{N}^L$ is $+\boldsymbol{N}$ while $\boldsymbol{N}^R$ is $-\boldsymbol{N}$. Next the jump in the surface fluxes can be derived by considering the surface flux term and expanding based on the boundary relations mentioned previously and in Figure 2.2, as follows:

$$\begin{aligned}
\int_{\partial\Omega_0} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} dA &= \int_{\partial\Omega_0^L} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\, dA - \int_{\Gamma(t)} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}^L dA + \int_{\partial\Omega_0^R} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\, dA + \int_{\Gamma(t)} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}^R dA; \\
&= \int_{\partial\Omega_0^L} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\, dA + \int_{\partial\Omega_0^R} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\, dA + \int_{\Gamma(t)} \llbracket \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} \rrbracket\, dA; \\
&= \int_{\Omega_0} \frac{\partial \boldsymbol{\mathcal{F}}_I}{\partial X_I}\, dV + \int_{\Gamma(t)} \llbracket \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} \rrbracket\, dA.
\end{aligned} \tag{2.33}$$

Similarly to the conserved variable term, this results in recovering the material derivative of the flux term and then an additional term that represents the jump in the flux variable $\llbracket \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} \rrbracket = \left(\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}^R - \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}^L \right)$ across the discontinuous surface. Equation (2.32) and (2.33) are combined to give:

$$\int_{\Omega_0} \left(\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \frac{\partial \boldsymbol{\mathcal{F}}_I}{\partial X_I} - \boldsymbol{\mathcal{S}} \right) dV = \int_{\Gamma(t)} \left(U \llbracket \boldsymbol{\mathcal{U}} \rrbracket - \llbracket \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} \rrbracket \right) dA.$$

Thus recovering the standard pointwise conservation equation as introduced previ-

ously in Section 2.4 which is smooth and differentiable:

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \sum_{I=1}^{3} \frac{\partial \boldsymbol{\mathcal{F}}_I}{\partial X_I} = \boldsymbol{\mathcal{S}} \quad \text{in} \quad \Omega_0,$$

together with the jump condition:

$$U[\![\boldsymbol{\mathcal{U}}]\!] = [\![\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}]\!] \quad \text{on} \quad \Gamma(t). \tag{2.34}$$

Equation (2.34) can be particularised for the first order framework, assuming identical material across the discontinuous surface, as follows:

$$U\rho_0[\![\boldsymbol{v}]\!] = -[\![\boldsymbol{P}]\!]\boldsymbol{N}; \tag{2.35a}$$

$$U[\![\boldsymbol{F}]\!] = -[\![\boldsymbol{v}]\!] \otimes \boldsymbol{N}; \tag{2.35b}$$

$$U[\![\boldsymbol{H}]\!] = -\boldsymbol{F}^{Ave} \boldsymbol{\times} ([\![\boldsymbol{v}]\!] \otimes \boldsymbol{N}); \tag{2.35c}$$

$$U[\![J]\!] = -\boldsymbol{H}^{Ave} : ([\![\boldsymbol{v}]\!] \otimes \boldsymbol{N}). \tag{2.35d}$$

where $\boldsymbol{F}^{Ave}$ and $\boldsymbol{H}^{Ave}$ are the average of the deformation gradient tensor and area map across the discontinuous surface. From these jump conditions it can be observed that the jump in the flux variables depends on only the jump in the first Piola-Kirchhoff stress tensor $[\![\boldsymbol{P}]\!]$ and velocity $[\![\boldsymbol{v}]\!]$. With these jump conditions now derived, expressions for wave speed U can also be derived. To do so, the jump conditions for the velocity and deformation gradient will be examined from Equations (2.35a) and (2.35b) only, as Equations (2.35c) and (2.35d) are simply an extension of Equation (2.35b).

2.6.2 Derivation of Simple Material Wave Speeds

For the scenario where the wave speed U is assumed to be equal to the speed of sound of the material, the acoustic wave speeds c for a specific constitutive model are typically derived from an eigenstructure analysis[5], resulting in two wave speeds, corresponding to the pressure wave speed c_p and the shear wave speed c_s respectively. For a simple constitutive model such as nearly incompressible neo-Hookean (see Section 2.5.2), a simpler approach can be utilised and is demonstrated here following [21]. Using this method, first the kinematic jump (Equation (2.35b)) can be post-multiplied by the discontinuity normal $\boldsymbol{N}$, and combined with the kinetic jump (Equation (2.35a)) to give:

$$c[\![\boldsymbol{F}]\!]\boldsymbol{N} = -[\![\boldsymbol{v}]\!] = \frac{1}{c\rho_0}[\![\boldsymbol{P}]\!]\boldsymbol{N}.$$

[5]See reference [106] for an example eigenstructure analysis in the context of the first order framework.

Rearranging this equation gives:

$$c^2\rho_0[\![\boldsymbol{F}]\!]\boldsymbol{N} = [\![\boldsymbol{P}]\!]\boldsymbol{N}. \tag{2.36}$$

To solve for the wave speeds, first a shear wave c_s (in the tangential spatial direction of $\boldsymbol{F}\boldsymbol{T}_\alpha$, where $\boldsymbol{T}_\alpha$ is the tangential vector with $\alpha = 1, 2$) can be considered making use of the definitions of the right Cauchy-Green tensor $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$ and second Piola-Kirchhoff Stress Tensor $\boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{P}$ to give:

$$\begin{aligned}
c_s^2\rho_0[\![\boldsymbol{F}]\!]\boldsymbol{N}\cdot\boldsymbol{F}\boldsymbol{T}_\alpha &= [\![\boldsymbol{P}]\!]\boldsymbol{N}\cdot\boldsymbol{F}\boldsymbol{T}_\alpha;\\
c_s^2\rho_0[\![\boldsymbol{F}^T\boldsymbol{F}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N}) &= [\![\boldsymbol{F}^T\boldsymbol{P}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N});\\
c_s^2\rho_0[\![\boldsymbol{C}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N}) &= [\![\boldsymbol{C}\boldsymbol{S}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N});
\end{aligned}$$

which results in:

$$c_s^2\rho_0 = \frac{[\![\boldsymbol{C}\boldsymbol{S}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N})}{[\![\boldsymbol{C}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N})}. \tag{2.37}$$

Now considering a simple neo-Hookean material (see Section 2.5.2) for which the second Piola-Kirchhoff stress tensor is defined as:

$$\boldsymbol{S} = \mu(\boldsymbol{I} - \boldsymbol{C}^{-1}) + \lambda(\ln J)\boldsymbol{C}^{-1}. \tag{2.38}$$

By substituting Equation (2.38) into Equation (2.37) and through algebraic manipulation results in:

$$\begin{aligned}
c_s^2\rho_0 &= \frac{[\![\boldsymbol{C}\boldsymbol{S}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N})}{[\![\boldsymbol{C}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N})},\\
&= \frac{(\mu[\![\boldsymbol{C}]\!] + \lambda[\![(\ln J)]\!]\boldsymbol{I}) : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N})}{[\![\boldsymbol{C}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N})},\\
&= \frac{\mu[\![\boldsymbol{C}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N}) + \lambda[\![(\ln J)]\!]\boldsymbol{T}_\alpha\cdot\boldsymbol{N}}{[\![\boldsymbol{C}]\!] : (\boldsymbol{T}_\alpha\otimes\boldsymbol{N})},\\
&= \mu.
\end{aligned}$$

This results in a acoustic shear wave speed for this constitutive model that coincides with the acoustic linear elastic shear wave:

$$c_s = \sqrt{\frac{\mu}{\rho_0}}. \tag{2.39}$$

Next, for the acoustic pressure wave speed c_p, by utilise the isochoric-volumetric de-

composition of first Piola-Kirchhoff defined as:

$$\boldsymbol{P} = \hat{\boldsymbol{P}} + p\boldsymbol{H}; \tag{2.40}$$

and assuming the jump in first Piola-Kirchhoff is dominated by the jump in the pressure, which is typical of problems with strong discontinuities such as contact [6], this can be written as:

$$[\![\boldsymbol{P}]\!]\boldsymbol{N} \approx [\![p\boldsymbol{H}]\!]\boldsymbol{N}; \qquad [\![\hat{\boldsymbol{P}}]\!]\boldsymbol{N} = \boldsymbol{0}.$$

By considering the case of an acoustic pressure wave speed c_p (in the normal spatial direction $\boldsymbol{n}$) utilising the definition of Nanson's rule as $\boldsymbol{n} = \frac{\boldsymbol{H}\boldsymbol{N}}{\Lambda_A}$ where $\Lambda_A = \boldsymbol{n} \cdot \boldsymbol{H}\boldsymbol{N}$:

$$\begin{aligned} c_p^2 \rho_0 \left([\![\boldsymbol{F}]\!]\boldsymbol{N} \right) \cdot \boldsymbol{n} &= \left([\![\boldsymbol{P}]\!]\boldsymbol{N} \right) \cdot \boldsymbol{n}, \\ c_p^2 \rho_0 [\![\boldsymbol{F}]\!]\boldsymbol{N} \cdot \left(\frac{1}{\Lambda_A} \boldsymbol{H}\boldsymbol{N} \right) &= [\![p\boldsymbol{H}]\!]\boldsymbol{N} \cdot \boldsymbol{n}, \\ c_p^2 \frac{\rho_0}{\Lambda_A} [\![\boldsymbol{H}^T \boldsymbol{F}]\!] : (\boldsymbol{N} \otimes \boldsymbol{N}) &= [\![p]\!]\Lambda_A, \end{aligned}$$

and since $\boldsymbol{H}^T \boldsymbol{F} = J\boldsymbol{I}$ the equation becomes:

$$\begin{aligned} c_p^2 \frac{\rho_0}{\Lambda_A} [\![J]\!] &= [\![p]\!]\Lambda_A, \\ c_p^2 \rho_0 &= \Lambda_A^2 \frac{[\![p]\!]}{[\![J]\!]}. \end{aligned} \tag{2.41}$$

In the very simplistic case of a nearly incompressible material where $p = \kappa(J-1)$:

$$c_p = \Lambda_A \sqrt{\frac{\kappa}{\rho_0}}. \tag{2.42}$$

From this neo-Hookean expression for the pressure wave speed, the linear elastic pressure wave speed is recovered when in the small deformation regime, $\Lambda_A = 1$. As mentioned previously this derivation can only be conducted using a simple constitutive model. For more complex constitutive models the wave speeds must be derived using eigenstructure analysis [21, 103, 105, 106, 108], however either method will produce exact expressions for the acoustic wave speeds of the material. This is highly beneficial in solving discontinuous problems such as contact where accurate wave propagation of materials is of utmost importance.

[6] In contact dynamics, the shock wave speed is dominated by the pressure wave speed, therefore it is useful to assume that the first Piola-Kirchhoff stress tensor in Equation (2.40) is dominated by the volumetric components of the stresses. Such that the derivation for the wave speed can neglect the deviatoric components.

2.6.3 Shock Wave Speeds Derived From Experimental Data

To accurately model the shock wave propagation it becomes apparent from experimental measurements and observation that for most metals the speed of the pressure shock wave U_p in the normal direction of wave propagation is linearly related to the speed of sound of the material and particle velocity v_p via a constant coefficient s defined as:

$$U_p = c_p + sv_p. \tag{2.43}$$

This linear relation is formulated based on one-dimensional experiments and observations [21, 125], therefore a general multi-dimensional expression for the shock pressure wave can be further defined as:

$$U_p = c_p - s\Lambda_A[\![v_n]\!], \tag{2.44}$$

where $[\![v_n]\!]$ is the jump in normal velocity across the shock front. In order to obtain an expression for the constitutive pressure, the pressure wave speed can then be expressed in terms of the jump in the volume ratio from Equation (2.35d) to give $[\![v_n]\!] = -\frac{U_p}{\Lambda_A}[\![J]\!]$. Therefore the pressure wave speed as a function of the jump in the volume ratio can be expressed as:

$$U_p = \frac{c_p}{1 - s[\![J]\!]}. \tag{2.45}$$

The definition of the pressure behind the shock, also known as the Hugoniot state, can then be derived from Equation (2.41) when considering the case where the material wave speed c is now substituted by the shock wave speed U as:

$$\begin{aligned} \frac{[\![p]\!]}{[\![J]\!]} &= \frac{U_p^2\rho_0}{\Lambda_A^2}; \\ &= \frac{c_p^2\rho_0}{\Lambda_A^2}\left[\frac{1}{(1 - s[\![J]\!])^2}\right]. \end{aligned} \tag{2.46}$$

For the scenario where the shock wave is travelling through undisturbed material, the jump in pressure is observed to be $[\![p]\!] = 0 - p_H$, where p_H is the Hugoniot pressure, and the jump in volume ratio is observed as $[\![J]\!] = 1 - J_H$, where J_H is the Hugoniot volume ratio. Thus by substituting these observed conditions into Equation (2.46), the shock pressure can be expressed as follows, rewriting the Hugoniot volume ratio as $J_H = J$ and assuming no cross sectional area changes gives:

$$p_H(J) = c_p^2\rho_0\left[\frac{J - 1}{[1 - s(1 - J)]^2}\right]. \tag{2.47}$$

The Hugoniot pressure is traditionally derived via the Mie-Grüneisen Equation of State (EOS) [125] by assuming the pressure is significantly large thus there is no deviatoric (shear) stress and is therefore decoupled from the stress tensor. This assumption allows the Hugoniot pressure model to be coupled with any appropriate deviatoric constitutive model such as those in Section 2.5. The expression in Equation (2.47), however, is only valid when the model is in compression i.e. volume ratio is less than one $(J < 1)$, if the model is in tension $(J > 1)$ the appropriate constitutive pressure formulation is required, matching the deviatoric model.

To visibly demonstrate the importance of the Hugoniot pressure in Equation (2.47) compared to a traditional constitutive pressure formulation, such as those presented in Section 2.5, the two different pressure models are plotted for a range of volume ratios in Figure 2.3. The figure shows a normalised pressure $\left(\frac{p}{E}\right)$, for a linear elastic scenario with no shear i.e. when $E = c_p^2 \rho_0$, comparing three scenarios. First, the linear model which is equivalent to $s = 0$, second, the Hugoniot model with $s = 0.5$ and lastly the Hugoniot model with $s = 1$.

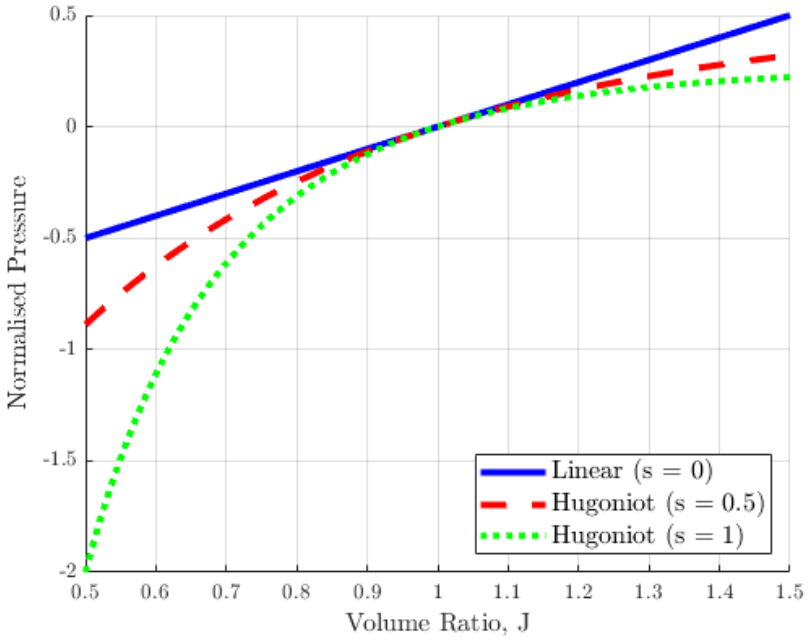


Figure 2.3: Comparison of normalised pressures $\left(\frac{p}{E}\right)$ using the linear constitutive pressure model and two Hugoniot pressure models where $s = \{0.5, 1\}$

It can be observed, when the material undergoes compression $(J < 1)$, the Hugoniot pressure becomes significantly larger dependant on the value of slope s in comparison to the linear model as the volume ratio decreases. While the material undergoes extension $(J > 1)$ and the Hugoniot pressure model becomes invalid, there is less separation

between the models with the linear model resulting in a higher pressure. When the material is in the small strain regime ($J \approx 1$) both models are identical even for different values of slope s.

For these types of scenarios the corresponding Hugoniot internal energy $\mathcal{E}_H$ will be derived here for completeness, by examining the conservation of energy from remark 2.2[7]. The total energy of an isothermal system E_T can be defined as the sum of the kinetic energy and internal potential energy as:

$$E_T = \frac{1}{2\rho_0}\boldsymbol{p}\cdot\boldsymbol{p} + \mathcal{E}\left(\boldsymbol{F},\boldsymbol{H},J\right), \tag{2.48}$$

with associated jump condition expressed as:

$$[\![E_T]\!] = \rho_0 \boldsymbol{v}^{Ave}\cdot[\![\boldsymbol{v}]\!] + [\![\mathcal{E}]\!]. \tag{2.49}$$

In order to obtain one complete expression for the jump in total energy, Equation (2.49) is then substituted into Equation (A.3) and decomposed[8] for an isothermal case, as follows:

$$\begin{aligned}\rho_0 \boldsymbol{v}^{Ave}\cdot[\![\boldsymbol{v}]\!] + [\![\mathcal{E}]\!] &= -\frac{1}{U}[\![\boldsymbol{P}^T\boldsymbol{v}]\!]\cdot\boldsymbol{N};\\ &= -\frac{1}{U}[\![\boldsymbol{P}\boldsymbol{N}]\!]\cdot\boldsymbol{v}^{Ave} - \frac{1}{U}\left(\boldsymbol{P}^{Ave}\boldsymbol{N}\right)\cdot[\![\boldsymbol{v}]\!]\end{aligned}$$

The first term on the right hand side can then be re-expressed by substitution of the linear momentum jump condition from Equation (2.35a) to give:

$$-\frac{1}{U}[\![\boldsymbol{P}\boldsymbol{N}]\!]\cdot\boldsymbol{v}^{Ave} = \rho_0[\![\boldsymbol{v}]\!]\cdot\boldsymbol{v}^{Ave}, \tag{2.50}$$

then by rearranging, the kinetic energy components cancel and an expression for the jump in internal energy can be obtained in terms of the first Piola-Kirchhoff stress tensor

[7] Traditionally the internal stress is obtained by taking the derivative of the strain energy functional, refer to Section 2.5, and vice-versa the internal energy can be recovered through integration. In this scenario however, the large shock wave (Jump) is not differentiable so the internal energy must be derived by an alternate method.

[8] Making use of the multiplicative property of jump conditions such that for the jump of two variables $[\![AB]\!]$ they can be decomposed as $[\![AB]\!] = A^{Ave}[\![B]\!] + [\![A]\!]B^{Ave}$ with $[\cdot]^{Ave} = \frac{1}{2}\left([\cdot]^L + [\cdot]^R\right)$.

and jump in deformation gradient as:

$$\begin{aligned}\llbracket \mathcal{E} \rrbracket &= -\rho_0 \boldsymbol{v}^{Ave} \cdot \llbracket \boldsymbol{v} \rrbracket + \rho_0 \boldsymbol{v}^{Ave} \cdot \llbracket \boldsymbol{v} \rrbracket - \frac{1}{U}\left(\boldsymbol{P}^{Ave}\boldsymbol{N}\right) \cdot \llbracket \boldsymbol{v} \rrbracket; \\ &= -\frac{1}{U}\boldsymbol{P}^{Ave} : \left(\llbracket \boldsymbol{v} \rrbracket \otimes \boldsymbol{N}\right); \\ &= \boldsymbol{P}^{Ave} : \llbracket \boldsymbol{F} \rrbracket.\end{aligned}$$

Now by recalling the isochoric-volumetric decomposition of the first Piola-Kirchhoff stress tensor from Equation (2.40), for the Hugoniot state as discussed previously, the deviatoric component can be neglected such that the jump in internal energy $\mathcal{E}$ can be approximated to only the volumetric component as:

$$\begin{aligned}\llbracket \mathcal{E} \rrbracket &= \hat{\boldsymbol{P}}^{Ave} : \llbracket \boldsymbol{F} \rrbracket + \left(p\boldsymbol{H}\right)^{Ave} : \llbracket \boldsymbol{F} \rrbracket; \\ &\approx \left(p\boldsymbol{H}\right)^{Ave} : \llbracket \boldsymbol{F} \rrbracket.\end{aligned}$$

This volumetric approximation can be simplified by noting that $\llbracket \boldsymbol{F} \rrbracket = \llbracket \boldsymbol{F}\boldsymbol{N} \rrbracket \otimes \boldsymbol{N}$ and rearranged by making use of the geometric jump conditions from Equation (2.35b) and (2.35d) to give:

$$\begin{aligned}\llbracket \mathcal{E} \rrbracket &= \left(p\boldsymbol{H}\right)^{Ave} : \llbracket \boldsymbol{F} \rrbracket; \\ &= \left(p\boldsymbol{H}\right)^{Ave} : \llbracket \boldsymbol{F}\boldsymbol{N} \rrbracket \otimes \boldsymbol{N}; \\ &= \left(p\boldsymbol{H}\right)^{Ave} \boldsymbol{N} \cdot \llbracket \boldsymbol{F}\boldsymbol{N} \rrbracket; \\ &= \left(p\boldsymbol{H}\right)^{Ave} \boldsymbol{N} \cdot \left(-\frac{1}{U}\llbracket \boldsymbol{v} \rrbracket\right); \\ &= p^{Ave}\left(-\boldsymbol{H}^{Ave}\boldsymbol{N} \cdot \frac{1}{U}\llbracket \boldsymbol{v} \rrbracket\right); \\ &= p^{Ave}\llbracket J \rrbracket. \end{aligned} \tag{2.51}$$

Provided the shock is propagating through an undisturbed material the jump conditions can be defined as:

$$\llbracket \mathcal{E} \rrbracket = 0 - \mathcal{E}_H; \quad \llbracket J \rrbracket = 1 - J; \quad p^{Ave} = \frac{1}{2}\left(0 + p_H\right);$$

where $\mathcal{E}_H$ is the Hugoniot internal energy and p_H is the Hugoniot pressure. By substituting these definitions into Equation (2.51) gives:

$$\begin{aligned}0 - \mathcal{E}_H &= \frac{1}{2}p_H\left(1 - J\right) \\ \mathcal{E}_H &= \frac{1}{2}p_H\left(J - 1\right).\end{aligned}$$

Finally, the Hugoniot pressure from Equation (2.47) is substituted to obtain the final expression for Hugoniot energy as:

$$\begin{aligned}\mathcal{E}_H(J) &= \frac{1}{2}(J-1)\left[\rho_0 c_p^2 \frac{(J-1)}{[1-s(1-J)]^2}\right] \\ &= \frac{1}{2}c_p^2\rho_0\left[\frac{(J-1)^2}{[1-s(1-J)]^2}\right].\end{aligned} \tag{2.52}$$

Similar to the Hugoniot pressure, this relation is only valid when the shock is significantly large and the material is undergoing compression.

Chapter 3

Continuum Contact Equations

3.1 Preliminaries

The first order framework for non-smooth large strain solid dynamics described in Chapter 2 will now be extended to describe dynamic multi-body contact at a continuum level. This chapter will define several important relations for contact dynamics, these include the impenetrability, traction balance and unitary contact conditions in Section 3.2. To solve these contact conditions the proposed formulation is applied to contact scenarios through the introduction of two different Riemann solvers, acoustic and non-linear, to exactly express the flux variables at the contact interface for various contact scenarios in Section 3.3. These conditions are then applied to a simple one-dimensional linear elastic contact example to derive the exact solution in Section 3.4. Lastly, in Section 3.5, the Hamiltonian energy is introduced for the contact-impact scenarios. In this chapter the contact conditions are defined in relation to two bodies, labelled as Body A (Master) and Body B (Slave) where all variables relating to an individual body will be expressed as superscript A or B. Several equations can be applied to both bodies therefore will be expressed as superscript f where $f = \{A, B\}$.

3.2 Continuum Equations

The motion and deformation of multiple bodies can be expressed with the definitions used in non-linear continuum mechanics, as discussed in Chapter 2 for a single body. These definitions for deformation gradient tensor $\boldsymbol{F}$, co-factor $\boldsymbol{H}$, Jacobian J and velocity $\boldsymbol{v}$ can simply be applied to multiple bodies with the addition of superscript $f = \{A, B\}$ to define each body individually. Therefore contact between two bodies can be described as follows, first consider two bodies as shown in Figure 3.1.

Initially, at $t = 0$, bodies A and B are not in contact with a gap between the bodies $\boldsymbol{\delta}$, where Ω_0^f refers to the initial volume, Γ_0^f refers to the initial boundary and $\boldsymbol{N}^f$ refers to the outward unit normal of each body respectively. Both bodies are then subjected to an individual mapping $\boldsymbol{\phi}^f(\boldsymbol{X}^f, t)$ resulting in the bodies moving or deforming to a spatial configuration where the boundaries of the bodies come into contact ($\boldsymbol{\delta} = 0$), producing a contact interface Γ^C and contact normal $\boldsymbol{n} = \boldsymbol{n}^A$ and $\boldsymbol{n}^A = -\boldsymbol{n}^B$. It is important to note that the contact interface refers to both contact surfaces associated with each body. At the contact interface two key conditions, namely the impenetrability and traction balance conditions, must be fulfilled to ensure continuity across both contact surfaces [12], no additional interface is required to enforce the contact conditions.

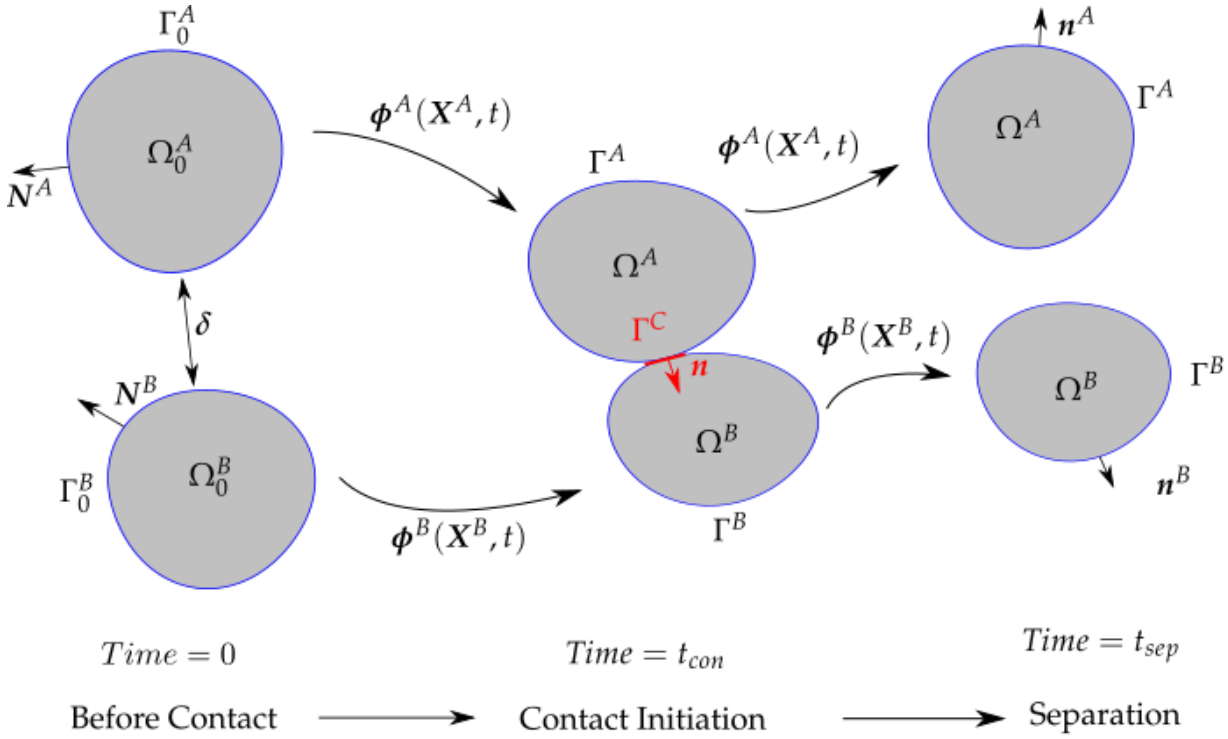


Figure 3.1: Kinematics of Two Contacting Bodies

3.2.1 Impenetrability Condition

The impenetrability condition for two bodies is defined as the intersection of the two bodies must be equal to zero. This is written as:

$$\Omega^A \cap \Omega^B = 0. \tag{3.1}$$

For problems where the displacement of the contact interface is relatively small, such that for any given position on the contact interface Γ^C, there is a corresponding position on Body A $\boldsymbol{x}^A$ and a matching position on Body B $\boldsymbol{x}^B$, for the entire duration of contact.

This condition can be expressed simply in terms of the displacement of the two bodies, by defining the normal gap or gap function as:

$$\delta_n = \left(\boldsymbol{x}^B - \boldsymbol{x}^A\right) \cdot \boldsymbol{n} \geq 0, \tag{3.2}$$

where $\boldsymbol{x}^f$ is defined as the sum of the initial position $\boldsymbol{X}^f$ and displacement $\boldsymbol{u}^f$ ($\boldsymbol{x}^f = \boldsymbol{X}^f + \boldsymbol{u}^f$), and $\boldsymbol{n}$ is the spatial outward unit normal of the master body. This condition enforces that when the normal gap is greater than zero ($\delta_n > 0$) the bodies are not in contact, however when the normal gap is equal to zero $\delta_n = 0$ the bodies are in contact. For these types of scenarios, the initial tangential movement is also assumed small and considered as contact-stick, such that the tangential gap $\boldsymbol{\delta_t}$ is defined as:

$$\boldsymbol{\delta}_t = \boldsymbol{\delta} - \delta_n \boldsymbol{n} = (\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n})\, \boldsymbol{\delta} = \boldsymbol{0} \quad \text{on} \quad \Gamma^C. \tag{3.3}$$

For contact interfaces with large displacements, such that for any given position on the contact interface Γ^C, the corresponding position on Body A and B vary with time t. This is typical for a contact interface with significant tangential movement such as frictionless contact. Therefore, the impenetrability condition is highly non-linear, as a result it cannot be expressed as an algebraic or differential equation [12] therefore it must be expressed in rate form as:

$$\gamma_n = \left(\boldsymbol{v}^A - \boldsymbol{v}^B\right) \cdot \boldsymbol{n} \leq 0 \quad \text{on} \quad \Gamma^C, \tag{3.4}$$

where $\boldsymbol{v}^f$ is the velocity across the contact interface. It can be noted that Equation (3.4) restricts the interpenetration rate for two points on the boundary to be either zero, which means the bodies are in contact, or negative, which means the bodies are moving apart. Equivalently the tangential components can be written in a similar form as:

$$\boldsymbol{\gamma_t} = \boldsymbol{v}_t^A - \boldsymbol{v}_t^B \leq 0 \quad \text{on} \quad \Gamma^C, \tag{3.5}$$

where $\boldsymbol{v}_t^f$ is the tangential decomposition of $\boldsymbol{v}^f$ defined as:

$$\boldsymbol{v}_t^f = \boldsymbol{v}^f - v_n^f \boldsymbol{n} = (\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n})\, \boldsymbol{v}^f. \tag{3.6}$$

In numerical schemes small interpenetration is generally allowed between contacting bodies, such as the penalty method, therefore Equation (3.1) above is not observed exactly for instances in time.

3.2.2 Traction Balance Condition

The second condition that must be fulfilled is that of the momentum balance across the contact interface which must be fulfilled by the tractions. For the tractions $\boldsymbol{t}^f$ at the boundary to fulfil this condition the sum of tractions must be equal to zero, since the boundary has no mass, and can be expressed as:

$$\boldsymbol{t}^A + \boldsymbol{t}^B = \boldsymbol{0}. \tag{3.7}$$

The tractions can be defined in terms of the material configuration as:

$$\boldsymbol{t}^f = \boldsymbol{P}^f \boldsymbol{N}^f, \tag{3.8}$$

where $\boldsymbol{P}^f$ is the first Piola-Kirchhoff stress tensor depending on the constitutive model as defined in Section 2.5. Therefore the normal tractions can be defined as:

$$t_n^f = \boldsymbol{t}^f \cdot \boldsymbol{n}; \tag{3.9}$$

where $\boldsymbol{n}$ is the master outward normal vector. As a result the momentum balance of the normal traction can be expressed as:

$$t_n^A + t_n^B = 0. \tag{3.10}$$

By not considering adhesion between contact surfaces in the normal direction, the normal tractions cannot be tensile therefore the normal tractions must be compressive or zero and can be expressed as:

$$t_n^C \equiv t_n^A(\boldsymbol{x}, t) = -t_n^B(\boldsymbol{x}, t) \leq 0; \tag{3.11}$$

since the normal of one of the bodies is selected to define the normal traction, the sign will depend on the choice of normal. Similarly the sum of tangential tractions must be equal to zero:

$$\boldsymbol{t}_t^A + \boldsymbol{t}_t^B = \boldsymbol{0}. \tag{3.12}$$

where t_t^f is the tangential decomposition of t^f defined as:

$$\boldsymbol{t}_t^f = \boldsymbol{t}^f - t_n^f \boldsymbol{n} = (\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n})\, \boldsymbol{t}^f \tag{3.13}$$

In the particular case of a frictionless model of contact the tangential tractions vanish as:

$$\boldsymbol{t}_t^A = \boldsymbol{t}_t^B = 0. \tag{3.14}$$

3.2.3 Unitary Contact Condition

The impenetrability condition and the traction balance condition can be combined into one condition known as the unitary contact condition and can be expressed in terms of the gap function δ_n for small displacement interfaces as:

$$t_n^C \delta_n = 0. \tag{3.15}$$

This condition must hold on the contact surface such that if the gap is equal to zero ($\delta_n = 0$) the bodies are in contact and the normal traction can follow Equation (3.11) ($t_n \leq 0$), while if the normal gap is greater than zero (δ_n), the normal tractions must be zero ($t_n = 0$) as the bodies are not in contact. In this research the gap function will be used to determine contact as only small displacement contact interfaces are considered for multi-body contact.

Remark 3.1. *Alternatively, for large displacement contact interfaces, the unitary contact condition can be expressed in a similar manner terms of the gap rate γ_n as:*

$$t_n^C \gamma_n = 0. \tag{3.16}$$

Again, this condition must hold on the contact surface such that if the gap rate is equal to zero ($\gamma_n = 0$) the bodies are in contact and the normal traction can follow Equation (3.11) *($t_n \leq 0$), while if the normal gap rate is less than zero (γ_n), the normal tractions must be zero ($t_n = 0$) as the bodies are not in contact.*

3.2.4 Contact Algorithm with Extension to Frictional Contact

To correctly enforce the contact conditions described above, the impenetrability, traction balance and unitary contact condition combine to form a set of equations which can be considered a particular case of the Karush-Kuhn-Tucker conditions from optimisation theory [15]. These contact conditions form a contact algorithm where the contact stage is determined dependant on these conditions. This contact algorithm is outlined in Algorithm 3.1 for extension to frictional contact scenarios where k is the friction coefficient. In this algorithm, the contact interface conditions and kinetic constraint are first predicted for contact-stick regime where there is no relative sliding. If the kinetic constraint is in compression then the interface is determined as in contact, otherwise the bodies are not in contact and separation will occur for bodies previously in contact. If the interface is in contact then the slip criterion f_s is checked which de-

termines if the tangential force is larger than opposing frictional force, in other words whether the contact interface is in the contact-stick or contact-slip regimes[1]. If the tangential force is smaller than the resisting frictional force then interface is in contact-stick regime, while if tangential force is larger than the resisting frictional force then interface is in contact-slip regime. With this algorithm now defined a method for expressing the contact interface conditions $\{\boldsymbol{t}^C, \boldsymbol{v}^C\}$ for each contact stage is required.

Algorithm 3.1: Classical Karush-Kuhn-Tucker Contact algorithm

if $\delta_n = 0$ **then**
  Obtain trial contact-stick traction: $\boldsymbol{t}^{C,\text{trial}} = \boldsymbol{t}^C$
  Determine the normal contact traction: $t_n^{C,\text{trial}} = \boldsymbol{n} \cdot \boldsymbol{t}^{C,\text{trial}}$
  if $t_n^{C,trial} < 0$ **then**
    Check slip criterion: $f_s = \|\boldsymbol{t}_t^{C,\text{trial}}\| - k\langle -t_n^C \rangle$
    if $f_s \leq 0$ **then**
      Contact-stick mode: $\boldsymbol{t}^C$ and $\boldsymbol{v}^C$
    else
      Contact-slip mode: $\boldsymbol{t}^C$ and $\boldsymbol{v}^C$
    end
  else
    Separation mode: $\boldsymbol{v}^C$ and $\boldsymbol{t}^C = \boldsymbol{0}$
  end
else
  Not in contact: $\boldsymbol{v}^C$ and $\boldsymbol{t}^C = \boldsymbol{0}$
end

3.3 Contact-Impact Interface Conditions

To satisfy the conditions discussed in the previous section, traditional methods utilise numerical approaches, such as the penalty or Lagrange multiplier methods to enforce these conditions, however, due to the limitations of the underlying computational method, these are not always strongly enforced. An advantage of the first order framework formulation, and a novelty for this research, is that the contact conditions can be strongly enforced by deriving exact expressions at the continuum level for the interface variables of velocity $\boldsymbol{v}^C$ and traction $\boldsymbol{t}^C$ from the jump conditions in Section 2.6.1, by using the approach known as a Riemann Solver [126]. In this section various contact conditions will be derived for two types of Riemann solvers, first the acoustic Riemann solver followed by a non-linear (consistent) Riemann solver, starting with the contact-

[1] In the case for frictionless contact, the friction coefficient is always zero therefore there is no opposing frictional force resulting in a contact interface that is always in the contact-slip regime.

stick contact condition. It should be noted that since a Riemann solver can apply to any discontinuous interface the superscript notation for bodies A and B will now represent the bulk of the body while superscript L and R will represent the left and right hand sides of any discontinuous interface, such as a contact or shock interface.

3.3.1 Acoustic Contact Conditions

Contact-Stick conditions

In the situation where infinite friction is present to prevent relative sliding, known as contact-stick, the expression for the contact variables $\{\boldsymbol{v}^C, \boldsymbol{t}^C\}$ can be derived by considering the jump conditions presented in Equations (2.34). For an acoustic Riemann solver [126, 127], first the normal components will be derived by considering the kinetic jump condition from Equation (2.35a), where for the acoustic case the pressure wave speed U_p is equal to the speed of sound of the material c_p, expressed across two bodies as:

$$c_p^L \rho_0^L [\![\boldsymbol{v}]\!] = -[\![\boldsymbol{P}]\!]^L(-\boldsymbol{N}^L); \qquad c_p^R \rho_0^R [\![\boldsymbol{v}]\!] = -[\![\boldsymbol{P}]\!]^R(-\boldsymbol{N}^R).$$

Multiplying by a unique spatial normal $\boldsymbol{n}$, and rearranging:

$$\begin{aligned} c_p^L \rho_0^L (v_n^L - v_n^C) &= t_n^L - t_n^C; \\ c_p^R \rho_0^R (v_n^R - v_n^C) &= -(t_n^R - t_n^C); \end{aligned}$$

which yields:

$$v_n^C = \frac{c_p^L \rho_0^L v_n^L + c_p^R \rho_0^R v_n^R}{c_p^L \rho_0^L + c_p^R \rho_0^R} + \frac{t_n^R - t_n^L}{c_p^L \rho_0^L + c_p^R \rho_0^R}; \tag{3.17a}$$

$$t_n^C = \frac{c_p^L \rho_0^L c_p^R \rho_0^R}{c_p^L \rho_0^L + c_p^R \rho_0^R} \left(\frac{t_n^L}{c_p^L \rho_0^L} + \frac{t_n^R}{c_p^R \rho_0^R} \right) + \frac{c_p^L \rho_0^L c_p^R \rho_0^R}{c_p^L \rho_0^L + c_p^R \rho_0^R} \left(v_n^R - v_n^L \right); \tag{3.17b}$$

where the normal components for each side of the interface are defined as:

$$v_n^L = \boldsymbol{n} \cdot \boldsymbol{v}^L; \qquad v_n^R = \boldsymbol{n} \cdot \boldsymbol{v}^R; \qquad t_n^L = \boldsymbol{n} \cdot \left(\boldsymbol{P}^L \boldsymbol{N}^L \right); \qquad t_n^R = -\boldsymbol{n} \cdot \left(\boldsymbol{P}^R \boldsymbol{N}^R \right).$$

It can be observed in Equation (3.17a) that the normal velocity at the discontinuous surface can be expressed in terms of a ratio of the acoustic pressure wave speed c_p and material density ρ_0 multiplied by the normal velocity v_n at either side of the discontinuous surface plus the ratio of the normal traction t_n at each side. Similarly, the normal traction at the discontinuous surface in Equation (3.17b) can be expressed as a ratio of the acoustic pressure wave speed and material density. From Equation (3.17a) and

(3.17b) it can be observed that the first term can be considered as the weighted average of the flux variable across the interface while the second term is a naturally occurring stabilisation from the bulk of each respective side of the interface.

For the contact-stick mode, to prevent relative sliding, a similar derivation for the tangential expression is conducted. In this scenario the shear shock wave speed is defined as c_s for example, from Equation (2.39) for linear elasticity. With the tangential velocity and tractions defined as:

$$\boldsymbol{v}_t = \boldsymbol{v} - v_n \boldsymbol{n}; \qquad \boldsymbol{t}_t = \boldsymbol{t} - t_n \boldsymbol{n}; \tag{3.18}$$

following a similar derivation as the normal components, the expressions for the tangential components are expressed as:

$$\boldsymbol{v}_t^C = \frac{c_s^L \rho_0^L \boldsymbol{v}_t^L + c_s^R \rho_0^R \boldsymbol{v}_t^R}{c_s^L \rho_0^L + c_s^R \rho_0^R} + \frac{\boldsymbol{t}_t^R - \boldsymbol{t}_t^L}{c_s^L \rho_0^L + c_s^R \rho_0^R}; \tag{3.19a}$$

$$\boldsymbol{t}_t^C = \frac{c_s^L \rho_0^L c_s^R \rho_0^R}{c_s^L \rho_0^L + c_s^R \rho_0^R} \left(\frac{\boldsymbol{t}_t^L}{c_s^L \rho_0^L} + \frac{\boldsymbol{t}_t^R}{c_s^R \rho_0^R} \right) + \frac{c_s^L \rho_0^L c_s^R \rho_0^R}{c_s^L \rho_0^L + c_s^R \rho_0^R} \left(\boldsymbol{v}_t^R - \boldsymbol{v}_t^L \right). \tag{3.19b}$$

It can be observed that the tangential components of the contact variables in Equation (3.19a) and (3.19b) are identical to the normal contact components, now represented in terms of the tangential velocity and tractions at the left and right of the interface with the shock wave speed now the shear wave speed c_s. For contact-stick contact condition, the complete post-impact velocity and traction can be expressed as:

$$\boldsymbol{v}^C = v_n^C \boldsymbol{n} + \boldsymbol{v}_t^C; \qquad \boldsymbol{t}^C = t_n^C \boldsymbol{n} + \boldsymbol{t}_t^C, \tag{3.20}$$

where $\boldsymbol{n}$ is the unique contact normal. Equations (3.20) enforces continuity of all velocity and traction components. With the contact-stick condition defined the expressions can degenerate to different contact conditions such as contact-slip and separation.

Contact-Slip conditions

The first additional and necessary contact condition is that of contact-slip where there are no shear components at the contact interface such that $c_s^R = 0$ and $\boldsymbol{t}_t^R = \boldsymbol{0}$, this simplifies the tangential components from Equation (3.19) to:

$$\boldsymbol{v}_t^C = \boldsymbol{v}_t^L - \frac{\boldsymbol{t}_t^L}{c_s^L \rho_0^L}; \qquad \boldsymbol{t}_t^C = \boldsymbol{0}, \tag{3.21}$$

while the normal components in Equation (3.17) remain unchanged and can be combined with Equation (3.21) through Equation (3.20) when the contact interface is in the

contact-slip mode.

Separation conditions

The next necessary contact condition is separation (or traction-free boundary) condition which occurs after the bodies have been in contact and have separated such that $c_p^R = c_s^R = t_n^R = 0$ and $\boldsymbol{t}_t^R = \boldsymbol{0}$ which significantly simplifies the components of both Equations (3.17) and (3.19) to:

$$v_n^C = v_n^L - \frac{t_n^L}{c_p^L \rho_0^L}; \qquad \boldsymbol{v}_t^C = \boldsymbol{v}_t^L - \frac{\boldsymbol{t}_t^L}{c_s^L \rho_0^L}; \tag{3.22a}$$

$$\boldsymbol{t}^C = \boldsymbol{0}. \tag{3.22b}$$

These equations can be applied at the contact region through Equation (3.20) to enforce the separation condition when the bodies are no longer in contact.

Rigid-body contact conditions

The next contact condition is the particular case of contact with a rigid body. In this case the wave speeds at the rigid side of the contact interface are $c_p^R = c_s^R \approx \infty$ therefore the Equation (3.20) reduces to:

$$t_n^C = t_n^L + c_p^L \rho_0^L (v_n^R - v_n^L); \qquad \boldsymbol{t}_t^C = \boldsymbol{t}_t^L + c_s^L \rho_0^L (\boldsymbol{v}_t^R - \boldsymbol{v}_t^L); \tag{3.23a}$$

$$\boldsymbol{v}^C = \boldsymbol{v}^R. \tag{3.23b}$$

These equations can be enforced again through Equation (3.20) for any rigid body contact interface conditions.

Pressure dominated contact interface

Another interesting contact condition is the case where the jump in the traction (or the first Piola Kirchhoff stress $\boldsymbol{P}$, Equation (2.19)) is dominated by the jump in the pressure component of the stress (which in this case is related to Σ_J, refer to Section 2.5), whilst the rest of the components of the stress $\{\boldsymbol{\Sigma_F}, \boldsymbol{\Sigma_H}\}$ can be neglected. This is the case when attempting to model problems with predominant nearly incompressible behaviour. Use of Equation (2.19) in conjunction with the Nanson's rule $\boldsymbol{H}^{\text{Ave}}\boldsymbol{N} = \Lambda_A \boldsymbol{n}$ (where Λ_A is the ratio between the current area and the undeformed area, see

Section 2.6.2), enables the jump in the traction vector to be:

$$\boldsymbol{t}^R - \boldsymbol{t}^L = [\![\boldsymbol{t}]\!] = [\![\boldsymbol{P}]\!]\boldsymbol{N} = [\![\boldsymbol{\Sigma_F} + \boldsymbol{\Sigma_H} \times \boldsymbol{F} + \Sigma_J \boldsymbol{H}]\!]\boldsymbol{N}; \tag{3.24a}$$

$$\approx [\![\Sigma_J]\!] \left(\boldsymbol{H}^{\text{Ave}} \boldsymbol{N} \right); \tag{3.24b}$$

$$= [\![\Sigma_J]\!] \Lambda_A \boldsymbol{n}. \tag{3.24c}$$

Neglecting the stress components $\boldsymbol{\Sigma_F}$ and $\boldsymbol{\Sigma_H}$ implies that $[\![\boldsymbol{\Sigma_F} + \boldsymbol{\Sigma_H} \times \boldsymbol{F}]\!]\boldsymbol{N} = \boldsymbol{0}$. The jump in the traction vector in the normal direction can now be derived by multiplying Equation (3.24c) with a spatial normal vector $\boldsymbol{n}$ to yields:

$$t_n^R - t_n^L = \boldsymbol{n} \cdot [\![\boldsymbol{t}]\!] = [\![\Sigma_J]\!] \Lambda_A. \tag{3.25}$$

It is interesting to notice that the jump in the tangential component of the traction vanishes. This is easily shown below as:

$$\boldsymbol{t}_t^R - \boldsymbol{t}_t^L = [\![\boldsymbol{t}_t]\!] = [\![\boldsymbol{t}]\!] - [\![t_n]\!]\boldsymbol{n}; \tag{3.26a}$$

$$= [\![\Sigma_J]\!] \Lambda_A \boldsymbol{n} - [\![\Sigma_J]\!] \Lambda_A \boldsymbol{n} = \boldsymbol{0}, \tag{3.26b}$$

by making use of Equations (3.25), (3.24c) and the relation where $\boldsymbol{t}_t = \boldsymbol{t} - \boldsymbol{t}_n \boldsymbol{n}$ thus giving the contact conditions for the normal components as:

$$v_n^C = \frac{c_p^L \rho_0^L v_n^L + c_p^R \rho_0^R v_n^R}{c_p^L \rho_0^L + c_p^R \rho_0^R} + \frac{1}{c_p^L \rho_0^L + c_p^R \rho_0^R} \left(\Sigma_J^R - \Sigma_J^L \right) \Lambda_A; \tag{3.27a}$$

$$t_n^C = \frac{c_p^L \rho_0^L c_p^R \rho_0^R}{c_p^L \rho_0^L + c_p^R \rho_0^R} \left(\frac{t_n^L}{c_p^L \rho_0^L} + \frac{t_n^R}{c_p^R \rho_0^R} \right) + \frac{c_p^L \rho_0^L c_p^R \rho_0^R}{c_p^L \rho_0^L + c_p^R \rho_0^R} \left(v_n^R - v_n^L \right); \tag{3.27b}$$

and for the tangential components as:

$$\boldsymbol{v}_t^C = \frac{c_s^L \rho_0^L \boldsymbol{v}_t^L + c_s^R \rho_0^R \boldsymbol{v}_t^R}{c_s^L \rho_0^L + c_s^R \rho_0^R}; \tag{3.28a}$$

$$\boldsymbol{t}_t^C = \frac{c_s^L \rho_0^L c_s^R \rho_0^R}{c_s^L \rho_0^L + c_s^R \rho_0^R} \left(\frac{\boldsymbol{t}_t^L}{c_s^L \rho_0^L} + \frac{\boldsymbol{t}_t^R}{c_s^R \rho_0^R} \right) + \frac{c_s^L \rho_0^L c_s^R \rho_0^R}{c_s^L \rho_0^L + c_s^R \rho_0^R} \left(\boldsymbol{v}_t^R - \boldsymbol{v}_t^L \right); \tag{3.28b}$$

These equations can be enforced again through Equation (3.20) for any pressure dominated contact interface.

Homogeneous contact interface

When considering the exact same material properties on the left and right sides of a point of contact, the density and the shock wave speeds are identical and constant for

both sides, namely $\rho_0^L = \rho_0^R = \rho_0$ and $c_p^L = c_p^R = c_p$ and $c_s^L = c_s^R = c_s$. Enforcing these conditions in Equation (3.17) and (3.19) simplifies these equations to:

$$v_n^C = \frac{1}{2}\left(v_n^L + v_n^R\right) + \frac{1}{2\rho_0 c_p}\left(t_n^R - t_n^L\right); \qquad t_n^C = \frac{1}{2}\left(t_n^L + t_n^R\right) + \frac{\rho_0 c_p}{2}\left(v_n^R - v_n^L\right), \tag{3.29a}$$

$$\boldsymbol{v}_t^C = \frac{1}{2}\left(\boldsymbol{v}_t^L + \boldsymbol{v}_t^R\right) + \frac{1}{2\rho_0 c_s}\left(\boldsymbol{t}_t^R - \boldsymbol{t}_t^L\right); \qquad \boldsymbol{t}_t^C = \frac{1}{2}\left(\boldsymbol{t}_t^L + \boldsymbol{t}_t^R\right) + \frac{\rho_0 c_s}{2}\left(\boldsymbol{v}_t^R - \boldsymbol{v}_t^L\right). \tag{3.29b}$$

Then in the particular case of nearly incompressible materials this substitution yields:

$$v_n^C = \frac{1}{2}\left(v_n^L + v_n^R\right) + \frac{1}{2\rho_0 c_p}[\![\Sigma_J]\!]\Lambda_A; \qquad t_n^C = \frac{1}{2}\left(t_n^L + t_n^R\right) + \frac{\rho_0 c_p}{2}\left(v_n^R - v_n^L\right), \tag{3.30a}$$

$$\boldsymbol{v}_t^C = \frac{1}{2}\left(\boldsymbol{v}_t^L + \boldsymbol{v}_t^R\right); \qquad \boldsymbol{t}_t^C = \frac{1}{2}\left(\boldsymbol{t}_t^L + \boldsymbol{t}_t^R\right) + \frac{\rho_0 c_s}{2}\left(\boldsymbol{v}_t^R - \boldsymbol{v}_t^L\right). \tag{3.30b}$$

From these expressions for various contact conditions, it can be observed more clearly that the Riemann solver approach is simply the sum of the average states (unstable) plus an additional upwind stabilisation that depends on the jumps which have been used in developing stabilised methods in order to alleviate unwanted spurious hour-glassing and pressure instabilities [100, 110, 112, 114–116]. By utilising an acoustic Riemann solver to compute the contact interface variables the exact solution for linear elastic materials is obtained. This however is only an approximation for more complex materials such as large strain hyperelastic materials, therefore a more accurate Riemann solver could be derived for each material model. In this research a non-linear (consistent) Riemann solver to address the Hugoniot shock model in Section 2.6.3 will be derived in the following section.

3.3.2 Non-Linear Contact Conditions

To obtain a more accurate expression for the contact conditions in the case of the Hugoniot shock model, from Section 2.6.3, a non-linear Riemann solver is derived in this section to express the contact conditions. The Hugoniot model only applies to the normal components $\{v_n^C, t_n^C\}$ therefore are derived here followed by the normal components for the rigid-body and homogeneous contact conditions.

Normal Contact Conditions

Following a similar procedure to the derivation for the acoustic Riemann solver in Section 3.3.1, first considering the kinetic jump condition from Equation (2.35a), where for the Hugoniot shock case the pressure wave speed U_p is expressed in Equation (2.44).

The normal component across across two bodies in terms of normal tractions is expressed as:

$$U_p^L \rho_0^L [\![v_n]\!]^L = [\![t_n]\!]^L; \qquad U_p^R \rho_0^R [\![v_n^R]\!] = -[\![t_n]\!]^R. \tag{3.31}$$

Starting by focusing on the left body. The first term in Equation (3.31), by substitution of the pressure wave speed U_p from Equation (2.44), can be expressed as:

$$[\![t_n]\!]^L = \rho_0^L \left(c_p^L - s\Lambda_A^L [\![v_n]\!]^L \right) [\![v_n]\!]^L; \tag{3.32a}$$

$$= \rho_0^L c_p^L [\![v_n]\!]^L - s\rho_0^L \Lambda_A^L \left([\![v_n]\!]^L \right)^2. \tag{3.32b}$$

where the jump in a variable is defined as $[\![\cdot]\!]^L = [\cdot]^C - [\cdot]^L$. The above expression after substitution and some simple algebra becomes:

$$s\rho_0^L \Lambda_A^L \left(v_n^C \right)^2 - \left(\rho_0^L c_p^L + 2s\rho_0^L \Lambda_A^L v_n^L \right) v_n^C = t_n^L - t_n^C - \rho_0^L c_p^L v_n^L - s\rho_0^L \Lambda_A^L \left(v_n^L \right)^2. \tag{3.33}$$

Analogously, with the definition of $[\![\cdot]\!]^R = [\cdot]^R - [\cdot]^C$, Equation (3.31) for the right body can also be obtained as:

$$s\rho_0^R \Lambda_A^R \left(v_n^C \right)^2 + \left(\rho_0^R c_p^R - 2s\rho_0^R \Lambda_A^R v_n^R \right) v_n^C = t_n^R - t_n^C + \rho_0^R c_p^R v_n^R - s\rho_0^R \Lambda_A^R \left(v_n^R \right)^2. \tag{3.34}$$

Equations (3.33) and (3.34) represent a system of two equations with two unknowns, namely normal contact traction t_n^C and normal contact velocity v_n^C (expressed in terms of the left and right normal tractions and velocity before the impact). By subtracting Equations (3.33) from (3.34) this will result in expressions for the normal components of velocity arranged in the form of a quadratic equation as:

$$\begin{aligned} &\left[s(\Lambda_A^R \rho_0^R - \Lambda_A^L \rho_0^L) \right] \left(v_n^C \right)^2 \\ &+ \left[\rho_0^L c_p^L + \rho_0^R c_p^R - 2s \left(\Lambda_A^R \rho_0^R v_n^R - \Lambda_A^L \rho_0^L v_n^L \right) \right] v_n^C \\ &- \left(t_n^R - t_n^L + \rho_0^L c_p^L v_n^L + \rho_0^R c_p^R v_n^R - s \left[\Lambda_A^R \rho_0^R \left(v_n^R \right)^2 - \Lambda_A^L \rho_0^L \left(v_n^L \right)^2 \right] \right) = 0; \end{aligned} \tag{3.35}$$

Similarly, adding Equations (3.33) and (3.34) would provide the expression for the contact traction vector as:

$$\begin{aligned} t_n^C = &\frac{1}{2} \left(t_n^L + t_n^R \right) + \frac{1}{2} \left(\rho_R^R c_p^R v_n^R - \rho_0^L c_p^L v_n^L \right) - \frac{1}{2} \left(\rho_0^R c_p^R - \rho_0^L c_p^L \right) v_n^C \\ &- \frac{s}{2} \left[\rho_0^R \Lambda_A^R \left(v_n^C - v_n^R \right)^2 + \rho_0^L \Lambda_A^L \left(v_n^C - v_n^L \right)^2 \right]. \end{aligned} \tag{3.36}$$

Alternatively the traction vector can be expressed only in terms of the normal contact velocity and left body variables as:

$$t_n^C = t_n^L + \rho_0^L c_p^L (v_n^C - v_n^L) - s\rho_0^L \Lambda_A^L \left(v_n^C - v_n^L\right)^2. \tag{3.37}$$

These expressions for the normal contact conditions are combined with the acoustic tangential expressions in Equations (3.19) via (3.20) in order to obtain full expressions for contact-stick contact condition. While for the contact-slip contact condition, the normal components are combined with the acoustic tangential components in Equation (3.21) to form the appropriate non-linear contact-slip conditions. Furthermore, for the separation contact conditions the expression in Equation (3.22) are directly applied as the Hugoniot model is no longer valid at the contact interface as it is in tension.

Rigid-Body Contact Conditions

For the particular contact condition where the material on the right is significantly stiffer, the pressure shock wave speed on the stiffer material is approximated to be $c_p^R \approx \infty$. With this definition the non-linear rigid-body normal contact conditions can be defined as:

$$v_n^C = v_n^R; \qquad t_n^C = t_n^L + \rho_0^L c_p^L (v_n^R - v_n^L) - s\rho_0^L \Lambda_A^L \left(v_n^R - v_n^L\right)^2. \tag{3.38}$$

It can be observed that only the velocity from the stiffer body v_n^R, enters the solution. For the particular case, when considering no-slip wall boundary condition, the values of v_n^R is set to zero. Again the normal component can be combined with the acoustic tangential component from Equation (3.23a) to for the complete rigid-body contact condition.

Homogeneous Normal Contact Conditions

The final non-linear contact condition, considered in this research, is for when considering the exact same material properties on the left and right sides of a point of contact. In this scenario, the density and the shock wave speeds are identical and constant for both sides of the contact interface i.e. $\rho_0^L = \rho_0^R = \rho_0$ and $c_p^L = c_p^R = c_p$. By enforcing these conditions in Equation (3.35), and approximating $\Lambda_A^L = \Lambda_A^R = \Lambda_A = \boldsymbol{n} \cdot \boldsymbol{H}^{\text{Ave}} \boldsymbol{N}$,

the v_n^C simplifies to:

$$v_n^C = \frac{1}{\left[c_p - s\Lambda_A\left(v_n^R - v_n^L\right)\right]}\left[\frac{c_p}{2}\left(v_n^L + v_n^R\right) + \frac{1}{2\rho_0}(t_n^R - t_n^L) - \frac{1}{2}s\Lambda_A\left(\left(v_n^R\right)^2 - \left(v_n^L\right)^2\right)\right]; \tag{3.39a}$$

$$= \frac{1}{2}\left(v_n^L + v_n^R\right) + \frac{1}{2\rho_0}\frac{1}{\left[c_p - s\Lambda_A\left(v_n^R - v_n^L\right)\right]}(t_n^R - t_n^L); \tag{3.39b}$$

and t_n^C simplifies to:

$$t_n^C = \frac{1}{2}\left(t_n^L + t_n^R\right) + \frac{\rho_0 c_p}{2}\left(v_n^R - v_n^L\right) - \frac{1}{2}s\rho_0\Lambda_A\left[(v_n^C - v_n^R)^2 + (v_n^C - v_n^L)^2\right]; \tag{3.40a}$$

$$= \frac{1}{2}\left(t_n^L + t_n^R\right) + \frac{\rho_0 c_p}{2}\left(v_n^R - v_n^L\right) - \frac{1}{4}s\rho_0\Lambda_A\left[(v_n^R - v_n^L)^2 + \frac{1}{\rho_0^2\left[c_p - s\Lambda_A\left(v_n^R - v_n^L\right)\right]^2}(t_n^R - t_n^L)^2\right]. \tag{3.40b}$$

It can be observed that these two expressions easily degenerate to the acoustic Riemann solver in Equation (3.29a) by enforcing $s = 0$. It is clear that by using either the acoustic or non-linear Riemann solver approach, the values at the interface are only dependant on the variables at the left and right of the interface therefore does not require any additional constraint or regularisation methodologies unlike tradition methods of computing and enforcing the contact interface. To demonstrate these contact conditions in practice, a simple one-dimensional exact solution will be derived in the next section for the local one-dimensional case, particularised for the impact of two identical linear elastic bars.

3.4 One-Dimensional Contact Analytical Solution

In this section, the contact theory discussed previously in this Chapter and the continuum equations from Chapter 2 will be applied to the simple case of the impact between two one-dimensional identical linear elastic bars. The purpose of this section is to demonstrate that the Riemann solver contact conditions derived in this Chapter, exactly enforce the contact conditions in combination with the first order conservation framework resulting in an exact analytical solution for this scenario. The solution calculated in this section will later be used to benchmark the computational method.

To derive the exact solution for this problem, the first order framework can be expressed

in one-dimension[2] as the one dimensional wave equation for each bar as:

$$\frac{\partial \boldsymbol{\mathcal{U}}_x}{\partial t} + \frac{\partial \boldsymbol{\mathcal{F}}_x}{\partial X} = \mathbf{0}; \qquad \text{with} \qquad \boldsymbol{\mathcal{U}}_x = \begin{bmatrix} \rho_0 v_x \\ F_{xX} \end{bmatrix}; \qquad \boldsymbol{\mathcal{F}}_x = - \begin{bmatrix} P_{xX} \\ v_x \end{bmatrix}, \tag{3.41}$$

with associated jump conditions:

$$c_p \rho_0 [\![v_x]\!] = - [\![P_{xX}]\!] N_X; \tag{3.42a}$$

$$c_p [\![F_{xX}]\!] = - [\![v_x]\!] N_X. \tag{3.42b}$$

where P_{xX} is the one-dimensional component of the linear elastic first Piola-Kirchhoff stress tensor, from Equation (2.26), expressed as:

$$P_{xX} = E(F_{xX} - 1). \tag{3.43}$$

The associated acoustic wave speed c_p is derived for the linear elastic constitutive model, based on the derivation in Section 2.6.2, by substituting Equation (3.42b) into Equation (3.42a) and making use of Equation (3.43) then rearranging as follows:

$$c_p \rho_0 [\![v_x]\!] = -E [\![F_{xX}]\!] N_x = \frac{E}{c_p} [\![v_x]\!];$$

$$c_p = \sqrt{\frac{E}{\rho_0}}. \tag{3.44}$$

In this linear elastic scenario, the shock wave produced by contact is small such that the shock wave speed U_p is equal to the speed of sound within the material c_p, therefore the acoustic wave speed will be used through the derivation. Equation (3.42a) gives the closed form solution for a one-dimensional contact point or interface of a homogeneous linear elastic material. This equation can then be evaluated for each contact stage i.e. in contact and separation following a simplified one-dimensional case of Algorithm 3.1 in order to derive the expressions at the contact interface. The contact stages can be observed in Figure 3.2 and can be described as follows.

[2]Since this is the one-dimensional first order wave equation the conservation of the area map and Jacobian no longer apply condensing the notation.

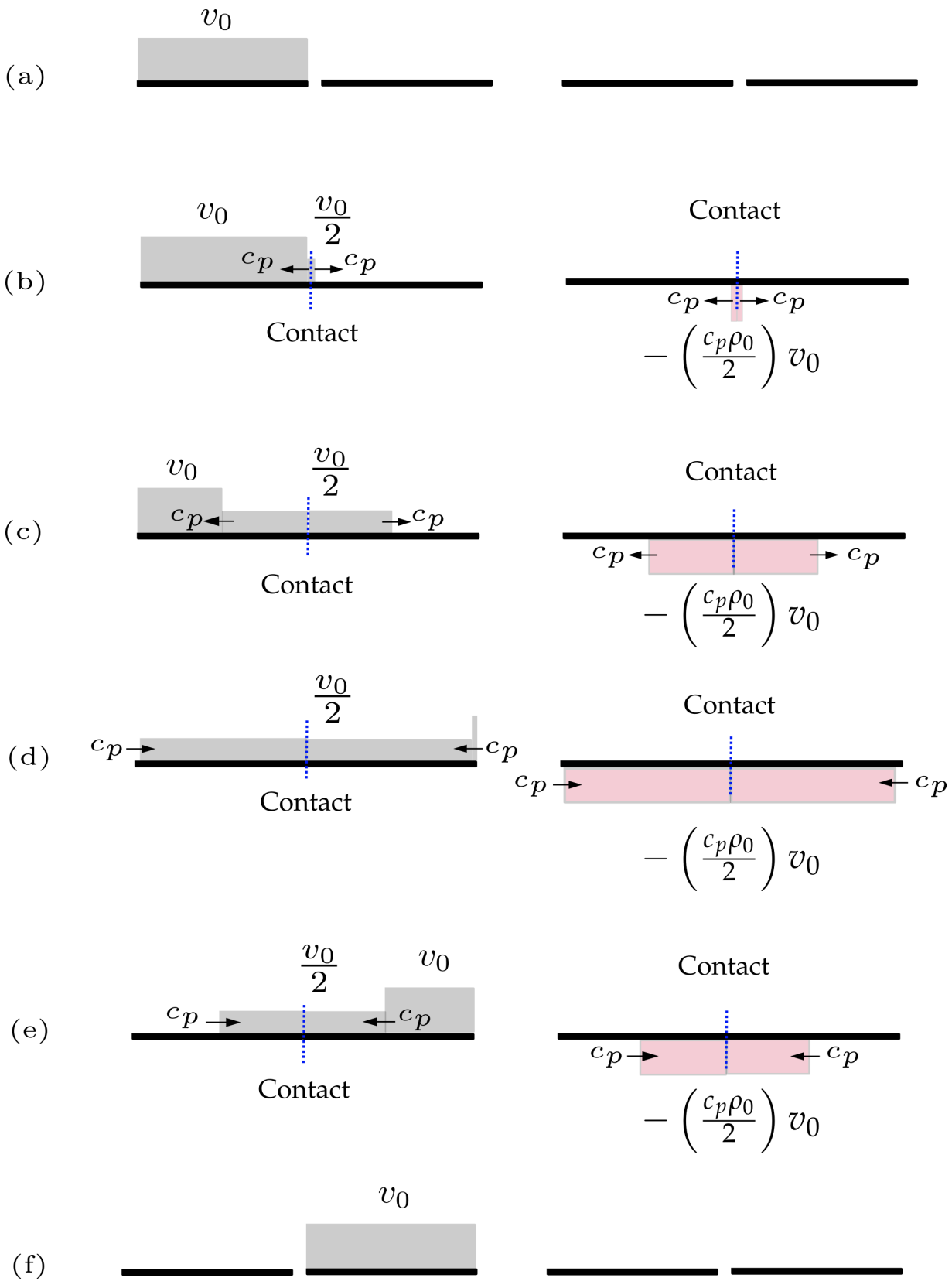


Figure 3.2: One-Dimensional Elastic Bar Impact: Wave solution at different times: (a) $t_0 = 0$, (b) $t_1 = \delta_N/v_0$, (c) $t_2 = t_1 + L/(2c_p)$,(d) $t_3 = t_1 + L/c_p$, (e) $t_4 = t_1 + 3L/(2c_p)$ and (f) $t_5 = t_1 + 2L/c_p$. δ_N is the initial gap between the two bars. The left column represents the velocity profile v_x and right column represents the stress profile (not traction)

At t_0 (Figure 3.2a) the bars have an initial gap δ_N and the left bar has initial velocity v_0 while the right bar is at rest. The time that contact occurs (Figure 3.2b) can simply be determined as $t = \delta_N / v_0$. Once contact occurs, the shock wave generated (both in velocity and stress) by the contact propagates at wave speed c_p in opposite directions along the length of each bar away from the contact interface (Figure 3.2c). When the compressive stress wave reaches the free end of each bar (Figure 3.2d) the wave is reflected back to the contact interface (Figure 3.2e), this wave varies depending on the prescribed boundary condition. In this scenario the reflection from the free end (i.e. traction is zero and velocity is doubled), results in a tensile stress wave which is an exact inversion of the compressive stress wave. While the velocity wave remains unchanged. Finally after the tensile stress wave returns to the contact interface (Figure 3.2f) the bars separate. To obtain the exact mathematical expressions for each contact stage for this scenario, it is possible to use the same procedure from Section 3.3 to obtain the one-dimensional Riemann solution based on the jump condition in Equation (3.42a) and (3.42b). Therefore, drawing comparison with Equation (3.29a) the one-dimensional solution of the contact interface conditions $\{v_x^C, t_x^C\}$ where $v_x^L = v_0$, $v_x^R = 0$ and $t_x^L = t_x^R = 0$ to give:

$$\begin{aligned} v_x^C &= \frac{1}{2}(v_0 + 0) + \frac{1}{2\rho_0 c_p}(0 - 0); \qquad & t_x^C &= \frac{1}{2}(0 + 0) + \frac{\rho_0 c_p}{2}(0 - v_0); \\ &= \frac{1}{2} v_0; & &= -\left(\frac{\rho_0 c_p}{2}\right) v_0, \end{aligned}$$

which is the common (continuous) velocity of both bars and the traction at the contact interface, this is the one-dimensional equivalent of contact-stick. Next this process can be used to determine the release velocities of both bars by substitution of $t^C = 0$ into Equation (3.42a) for each bar respectively since the contact interface cannot support tension to give:

$$c_p \rho_0 \left(v_x^L - v_x^{C,L}\right) = \left(t_x^L - 0\right); \quad c_p \rho_0 \left(v_x^R - v_x^{C,R}\right) = -\left(t_x^R - 0\right);$$

and by substitution of $v_x^L = t_x^L = 0$ for the left bar and $t_x^R = 0$, $v_x^R = v_0$ for the right bar to give:

$$\begin{aligned} v_x^{C,L} &= v_x^L - \frac{t_x^L}{c_p \rho_0}; \quad & v_x^{C,R} &= v_x^R + \frac{t_x^R}{c_p \rho_0}; \\ v_x^{C,L} &= 0; & v_x^{C,R} &= v_0. \end{aligned} \tag{3.45}$$

It is clear from these release velocities that the kinetic energy is transferred fully from the left bar to the right at the point of contact with no loss in energy.

Through the one-dimensional derivation of the exact solution, it can clearly be observed

that by using the acoustic Riemann solver approach for the interface of contact scenarios, the velocity and stress (tractions in multi-dimensions) can be explicitly enforced at a continuum level with exact expression for linear elastic materials. Harnessing this approach for enforcing the contact conditions in a computational method, would provide a significant advantage over traditional FEM based solvers, since the enforcement of the contact constraints in these solvers are either enforced through the penalty or additional constraint methodologies thus reducing the contact interface accuracy or efficiency. Applying this Riemann solver approach to computational contact methodologies forms one of the novelties of this thesis and will be discussed with the computational implementation in Chapter 4.

3.5 Second Law of Thermodynamics

In order assess the stability of the proposed numerical algorithm through the proof of entropy production, in Chapter 4, it is useful to introduce the Hamiltonian $\mathcal{H}(\boldsymbol{X}, t)$ [21, 118]. For the isothermal case, this is a generalised convex entropy function of the system of conservation Equations (2.13), coinciding with the definition of total energy per unit of undeformed volume. Therefore, the Hamiltonian $\mathcal{H}$ can be defined as:

$$\mathcal{H}(\boldsymbol{X}, t) = \hat{\mathcal{H}}(\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J, \boldsymbol{\alpha}) = \frac{1}{2\rho_o}\boldsymbol{p} \cdot \boldsymbol{p} + \mathcal{E}(\boldsymbol{F}, \boldsymbol{H}, J, \boldsymbol{\alpha}), \tag{3.46}$$

which represents the summation of the kinetic energy per unit of undeformed volume (the first term on the right hand side of Equation (3.46)) and the internal energy $\mathcal{E}$ expressed in terms of the three deformation measures $\{\boldsymbol{F}, \boldsymbol{H}, J\}$ and a set of state variables [128–130] collected in the form of a tensor $\boldsymbol{\alpha}$ [3]. It is important to note that $\mathcal{H}(\boldsymbol{X}, t)$ and $\hat{\mathcal{H}}(\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J, \boldsymbol{\alpha})$ represent alternative functional representations of the same quantity.

It is useful to revisit the second law of thermodynamics when written in terms of the Hamiltonian. Taking the derivatives of $\hat{\mathcal{H}}$ from Equation (3.46) with respect to its arguments, the time rate of the Hamiltonian when considering one body potentially in

[3] For example, plastic deformation or similar.

contact, can be obtained via the chain rule as follows:

$$\begin{aligned}\frac{d}{dt}\int_{\Omega_0}\mathcal{H}\,dV &= \int_{\Omega_0}\frac{\partial\hat{\mathcal{H}}(\boldsymbol{p},\boldsymbol{F},\boldsymbol{H},J,\boldsymbol{\alpha})}{\partial t}\,dV,\\ &= \int_{\Omega_0}\left(\frac{\partial\hat{\mathcal{H}}}{\partial\boldsymbol{p}}\cdot\frac{\partial\boldsymbol{p}}{\partial t}+\frac{\partial\hat{\mathcal{H}}}{\partial\boldsymbol{F}}:\frac{\partial\boldsymbol{F}}{\partial t}+\frac{\partial\hat{\mathcal{H}}}{\partial\boldsymbol{H}}:\frac{\partial\boldsymbol{H}}{\partial t}+\frac{\partial\hat{\mathcal{H}}}{\partial J}\frac{\partial J}{\partial t}+\frac{\partial\hat{\mathcal{H}}}{\partial\boldsymbol{\alpha}}:\frac{\partial\boldsymbol{\alpha}}{\partial t}\right)dV,\\ &= \int_{\Omega_0}\left(\boldsymbol{v}\cdot\frac{\partial\boldsymbol{p}}{\partial t}+\boldsymbol{\Sigma}_{\boldsymbol{F}}:\frac{\partial\boldsymbol{F}}{\partial t}+\boldsymbol{\Sigma}_{\boldsymbol{H}}:\frac{\partial\boldsymbol{H}}{\partial t}+\Sigma_J\frac{\partial J}{\partial t}+\frac{\partial\mathcal{E}}{\partial\boldsymbol{\alpha}}:\frac{\partial\boldsymbol{\alpha}}{\partial t}\right)dV,\\ &= \int_{\Omega_0}\left(\boldsymbol{v}\cdot\frac{\partial\boldsymbol{p}}{\partial t}+(\boldsymbol{\Sigma}_{\boldsymbol{F}}+\boldsymbol{\Sigma}_{\boldsymbol{H}}\times\boldsymbol{F}+\Sigma_J\boldsymbol{H}):\boldsymbol{\nabla}_0\boldsymbol{v}+\frac{\partial\mathcal{E}}{\partial\boldsymbol{\alpha}}:\frac{\partial\boldsymbol{\alpha}}{\partial t}\right)dV,\\ &= \int_{\Omega_0}\left(\boldsymbol{v}\cdot\frac{\partial\boldsymbol{p}}{\partial t}+\boldsymbol{P}:\boldsymbol{\nabla}_0\boldsymbol{v}+\frac{\partial\mathcal{E}}{\partial\boldsymbol{\alpha}}:\frac{\partial\boldsymbol{\alpha}}{\partial t}\right)dV,\end{aligned} \tag{3.47}$$

where, Equations (2.13) and (2.19) have been substituted in the third and fifth lines of Equation (3.47), respectively. Subsequently, the linear momentum balance principle from Equation (2.6) can be substituted into Equation (3.47) to give:

$$\frac{d}{dt}\int_{\Omega_0}\mathcal{H}\,dV = \int_{\Omega_0}\left[\boldsymbol{v}\cdot(\rho_0\boldsymbol{b})+\boldsymbol{v}\cdot\mathrm{DIV}\boldsymbol{P}+\boldsymbol{P}:\boldsymbol{\nabla}_0\boldsymbol{v}+\frac{\partial\mathcal{E}}{\partial\boldsymbol{\alpha}}:\frac{\partial\boldsymbol{\alpha}}{\partial t}\right]dV. \tag{3.48}$$

By recalling that $\boldsymbol{v}\cdot\mathrm{DIV}\boldsymbol{P}+\boldsymbol{P}:\boldsymbol{\nabla}_0\boldsymbol{v}=\mathrm{DIV}\left(\boldsymbol{P}^T\boldsymbol{v}\right)$, the above equation reduces to:

$$\frac{d}{dt}\int_{\Omega_0}\mathcal{H}\,dV = \int_{\Omega_0}\left[\boldsymbol{v}\cdot(\rho_0\boldsymbol{b})+\mathrm{DIV}(\boldsymbol{P}^T\boldsymbol{v})+\frac{\partial\mathcal{E}}{\partial\boldsymbol{\alpha}}:\frac{\partial\boldsymbol{\alpha}}{\partial t}\right]dV. \tag{3.49}$$

By performing integration by parts of the divergence term in Equation (3.49), and after some re-arrangement, it yields:

$$\frac{d}{dt}\int_{\Omega_0}\mathcal{H}\,dV-\dot{\Pi}_{\mathrm{ext}}=-\dot{D}, \tag{3.50}$$

where $\dot{\Pi}_{\mathrm{ext}}$ denotes the power introduced by external forces, defined as:

$$\dot{\Pi}_{\mathrm{ext}}=\int_{\Omega_0}\boldsymbol{v}\cdot(\rho_0\boldsymbol{b})\,dV+\int_{\partial\Omega_0\setminus\Gamma}\boldsymbol{v}^B\cdot\boldsymbol{t}^B\,dA+\int_{\Gamma}\boldsymbol{v}^C\cdot\boldsymbol{t}^C\,dA. \tag{3.51}$$

Here, Γ represents the boundary faces on contact region and $\partial\Omega_0\setminus\Gamma$ represents the remaining boundary faces that are not in contact. In the above expression, the first term on the right hand side represents external force acting on a body, the second term represents the non-contact boundary forces obtained via the enforcement of standard Neumann or Dirichlet boundary conditions, and the third term represents the contact boundary forces describing appropriately the contact-impact phenomenon. To further examine this, consider the case of elasto-plasticity [112, 131, 132] where the elastic en-

ergy is expressed in terms of elastic left Cauchy-Green tensor $\boldsymbol{b}_e = \boldsymbol{F}\boldsymbol{C}_p^{-1}\boldsymbol{F}^T$. Therefore in this case, the internal state variable is the inverse of the plastic right Cauchy Green tensor as, $\boldsymbol{\alpha} = \boldsymbol{C}_p^{-1}$. With this definition, the rate of plastic dissipation $\dot{D}$ described in Equation (3.50) becomes:

$$\dot{D} = -\int_{\Omega_0} \frac{\partial \mathcal{E}}{\partial \boldsymbol{C}_p^{-1}} : \frac{\partial \boldsymbol{C}_p^{-1}}{\partial t}\, dV. \tag{3.52}$$

In the scenario where the rate of plastic strain $\dot{\bar{\varepsilon}}_p$ has been defined as the work conjugate to the von-Mises equivalent stress $\bar{\tau}$ [131], the equation above can be alternatively expressed as [131]:

$$\dot{D} = \int_{\Omega_0} \dot{\bar{\varepsilon}}_p \bar{\tau}\, dV; \qquad \bar{\tau} = \sqrt{\frac{3}{2}\,(\hat{\boldsymbol{\tau}} : \hat{\boldsymbol{\tau}})}, \tag{3.53}$$

where $\hat{\boldsymbol{\tau}}$ represents the deviatoric component of the Kirchhoff stress. It is important to note here that in the above expression, the rate of dissipation is always non-negative, that is $\dot{D} \geq 0$, Equation (3.50) can be transformed into the following inequality:

$$\frac{d}{dt}\int_{\Omega_0} \mathcal{H}\, dV - \dot{\Pi}_{\text{ext}} \leq 0, \tag{3.54}$$

which represents a valid expression for the global statement of the second law of thermodynamics [133]. Satisfaction of inequality (3.54) is a necessary *ab initio* condition to ensure stability, otherwise referred to as the Coleman–Noll procedure [114]. This key concept will be further exploited in the next Chapter at a semi-discrete level.

Part III

Computational Implementation

Chapter 4

Numerical Scheme

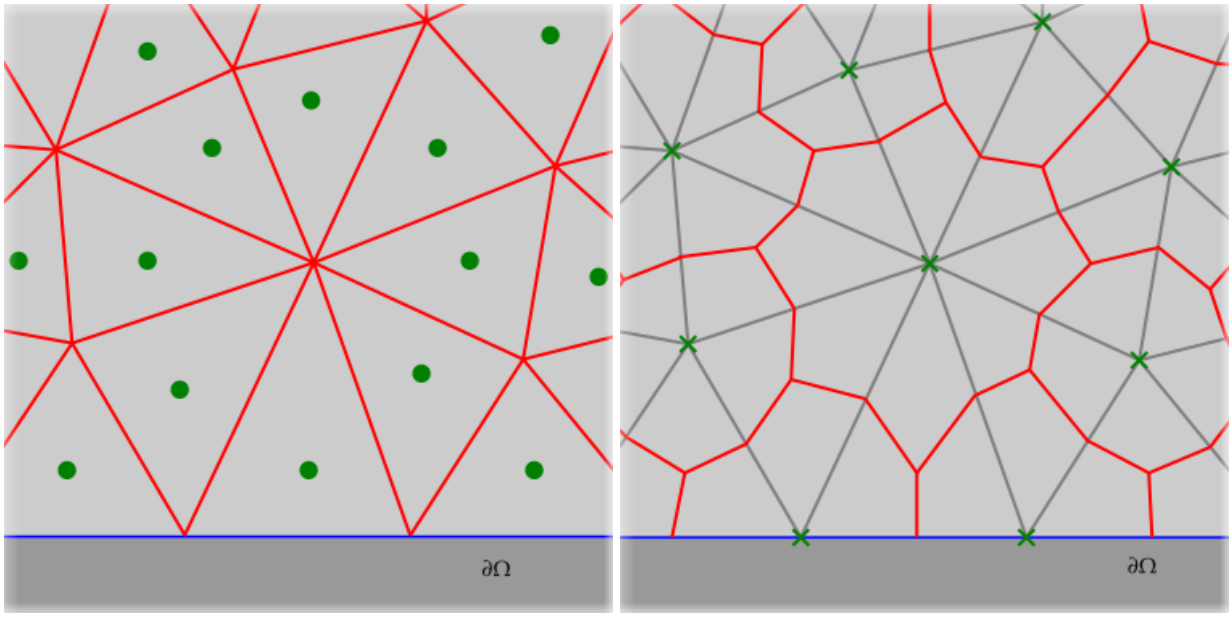
4.1 Preliminaries

In this Chapter the continuum formulation for contact dynamics discussed in Part II, will be discretised for computational implementation. In this research the first order framework is spatially discretised using the finite volume method, introduced in Section 4.2 and particularised for the vertex centred finite volume method (VCFVM) with linear flux reconstruction and limiter in Section 4.3. The stability of this semi-discrete formulation is assessed in terms of the entropy production in Section 4.3.3. The semi-discrete equations will then be extended to explicit dynamics with a two stage Runge-Kutta time integrator, explored in Section 4.4.

4.2 Finite Volume Method

In order for the continuum formulation discussed in Part II to be solved computationally the first order framework must be discretised with an appropriate computational method. As discussed in Part I, traditional computational methodologies for explicit contact dynamics, in particular the FEM, have a series of shortcomings stemming from the computational method, therefore an alternate computational method will be considered in this research. By formulating the solid dynamic problem as the set of first order conservation equations, in terms of fluxes, its easy to draw comparisons with CFD conservation formulations. These formulations are commonly discretised using the Finite Volume Method (FVM), first introduced separately by McDonald [134], MacCormack [135] and extended to three dimensions by Rizzi [136] which are based on direct discretisation of the global conservation equations [14]. The advantage of this

direct integral discretisation is that the global conservation of the primary variables[1] are automatically satisfied at a discrete level and fluxes are not required to be continuous therefore are suitable for scenarios where shock waves are present [14, 137, 138], an important property to consider for contact scenarios. To discretise a conservation equation using the FVM, the domain is split into a series of distinct finite volumes known as control volumes, the definition of which varies based on the type of FVM, either cell-centred or vertex-centred, based on the location of the stored primary conserved variables, as shown in Figure 4.1.



(a) Cell-Centred Finite Volume (b) Vertex-Centred Finite Volume

Figure 4.1: Control volume definitions (red) for CCFVM and VCFVM with primary variable locations (green) in two dimensions

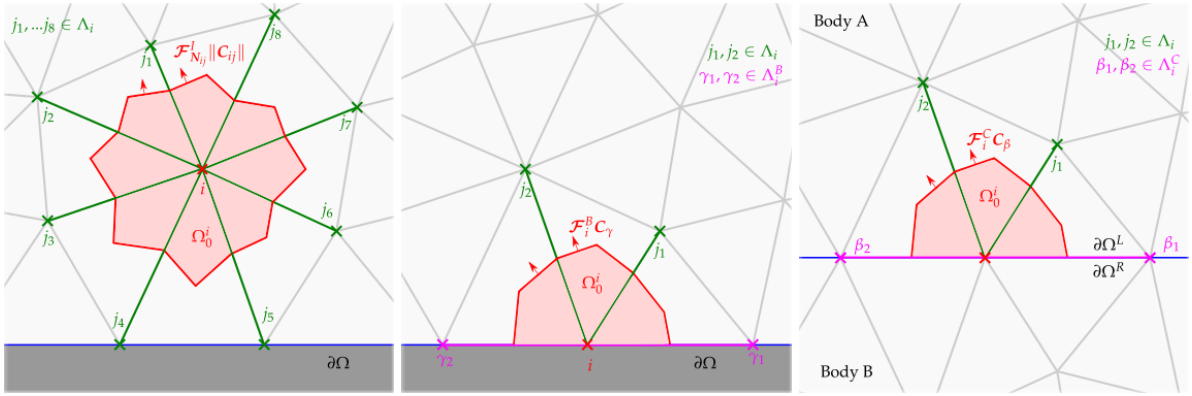
In the Cell Centred Finite Volume Method (CCFVM), the control volumes are defined by a traditional primary mesh split into nodes, edges and cells (elements) with the individual control volumes defined as the cells bounded by the associated cell edges/faces, see Figure 4.1a. In this scheme the conserved variables are located at the centre of each cell which requires projection to the boundary faces in order to compute the boundary conditions. On the other hand, VCFVM is where the control volumes are defined centred around each node (vertex) by constructing a secondary or dual mesh onto the primary mesh, see Figure 4.1b. This gives increased flexibility to the definition of the control volume compared with the cell-centred approach, however requires additional pre-processing to create the dual mesh. This flexible control volume definition results in a less computational intensive scheme for tetrahedral meshes since there are 5 to 6 times

[1]For example in CFD, mass, momentum and energy.

more cells than vertices in a tetrahedral mesh thus requiring less evaluation and storage of the conserved variables [139]. This provides a significant benefit in large strain solid dynamics where complex computationally intensive constitutive models require evaluation at each conserved variable location, significantly increasing the computational cost. Another key benefit of VCFVM is that the conserved variables are stored directly on the boundary, therefore do not require projection and boundary conditions can be strongly enforced unlike CCFVM. These differences naturally benefit VCFVM in contact dynamic scenarios as the contact constraints can be strongly enforced at the boundary vertices and can provide fast solutions for complex material models, therefore VCFVM will be the spatial discretisation for this research and discussed in the following section.

4.3 Vertex Centred Finite Volume Method

As mentioned previously, VCFVM is a node (vertex) based computational discretisation where the conserved variables are solved for and stored at the nodes of the primary mesh. The domain is first discretised using traditional unstructured meshing techniques to create a primary mesh. This primary mesh is then split into a series of control volumes through a secondary or dual mesh which can be constructed by a variety of techniques, in this research the median dual approach for tetrahedral meshes [140] will be utilised which constructs the dual mesh by connecting the edge midpoints to cell centroids. In order to apply this spatial computational method the notation from Figure 4.2 for two dimensions will be used to discretise the first order conservation equations in the following sections.



(a) (b) (c)

Figure 4.2: Notation for VCFVM in two dimensions (a) Control volume for interior node i (b) Control volume for boundary node i (c) Control volume for contact boundary node i

4.3.1 Semi-Discrete Conservation Equation

To apply the VCFVM to the first order framework, first Equation (2.13) is integrated in space for any undeformed control volume Ω_0^i and applying the Green-Gauss divergence theorem [14], the equation becomes:

$$\Omega_0^i \frac{d\boldsymbol{\mathcal{U}}_i}{dt} = -\int_{\partial\Omega_0^i} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} dA + \Omega_0^i \boldsymbol{\mathcal{S}}_i, \tag{4.1}$$

where $\boldsymbol{\mathcal{U}}_i$ and $\boldsymbol{\mathcal{S}}_i$ are the conservation variables and source terms associated with control volume i, respectively. While $\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}$ is the flux vector previously defined in Equation (2.16). By following the approach in [114] the surface integral can be approximated by an upwinding numerical flux and then reformulated for computational efficiency in Total Lagrangian form to[2]:

$$\Omega_0^i \frac{d\boldsymbol{\mathcal{U}}_i}{dt} = -\left(\sum_{j\in\Lambda_i} \boldsymbol{\mathcal{F}}^I_{\boldsymbol{N}_{ij}} \|\boldsymbol{C}_{ij}\| + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{\mathcal{F}}_i^B \boldsymbol{C}_\gamma + \sum_{\beta\in\Lambda_i^C} \boldsymbol{\mathcal{F}}_i^C \boldsymbol{C}_\beta \right) + \Omega_0^i \boldsymbol{\mathcal{S}}_i, \tag{4.2}$$

where $j \in \Lambda_i$ is the set of connecting control volumes j that are associated with control volume i and $\boldsymbol{C}_{ij}$ is the area vector associated with edge $i-j$. For a mean undeformed area vector $\boldsymbol{C}_{ij}$ satisfies the reciprocal relation: $\boldsymbol{C}_{ij} = -\boldsymbol{C}_{ji}$. Similarly $\gamma \in \Lambda_i^B$ and $\beta \in \Lambda_i^C$ are the sets of connecting non-contact boundary faces γ and contact boundary faces β that are associated with control volume i with $\boldsymbol{C}_{\gamma,\beta} = \frac{A_{\gamma,\beta}}{3}\boldsymbol{N}_{\gamma,\beta}$ representing the respective boundary area vector. The terms within the parenthesis are the evaluation of the control volume interior interface flux $\boldsymbol{\mathcal{F}}^I_{\boldsymbol{N}_{ij}}$, non-contact boundary flux $\boldsymbol{\mathcal{F}}_i^B$ and contact boundary flux $\boldsymbol{\mathcal{F}}_i^C$. The internal interface flux $\boldsymbol{\mathcal{F}}^I_{\boldsymbol{N}_{ij}} = \boldsymbol{\mathcal{F}}^{\boldsymbol{C}}_{\boldsymbol{N}}\left(\boldsymbol{\mathcal{F}}^L_{ij}, \boldsymbol{\mathcal{F}}^R_{ij}, \boldsymbol{N}_{ij}\right)$ can be evaluated from the contact-stick condition[3] which depends on both sides of the midpoint of edge ij. The non-contact and contact boundary flux $\boldsymbol{\mathcal{F}}_i^{B,C}$ are evaluated based on the weighted average approach [141] and is defined in three dimensions as:

$$\boldsymbol{\mathcal{F}}_i^B = \frac{6\boldsymbol{\mathcal{F}}_i^B + \boldsymbol{\mathcal{F}}_j^B + \boldsymbol{\mathcal{F}}_k^B}{8}; \quad \boldsymbol{\mathcal{F}}_i^C = \frac{6\boldsymbol{\mathcal{F}}_i^C + \boldsymbol{\mathcal{F}}_j^C + \boldsymbol{\mathcal{F}}_k^C}{8}, \tag{4.3}$$

where j and k are the other nodes that define boundary face γ or β with node i. The non-contact boundary flux $\boldsymbol{\mathcal{F}}_i^B$ can be evaluated based on appropriate Dirichlet or Neumann boundary conditions (refer to [100]) while the contact boundary flux $\boldsymbol{\mathcal{F}}_i^C$ can be

[2]For hypervelocity impacts, fracture, or large sliding interfaces Arbitrary Lagrangian Eulerian (ALE) formulation may be required to address excessively large mesh/geometry distortion.

[3]The interior numerical flux between contacting control volumes can be considered a particular case of contact-stick with infinite friction hence Equation (3.20) for the acoustic Riemann solver can be implemented to approximate the interface flux.

evaluated based on appropriate contact boundary conditions from Section 3.3. It is worth noting that when boundary face β is not in contact, $\boldsymbol{C}_\beta = \boldsymbol{0}$.

Equation (4.1) is particularised for the full first order conservation framework as:

$$\Omega_0^i \frac{d\left(\rho_0 \boldsymbol{v}\right)_i}{dt} = \sum_{j\in\Lambda_i} \boldsymbol{t}^I \left\|\boldsymbol{C}_{ij}\right\| + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{t}_i^B \left\|\boldsymbol{C}_\gamma\right\| + \sum_{\beta\in\Lambda_i^C} \boldsymbol{t}_i^C \left\|\boldsymbol{C}_\beta\right\| + \Omega_0^i \left(\rho_0 \boldsymbol{b}_0^i\right); \tag{4.4a}$$

$$\Omega_0^i \frac{d\boldsymbol{F}_i}{dt} = \sum_{j\in\Lambda_i} \boldsymbol{v}^I \otimes \boldsymbol{C}_{ij} + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{v}_i^B \otimes \boldsymbol{C}_\gamma + \sum_{\beta\in\Lambda_i^B} \boldsymbol{v}_i^C \otimes \boldsymbol{C}_\beta; \tag{4.4b}$$

$$\Omega_0^i \frac{d\boldsymbol{H}_i}{dt} = \sum_{j\in\Lambda_i} \boldsymbol{F}^{Ave} \boldsymbol{\times} \left(\boldsymbol{v}^I \otimes \boldsymbol{C}_{ij}\right) + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{F}_i \boldsymbol{\times} \left(\boldsymbol{v}_i^B \otimes \boldsymbol{C}_\gamma\right) + \sum_{\beta\in\Lambda_i^C} \boldsymbol{F}_i \boldsymbol{\times} \left(\boldsymbol{v}_i^C \otimes \boldsymbol{C}_\beta\right); \tag{4.4c}$$

$$\Omega_0^i \frac{dJ_i}{dt} = \sum_{j\in\Lambda_i} \boldsymbol{v}^I \cdot \left(\boldsymbol{H}^{Ave} \boldsymbol{C}_{ij}\right) + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{v}_i^B \cdot \left(\boldsymbol{H}_i \boldsymbol{C}_\gamma\right) + \sum_{\beta\in\Lambda_i^C} \boldsymbol{v}_i^C \cdot \left(\boldsymbol{H}_i \boldsymbol{C}_\beta\right); \tag{4.4d}$$

where $\boldsymbol{v}^I$, $\boldsymbol{t}^I$ are the internal interface velocity and traction computed from the Riemann solver for contact-stick in Equation (3.20), $\boldsymbol{t}_i^B$ and $\boldsymbol{v}_i^B$ are the boundary traction and velocity respectively depending on the appropriate boundary conditions, referring to [100]. $\boldsymbol{t}_i^C$ and $\boldsymbol{v}_i^C$ are the contact traction and velocity respectively computed based on the conditions in Section 3.3, $[\cdot]^{Ave} = \frac{1}{2}\left([\cdot] + [\cdot]\right)$ are the average of the respective variable with the adjacent control volume. By substituting the contact interface variables directly for each control volume, thus producing Equation (4.4), this does not ensure discrete satisfaction of the involutions defined in Equation (2.11). Following the work Hassan *et al.* [114], one viable option to ensure the discrete satisfaction of the involutions is to approximate the updates of $\boldsymbol{F}$ (Equation (4.4b)) and $\boldsymbol{H}$ (Equation (4.4c)) by using central difference approximations for $\boldsymbol{v}^I$, for example, only using the first term of Equations (3.17a) and (3.19a) for the acoustic conditions, while removing second term containing the upwinding stabilisation. For consistency, $\boldsymbol{F}^{Ave}$ and $\boldsymbol{H}^{Ave}$ will be replaced with $\boldsymbol{F}_i$ and $\boldsymbol{H}_i$. Assuming the jump in traction is dominated by the jump in

pressure, refer to Section 3.3, the geometric conservation equations reduce to:

$$\Omega_0^i \frac{d\boldsymbol{F}_i}{dt} = \sum_{j\in\Lambda_i} \boldsymbol{v}_{ij}^{WAve} \otimes \boldsymbol{C}_{ij} + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{v}_i^B \otimes \boldsymbol{C}_\gamma + \sum_{\beta\in\Lambda_i^B} \boldsymbol{v}_i^C \otimes \boldsymbol{C}_\beta; \tag{4.5a}$$

$$\Omega_0^i \frac{d\boldsymbol{H}_i}{dt} = \boldsymbol{F}_i \times \left(\sum_{j\in\Lambda_i} \boldsymbol{v}_{ij}^{WAve} \otimes \boldsymbol{C}_{ij} + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{v}_i^B \otimes \boldsymbol{C}_\gamma + \sum_{\beta\in\Lambda_i^C} \boldsymbol{v}_i^C \otimes \boldsymbol{C}_\beta \right); \tag{4.5b}$$

$$\begin{aligned} \Omega_0^i \frac{dJ_i}{dt} = \boldsymbol{H}_i : & \left(\sum_{j\in\Lambda_i} \boldsymbol{v}_{ij}^{WAve} \otimes \boldsymbol{C}_{ij} + \sum_{\gamma\in\Lambda_i^B} \boldsymbol{v}_i^B \otimes \boldsymbol{C}_\gamma + \sum_{\beta\in\Lambda_i^C} \boldsymbol{v}_i^C \otimes \boldsymbol{C}_\beta \right) \\ & + \sum_{j\in\Lambda_i} S_{ij}^{\Sigma_J} \left(\Sigma_J^R - \Sigma_J^L \right); \end{aligned} \tag{4.5c}$$

where $\boldsymbol{v}_{ij}^{WAve}$ is the weighted average velocity defined as:

$$\boldsymbol{v}^{WAve} = v_n^{WAve}\boldsymbol{n} + \boldsymbol{v}_t^{WAve}, \tag{4.6}$$

where:

$$v_n^{WAve} = \frac{c_p^L \rho_0^L v_n^L + c_p^R \rho_0^R v_n^R}{c_p^L \rho_0^L + c_p^R \rho_0^R}; \quad \boldsymbol{v}_t^{WAve} = \frac{c_s^L \rho_0^L \boldsymbol{v}_t^L + c_s^R \rho_0^R \boldsymbol{v}_t^R}{c_s^L \rho_0^L + c_s^R \rho_0^R}, \tag{4.7}$$

at the interface of edge ij, $\boldsymbol{v}_i^B$ and $\boldsymbol{v}_i^C$ are the non-contact and contact boundary velocities without stabilisation based on the boundary conditions from Section 3.3 and [100]. Importantly, $S_{ij}^{\Sigma_J}$ is the strictly positive Riemann upwind stabilisation based on the pressure component (see Section 3.3) which cannot be included in Equations (4.5a) and (4.5b), but since there is no involution for Equation (4.5c), the stabilisation term can be included and is defined as the stabilisation terms from Equation (3.20) as:

$$S_{ij}^{\Sigma_J} = \frac{1}{2\rho_0 c_p} \frac{\boldsymbol{c}_{ij} \cdot \boldsymbol{c}_{ij}}{\|\boldsymbol{C}_{ij}\|}; \tag{4.8}$$

where $\boldsymbol{c}_{ij} = \boldsymbol{H}^{Ave}\boldsymbol{C}_{ij}$. It can be observed that in Equation (4.5) that the stabilisation terms are not present in Equation (4.5a) and (4.5b) thus ensuring that these equations satisfy the involutions. The mappings $\boldsymbol{F}_i$, $\boldsymbol{H}_i$ in Equations (4.5b) and (4.5c) are naturally curl and divergence free as the equations are formulated in terms of a material discrete gradient of a continuous velocity field [103]. By utilising the equations above and taking the flux variables, $\boldsymbol{\mathcal{F}}_{ij}^L\left(\boldsymbol{\mathcal{U}}_{ij}^L\right)$ and $\boldsymbol{\mathcal{F}}_{ij}^R\left(\boldsymbol{\mathcal{U}}_{ij}^R\right)$, as piecewise constants (i.e. $\boldsymbol{\mathcal{U}}_{ij}^L = \boldsymbol{\mathcal{U}}_i$ and $\boldsymbol{\mathcal{U}}_{ij}^R = \boldsymbol{\mathcal{U}}_j$), the resulting solution would be first order in space [14, 139]. Therefore to ensure accuracy of the scheme, a linear reconstruction procedure is utilised with a Barth-Jespersen slope limiter [142] for the flux variables, $\boldsymbol{\mathcal{F}}_{ij}^L$ and $\boldsymbol{\mathcal{F}}_{ij}^R$ to compute the variables at the control volume interfaces and will be discussed in the following section.

4.3.2 Flux Reconstruction and Limiter

In order to obtain a second order spatial accuracy, a linear reconstruction procedure is utilised. This procedure can be described by referring to the one-dimensional representation in Figure 4.3, where the red dots represent the nodes (position of conserved variable), the blue rectangles represent each control volume and the black lines represent the linear reconstruction of the fluxes. For a given node i which has two connected nodes j_1 and j_2, the flux $\mathcal{F}_{ij}$ at each control volume interface along each edge ij, are reconstructed (black line) from conserved variable $\mathcal{U}_i$ by first computing the flux gradient which is defined in multi-dimensional form following the Green-Gauss approach [142] as:

$$\Omega_0^i \boldsymbol{G}_i = \sum_{j \in \Lambda_i} \frac{\boldsymbol{\mathcal{F}}\left(\boldsymbol{\mathcal{U}}_i\right) + \boldsymbol{\mathcal{F}}\left(\boldsymbol{\mathcal{U}}_j\right)}{2} \otimes \boldsymbol{C}_{ij} + \sum_{\gamma \in \Lambda_i^{B,C}} \left(\boldsymbol{\mathcal{F}}\left(\boldsymbol{\mathcal{U}}_i\right) \otimes \boldsymbol{N}_\gamma\right) \frac{A_\gamma}{3}, \tag{4.9}$$

where the first term on the right hand side is simply the sum of the area weighted average of internal fluxes while the second term is the sum of the area weighted average of the boundary fluxes. With this flux gradient, the fluxes at the left and right sides of the flux interface ij can be linearly reconstructed based on the distance to the interface from the respective node as:

$$\boldsymbol{\mathcal{F}}_{ij}^L = \boldsymbol{\mathcal{F}}\left(\boldsymbol{\mathcal{U}}_i\right) + \tfrac{1}{2}\Phi_i \boldsymbol{G}_i \Delta \boldsymbol{X}; \quad \boldsymbol{\mathcal{F}}_{ij}^R = \boldsymbol{\mathcal{F}}\left(\boldsymbol{\mathcal{U}}_j\right) - \tfrac{1}{2}\Phi_j \boldsymbol{G}_j \Delta \boldsymbol{X}, \tag{4.10}$$

where $\Delta \boldsymbol{X} = \left(\boldsymbol{X}_j - \boldsymbol{X}_i\right)$, is the difference between nodal positions and $\Phi_{i,j}$ is the slope limit function which can be defined by any appropriate limiter, in this research the slope limiter will be defined using the Barth-Jespersen limiter [142] outlined per component in Algorithm 4.1. It can be observed that for a limiter where a component $\Phi_i = 1$ the complete linear reconstruction is employed, however in the vicinity of a shock this complete linear reconstruction would result in local extrema at the interface therefore a limiter is used to constrain the reconstruction such that no local extrema is created [143] but this sacrifices the order of accuracy near the shock.

4.3.3 Entropy Production

In this section, the inequality from Equation (3.54) in Section 3.5, is assessed for the set of semi-discrete Equations, (4.4a), (4.5a), (4.5b) and (4.5c). For illustrative purposes, the body under consideration is said to be acoustic and homogeneous. Additionally, in the following proof, it is assumed that piecewise constant approximation (first order in space) is used for variables across each control volume known as the Godunov approach [126, 127]. Making use of Equations (3.53), the semi-discrete counterpart of

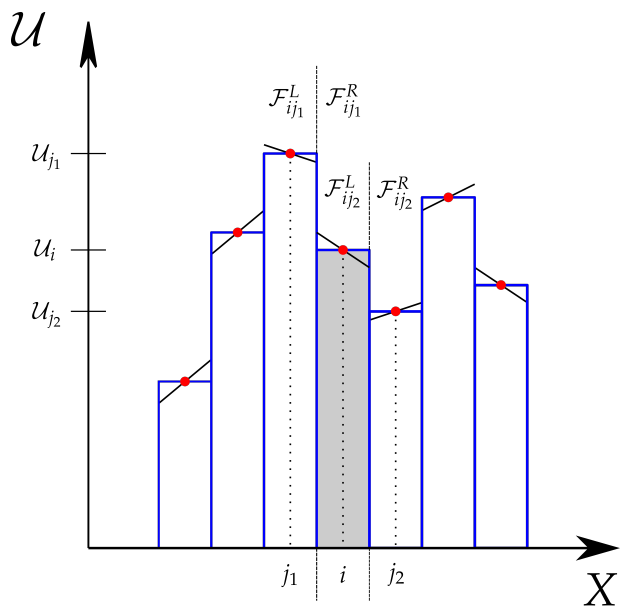


Figure 4.3: One-dimensional representation of piecewise linear reconstruction

Equation (3.47) is:

$$\sum_i \Omega_0^i \frac{d\mathcal{H}_i}{dt} = \sum_i \Omega_0^i \left[\boldsymbol{v}_i \cdot \frac{d\boldsymbol{p}_i}{dt} + \boldsymbol{\Sigma}^i_{\boldsymbol{F}} : \frac{d\boldsymbol{F}_i}{dt} + \boldsymbol{\Sigma}^i_{\boldsymbol{H}} : \frac{d\boldsymbol{H}_i}{dt} + \Sigma^i_J \frac{dJ_i}{dt} - \frac{1}{\Omega_0^i}\dot{D}_i \right]; \tag{4.11a}$$

$$= \sum_i \Omega_0^i \left[\boldsymbol{v}_i \cdot \frac{d\boldsymbol{p}_i}{dt} + \left(\boldsymbol{\Sigma}^i_{\boldsymbol{F}} + \boldsymbol{\Sigma}^i_{\boldsymbol{H}} \times \boldsymbol{F}_i + \Sigma^i_J \boldsymbol{H}_i \right) : \frac{d\boldsymbol{F}_i}{dt} - \frac{1}{\Omega_0^i}\dot{D}_i \right] + \sum_i \sum_{j \in \Lambda_i} \Sigma^i_J S^{\Sigma_J}_{ij} (\Sigma^j_J - \Sigma^i_J); \tag{4.11b}$$

$$= \sum_i \Omega_0^i \left[\boldsymbol{v}_i \cdot \frac{d\boldsymbol{p}_i}{dt} + \boldsymbol{P}_i : \frac{d\boldsymbol{F}_i}{dt} \right] + \sum_i \sum_{j \in \Lambda_i} \Sigma^i_J S^{\Sigma_J}_{ij} (\Sigma^j_J - \Sigma^i_J) - \sum_i \dot{D}_i, \tag{4.11c}$$

where, Equations (4.5a) - (4.5c) and Equation (2.19) have been substituted into the second and third lines of (4.11), respectively. Subsequently, the linear momentum conservation Equation (4.4a), the deformation gradient conservation Equation (4.5a) can be substituted and, after some algebra, gives:

$$\begin{aligned} \sum_i \Omega_0^i \frac{d\mathcal{H}_i}{dt} = &\sum_i \sum_{j \in \Lambda_i} \frac{1}{2} \left[\boldsymbol{t}_i \cdot \boldsymbol{v}_j - \boldsymbol{t}_j \cdot \boldsymbol{v}_i \right] \|\boldsymbol{C}_{ij}\| + \sum_i \sum_{j \in \Lambda_i} \boldsymbol{v}_i \cdot \boldsymbol{S}^{\boldsymbol{v}}_{ij} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \\ &+ \sum_i \sum_{j \in \Lambda_i} \Sigma^i_J S^{\Sigma_J}_{ij} (\Sigma^j_J - \Sigma^i_J) - \sum_i \dot{D}_i + \dot{\Pi}_{\text{ext}}. \end{aligned} \tag{4.12a}$$

Here, $\dot{\Pi}_{\text{ext}}$ denotes the semi-discrete power contribution, expressed as:

$$\dot{\Pi}_{\text{ext}} = \sum_i \Omega_0^i \boldsymbol{v}_i \cdot (\rho_0 \boldsymbol{b}_0^i) + \sum_\gamma A_0^\gamma \boldsymbol{t}_i^B \cdot \boldsymbol{v}_i^B + \sum_\beta A_0^\beta \boldsymbol{t}_i^C \cdot \boldsymbol{v}_i^C, \tag{4.13}$$

Algorithm 4.1: Component-Wise Barth and Jespersen Slope Limiter

Input : $\mathcal{U}_i, \mathcal{U}_j$
Output: Φ_i

1. Compute minimum and maximum values of variable $\mathcal{U}$:

$$\mathcal{U}_i^{min} = \min_{j \in \Lambda_i} \left(\mathcal{U}_i, \mathcal{U}_j\right); \quad \mathcal{U}_i^{max} = \max_{j \in \Lambda_i} \left(\mathcal{U}_i, \mathcal{U}_j\right)$$

2. Compute an unlimited reconstruction at the flux interface:

$$\mathcal{U}_{ij} = \mathcal{U}_i + \tfrac{1}{2}\boldsymbol{G}_i \cdot \left(\boldsymbol{X}_j - \boldsymbol{X}_i\right); \quad \forall j \in \Lambda_i$$

3. Obtain a maximum allowable value of Φ_{ij} each flux interface:

$$\Phi_{ij} = \begin{cases} \min\left(1, \frac{\mathcal{U}_i^{max} - \mathcal{U}_i}{\mathcal{U}_{ij} - \mathcal{U}_i}\right), & \text{if} \quad \mathcal{U}_{ij} - \mathcal{U}_i > 0; \\ \min\left(1, \frac{\mathcal{U}_i^{min} - \mathcal{U}_i}{\mathcal{U}_{ij} - \mathcal{U}_i}\right), & \text{if} \quad \mathcal{U}_{ij} - \mathcal{U}_i < 0; \\ 1, & \text{if} \quad \mathcal{U}_{ij} - \mathcal{U}_i = 0 \end{cases}$$

4. Select the limiter associated to the control volume:

$$\Phi_i = \min_{j \in \Lambda_i} \left(\Phi_{ij}\right)$$

and the positive definite matrices are:

$$\boldsymbol{S}_{ij}^{\boldsymbol{v}} = \frac{\rho_0 c_p}{2}\left(\boldsymbol{n} \otimes \boldsymbol{n}\right) + \frac{\rho_0 c_s}{2}\left(\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n}\right). \tag{4.14}$$

Noting that the nested summation is carried out over control volumes in Equation (4.12) and the anti-symmetric nature of the first line of the right hand side, it can be conclude that these terms cancel and thus Equation (4.12) reduces to:

$$\sum_i \Omega_0^i \frac{d\mathcal{H}_i}{dt} - \dot{\Pi}_{\text{ext}} = \sum_i \sum_{j \in \Lambda_i} \boldsymbol{v}_i \cdot \left(\boldsymbol{S}_{ij}^{\boldsymbol{v}}\left(\boldsymbol{v}_j - \boldsymbol{v}_i\right)\right) + \sum_i \sum_{j \in \Lambda_i} \Sigma_J^i S_{ij}^{\Sigma_J}(\Sigma_J^j - \Sigma_J^i) - \sum_i \dot{D}_i, \tag{4.15a}$$

$$= \sum_i \sum_{j \in \Lambda_i} \boldsymbol{v}_j \cdot \left(\boldsymbol{S}_{ji}^{\boldsymbol{v}}\left(\boldsymbol{v}_i - \boldsymbol{v}_j\right)\right) + \sum_i \sum_{j \in \Lambda_i} \Sigma_J^j S_{ji}^{\Sigma_J}(\Sigma_J^i - \Sigma_J^j) - \sum_i \dot{D}_i. \tag{4.15b}$$

It is worth pointing out that the first two terms on the right hand side can be equivalently written by swapping indices i and j. Simply averaging the first line and the second line of the equation above, and noticing that $\boldsymbol{S}_{ij}^{\boldsymbol{v}} = \boldsymbol{S}_{ji}^{\boldsymbol{v}}$ and $S_{ij}^{\Sigma_J} = S_{ji}^{\Sigma_J}$, an alter-

native expression can be defined as:

$$\sum_i \Omega_0^a \frac{d\mathcal{H}_i}{dt} - \dot{\Pi}_{\text{ext}} = -\frac{1}{2}\sum_i \sum_{j\in\Lambda_i} \left[(\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot \boldsymbol{S}_{ij}^{\boldsymbol{v}} (\boldsymbol{v}_j - \boldsymbol{v}_i) + S_{ij}^{\Sigma_J} \left(\Sigma_J^j - \Sigma_J^i \right)^2 \right] - \sum_i \Omega_0^j \dot{\bar{\varepsilon}}_p^i \bar{\tau}_i. \tag{4.16}$$

It can be observed that, the first term on the right hand side of Equation (4.16) is always non-positive. Moreover, in the case of elasto-plasticity, the second term involving the rate of plastic dissipation is also non-positive, thus satisfying the second law of thermodynamics from the inequality in Equation 3.54.

4.4 Time Discretisation

To achieve a numerical scheme that has approximately second order accuracy, a second order time integrator must be selected to be coupled with the linear reconstruction and Barth-Jesperson limiter implemented for the spatial discretisation. The time integrator implemented to advance the simulation can be one of many well-known methods, to ensure robustness and simplicity Time Variation Diminishing (TVD) Two Stage Runge-Kutta (RK2) scheme was implemented [14]. This scheme consists of two stages and an average step, which can be expressed as follows:

$$\boldsymbol{\mathcal{U}}_i^* = \boldsymbol{\mathcal{U}}_i^n + \Delta t \dot{\boldsymbol{\mathcal{U}}}_i^n \left(\boldsymbol{\mathcal{U}}_i^n, t^n \right), \tag{4.17a}$$

$$\boldsymbol{\mathcal{U}}_i^{**} = \boldsymbol{\mathcal{U}}_i^* + \Delta t \dot{\boldsymbol{\mathcal{U}}}_i^* \left(\boldsymbol{\mathcal{U}}_i^*, t^{n+1} \right), \tag{4.17b}$$

$$\boldsymbol{\mathcal{U}}_i^{n+1} = \frac{1}{2} \left(\boldsymbol{\mathcal{U}}_i^n + \boldsymbol{\mathcal{U}}_i^{**} \right). \tag{4.17c}$$

The time step Δt is computed based on the standard Courant-Friedrichs-Lewy (CFL) condition which ensures stability of the scheme such that:

$$\Delta t = \alpha_{CFL} \frac{h}{U_{p,max}^n}, \tag{4.18}$$

where α_{CFL} is the CFL number, h is the minimum characteristic length of all control volumes and $U_{p,max}^n$ is the maximum shock pressure wave speed at a given time step n. This fully discrete numerical scheme, was first implemented in MATLAB for one and two dimensional scenarios then extended to three-dimensions through a purpose built solver implemented in open-source software "OpenFOAM". An introduction to OpenFOAM and the creation of this solver will be discussed in the next chapter.

Chapter 5

OpenFOAM Computational Framework

5.1 Preliminaries

In this Chapter, the three dimensional implementation of the proposed first order conservation framework for large strain explicit contact dynamics into open source software "OpenFOAM" is presented, first a brief introduction to OpenFOAM is presented in Section 5.2. This is then followed by the introduction and implementation of the multiple body contact solver for the proposed method in Section 5.3, including the purpose built VCFVM library and an introduction to the arbitrary mesh interface library for non-matching interfaces. Lastly, an example workflow of how to setup an example multi-body contact simulation using the custom built solver is explored in Section 5.4 with relevant mesh and input files included for completeness in Appendix B and C.

5.2 Introduction to OpenFOAM

OpenFOAM [120] is a well known cell-centred finite volume package for a variety of academic or industrial applications and is licensed under the General Public License (GPL) which gives anyone the ability to download, install, use and modify the source code of the software package to fit a desired application. OpenFOAM is implemented in C++, an object oriented programming language [144], providing the traditional benefits of abstraction, inheritance, polymorphism and operator overloading thus producing a large and efficient library of classes for the user to take advantage of including complex mathematical and physical models or algorithms.

OpenFOAM has been primarily developed with a focus towards CFD applications, however with the large library of complex mathematical models and algorithms already developed for these applications, the solvers can easily be modified or extended

to include various continuum mechanics problems such as solid dynamics. The standard OpenFOAM distribution is limited to small strain solid mechanic problems with several unofficial branches of OpenFOAM expanding to large strain problems, such as `"foam-extend"`. OpenFOAM is designed to create executables, known as applications, from a vast C++ library which are characterised as either a solver or utility. A solver is designed to solve a desired continuum mechanics problem such as `"icoFoam"` which is designed for incompressible fluid dynamics. While a utility is designed for data manipulation such as `"gmshToFoam"` which is used to convert a mesh file created using mesh generation software, gmsh [145] (*.msh) to the format readable by OpenFOAM. To develop a solver for the proposed method, first the general OpenFOAM problem definition (pre-processing) must be examined prior to concentrating on the definition and solving of the proposed method.

5.2.1 OpenFOAM Problem Definition

To simulate a problem using OpenFOAM, the desired problem must first be defined by creating the test case directory, geometry, initial and boundary conditions along with the mesh for the given problem, this procedure is known as pre-processing. For OpenFOAM simulations, first the test case directory must be created based on the typical OpenFOAM directory structure, as shown in Figure 5.1 for single region and multiple region cases, the selection of which varies depending on the problem scenario and geometry. The typical case directory consists of four main components, first, is the definition of the appropriate geometry and mesh. In general to obtain a tetrahedral mesh[1] , the geometry is defined and meshed using an external Computer Aided Design (CAD) and mesh generation package such as SolidWorks, ANSYS or gmsh before requiring conversion to the OpenFOAM format through an appropriate OpenFOAM utility. Throughout this research, gmsh was used to generate and mesh the geometries hence the `*.msh` file extension is used to define the mesh file and was converted to the OpenFOAM format using the `gmshToFoam` utility, storing the OpenFOAM mesh in sub-directory `polyMesh`. The appropriate boundary conditions can then be defined through the `polyMesh/boundary` file. For simulations where the mesh remains unchanged throughout the simulation, the `polyMesh` sub-directory is stored within, the second main component of the case directory, the `constant` directory. This directory is used to store any data that is constant throughout the simulation, for example the mesh or various continuum properties[2]. The third component is the `system` directory which defines all relevant information to describe and control the simulation. This includes

[1] If requiring a hexahdral mesh, OpenFOAM has a utility for hexahedral mesh generation `snappyHexMesh` designed for predefined geometry files.

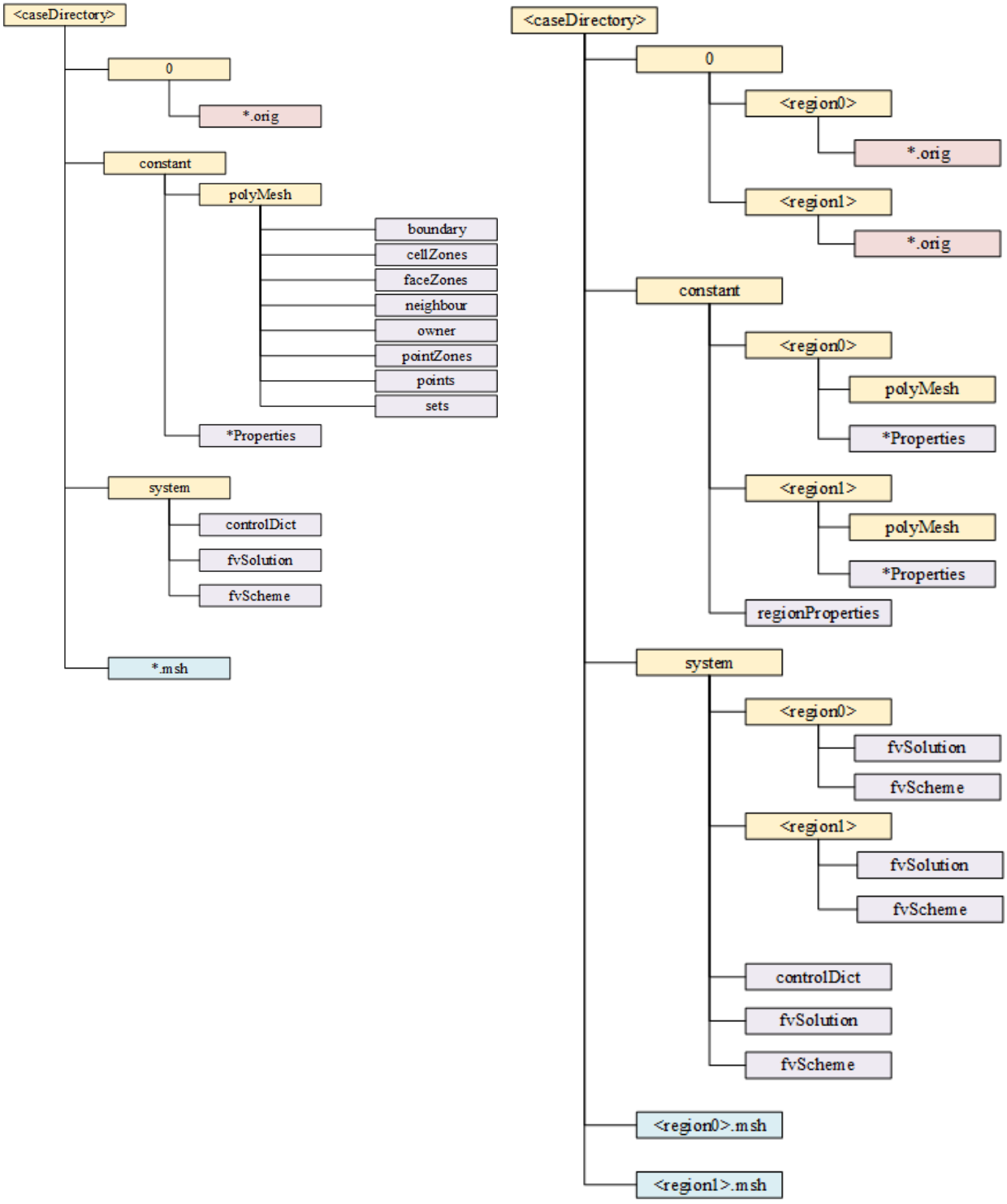
[2] For Example `fluidProperties`, `thermoProperties` or `mechanicalProperties`

the control dictionary, `controlDict`, file which specifies various system parameters required such as the solver name, start time, end time and CFL number. The `fvSchemes` and `fvSolution` are used to control the finite volume scheme with various parameters such as discretisation methods for gradient or divergence equations. The last component is the initial time directory, `0`, containing all initial conditions, such as velocity, required by the solver and is specified within this directory as `*.orig` file extension.

By comparing Figure 5.1a and 5.1b it is clear that both single region and multi-region case structures follow a similar format with two primary differences. First is that the multiple region case structure uses sub-directories for each time (`0`), `constant` and `system` directories in order to define the initial conditions, mesh and system parameters associated with each region. Second, is the addition of the `regionProperties` file which defines the number of regions and region names to be read by the solver.

By following one of these case structures, any type of simulation can be created for an appropriate solver using the existing libraries in OpenFOAM, therefore, to simulate problems for explicit large strain contact dynamics using the proposed method, a purpose-built OpenFOAM solver was created as part of this research. This custom solver has limited OpenFOAM integration through utilisation of the main algorithms, case structure, mathematical and field components of the OpenFOAM C++ library where possible.

The primary challenge associated with development of a vertex centred solver in OpenFOAM is that the base C++ library lacks any template objects or functions for vertex (point) based operations or algorithms. This meant that the purpose built solver developed for this research was developed as a stand-alone solver with custom algorithms, operations and class objects, as required for the application, with OpenFOAM as the foundation. This limits the scalability of the current solver as significant development time would be required to either extend the current solver efficiently as required, or to fully integrate a vertex centred discretisation based solver into OpenFOAM. Therefore, throughout this chapter only snapshots of the code will be provided with appropriate outlining algorithms in order to explain the implementation of the vertex based solver for brevity.



(a) Single Region Case Structure (b) Multiple Region Case Structure

Figure 5.1: Typical OpenFOAM Case Structure

5.3 Contact Dynamic Solver Implementation

Following the typical OpenFOAM case structure in Figure 5.1, this section introduces the implementation in OpenFOAM of the proposed method, by considering each contacting body as an individual region i.e. Figure 5.1b will be the base case structure for this research[3]. First the underlying custom vertex centred C++ library, developed as part of this research, required for the proposed method will be introduced in Section 5.3.1. The explicit multi-body contact solver will then be introduced in Section 5.3.2 with an example workflow, this is then extended to non-matching meshes through the introduction of an OpenFOAM library called Arbitrary Mesh Interface (AMI) in Section 5.3.3.

5.3.1 Custom Vertex Centred OpenFOAM Library

In order to develop a multi-body contact solver based on the the proposed method in OpenFOAM, a series of custom class objects were created for the vertex centred methodology, that will be introduced in this section. A summary of the `vertexCentred` library is shown in Figure 5.2 and is categorised similar to OpenFOAM into directories as applications and libraries.

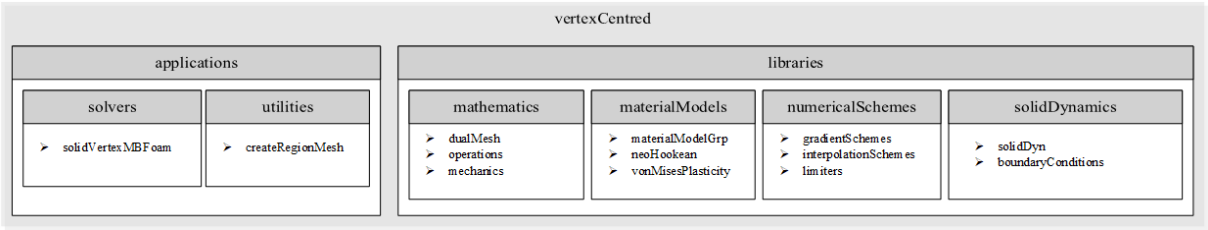


Figure 5.2: Vertex Centred First Order Framework Library Structure

Each directory is described as follows:

- `./applications/solvers` - this directory contains the solver for large strain contact dynamics using the first order framework discretised using VCFVM and will be discussed further in Section 5.3.2.
- `./libraries/mathematics` - consists of the underlying vertex based mathematics and constructions required for this research, including the `dualMesh` construction, mathematical `operations` and continuum `mechanic` computations.

[3]While either case structure could be used as the basis of the multi-body contact solver, the multiple region structure was chosen for ease of implementation but requires more pre-processing.

- `./libraries/materialModels` - this directory contains definitions for all material models including the `runTime` selector, `materialModelGrp`, for the `neoHookean` and `vonMises-Plasticity` material models as described in Section 2.5.
- `./libraries/numericalSchemes` - this directory contains definitions for numerical based schemes such as vertex based `gradientSchemes`, `interpolationSchemes` and `limiters` as described in Section 4.3.2.
- `./libraries/solidDynamics` - consists of the definitions for the vertex based first order framework in `solidDyn` based on Section 4.3 and the definitions for boundary conditions in `boundaryConditions`, importantly both single and multi-body contact algorithms based on the contact equations in Section 3.3.

Since this research is focused on the development of a explicit contact dynamic solver, the underlying class objects from `./libraries` will not be presented, therefore refer to theory presented in Chapters 2, 3 and 4.

5.3.2 Explicit Multi-Body Contact Solver

By utilising the custom VCFVM library introduced in Section 5.3.1 an explicit multi-body contact solver was developed. The solver developed as part of this research is shown in Listing 5.1 with relevant comments to describe the workflow.

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2011-2019 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

Application
    solidVertexMBFoam

Description
    A large strain multi-body contact dynamics solver based on a linear
```

```
    momentum/strains mixed formulation. An explicit Total Lagrangian
    formulation utilising a monolithic Total Variation Diminishing
    Runge-Kutta time integrator.


\*---------------------------------------------------------------------------*/
// Include required header files
#include "fvCFD.H"
#include "pointFields.H"
#include "regionProperties.H"
#include "solidDyn.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[])
{
    #include "setRootCase.H"          // Set path and root case directories
    #include "createTime.H"           // Initialise time variables
    #include "createMultiMesh.H"      // Read regions and generate primary meshes
    #include "readControls.H"         // Read control parameters
    #include "createSolidDynamics.H"  // Generate vertex based regions for solid dynamics

    // Main time loop
    while (runTime.loop())
    {
        // Calculate and set minimum time increment across all bodies
        deltaT.value() = GREAT;
        forAll(bodyRegions, bodyNum)
        {
            solidRegions[bodyNum].updateMaxWaveSpeed();
            deltaT.value() = min(deltaT.value(),(cfl*solidRegions[bodyNum].h_maxUp().value()));
        }
        runTime.setDeltaT(deltaT);

        // Update time and time step
        t += deltaT; tstep++;

        // Print to console current time variables
        Info<< "\nTime Step = " << tstep << "\n deltaT = " << deltaT.value() << " s"
          << "\n Time = " << t.value() << " s" << endl;

        // Set previous time step variables for all bodies
        forAll(bodyRegions, bodyNum)
        {
            solidRegions[bodyNum].setOldTime();
        }

        // Compute Runge-Kutta Stages
        for (int i=0; i<RKstages; i++)
        {
            // Update Boundary and Set half stage variables for all bodies
            forAll(bodyRegions, bodyNum)
            {
                solidRegions[bodyNum].updateBoundaryConditions();
                solidRegions[bodyNum].setHalfStageVariables();
            }


```

```
            // Solve and update stage for all bodies
88          forAll(bodyRegions, bodyNum)
            {
90              // Solve stage
                solidRegions[bodyNum].solveStage(t,deltaT,i);
92
                if (i < RKstages-1)
94              {
                    // Update stage
96                  solidRegions[bodyNum].updateStage(t,deltaT);
                }
98          }
        }
100
      // Advance time for all conserved variables and update variables for all bodies
102     forAll(bodyRegions, bodyNum)
        {
104         solidRegions[bodyNum].advanceTime(deltaT);
            solidRegions[bodyNum].updateStage(t,deltaT);
106         solidRegions[bodyNum].updateMaterialModel();
        }
108

110     // Output time step results for all bodies
        if (runTime.outputTime())
112     {
            forAll(bodyRegions, bodyNum)
114         {
                solidRegions[bodyNum].output();
116         }
        }
118     // Print to console when complete
        Info<< " Simulation completed = "
120       << (t.value()/runTime.endTime().value())*100 << "%" << endl;
    }
122
    // Print to console simulation statistics
124 Info<< "\nExecutionTime = " << runTime.elapsedCpuTime() << " s"
      << "  ClockTime = " << runTime.elapsedClockTime() << " s"
126   << nl << "End\n" << endl;

128 return 0;
}
130
// ************************************************************************* //
```

Listing 5.1: Source Code of Explicit VCFVM Multi-Body Contact Solver

This purpose built solver and VCFVM library is capable of solving explicit contact dynamics problems for one or two bodies with the possibility of extension to three or more bodies. By developing this solver based on the proposed method presented in Chapters 2, 3 and 4, simple conforming contact scenarios can be solved, however an important challenge of three dimensional multi-body contact is when the contact occurs between non-matching contact interfaces. Therefore, to be able to solve non-matching contact

interfaces an additional OpenFOAM library known as Arbitrary Mesh Interface (AMI) was adapted for application with the contact solver.

5.3.3 Arbitrary Mesh Interface

One challenging aspect of multi-body contact is to address when two meshes come into contact with two completely independent geometries and associated meshes, resulting in a non-matching interface i.e. nodes and cells that do not overlap exactly. For the proposed method this proves challenging as the problem variables are stored and calculated at the nodes (vertex) for each mesh, and when two bodies come into contact this information must be passed accurately between the bodies to appropriate nodal positions. For one and two dimensions, this is trivial by carefully considering the problem setup and mesh generation, however for three dimensions the treatment of non-matching meshes is essential as meshes become highly unstructured due to complex geometries.

To tackle this challenge several methods exist as discussed previously for FEM methods or other approaches for example, generalised grid interface [146] for finite volume methods. Since this research is focused on the development of the purpose built OpenFOAM contact solver for the proposed method, assessing the capabilities and implementation of these methods is outside the scope of this thesis, therefore, harnessing the pre-existing OpenFOAM libraries is a priority in this scenario. OpenFOAM has a pre-developed library for multi-region computations including the transfer of problem variables for non-matching meshes through a method known as Arbitrary Mesh Interface (AMI) this library was developed based on the algorithm presented by Farrell and Maddison [147].

This AMI library has been designed and fully parallelised for meshes which overlap such as those for rotating fans in fluid dynamics, sliding interfaces or mapping fields for coupled simulations. This approach works by taking two overlapping meshes and projecting the nodes for one mesh onto the other then identifying the nearest nodes associated with each cell. OpenFOAM then provides several options to calculate the interpolation between the current face and the nearest cell. The determination of which method is best for the application to contact dynamics is outside the scope of this research hence a robust method based on the area of each overlapping face was selected termed "faceAreaWeight" or "partialFaceAreaWeight" in OpenFOAM. These methods would provide a weighting based on the overlapping area of each mesh face for the interpolation of the cell variables from one mesh face to the other.

The AMI library has a number of different methods, functions and boundary condi-

tions for applications to CFD problems, however these are implemented for the transfer of cell-to-cell data due to the cell based nature of OpenFOAM. As a result, to utilise these methods in the newly developed vertex-centred solver, the vertex variables require manipulation so that the pre-existing AMI code can be harnessed. To do this, each face associated with a contact patch would require an associated surface variable stored at the centre of each face, for each variable to be interpolated $\{\boldsymbol{v}, \boldsymbol{t}\}$. These variables would then be interpolated between contact patches using the AMI library then linearly reconstructed from the face centre to the node using a gradient calculated based on surrounding surface variables following a two dimensional variation of the method in Section 4.3.2 without application of a limiter. This procedure for clarity is summarised in Algorithm 5.1 and depicted in Figure 5.3 as a two-dimensional example. Algorithm 5.1 was implemented within the `boundaryConditions/contact` class object and thus would give the multi-body solver the ability to solve problems with both matching and non-matching meshes.

Algorithm 5.1: The non-matching mapping procedure in two dimensions.

1 (1) Obtain averaged variables at face centroid f: $\{\boldsymbol{t}_f^L, \boldsymbol{v}_f^L\} \leftarrow \{\boldsymbol{t}_i^L, \boldsymbol{v}_i^L\}$;

2 (2) Map face variables from "L" to "R" surface using AMI face area projection weighting $\{\boldsymbol{t}_f^{\text{Map}}, \boldsymbol{v}_f^{\text{Map}}\}$;

3 (3) Reconstruct face nodal variables $\{\boldsymbol{t}_{if}^{\text{Map}}, \boldsymbol{v}_{if}^{\text{Map}}\}$ via piecewise linear reconstruction;

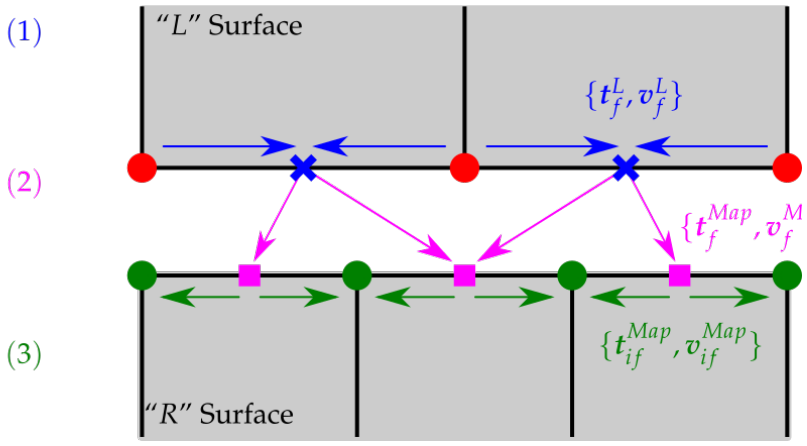


Figure 5.3: Two-dimensional vertex based mapping algorithm to project $\{\boldsymbol{t}_i, \boldsymbol{v}_i\}$ from "L" surface to "R" surface.

5.4 Example Problem Setup and WorkFlow

By implementing the multi-body solver as described in this Chapter, a robust vertex centred solver for large strain contact dynamics based on the first order conservation framework was developed in OpenFOAM. To demonstrate how to this solver can be applied to multi-body contact simulations an example workflow will be demonstrated in this section. The example provided here is the procedure used to generate the OpenFOAM results in Chapter 6 and 7, in particular the one dimensional two identical linear elastic bar impact in Section 6.2. In order to simulate this contact example in OpenFOAM, the traditional CAE simulation workflow is followed as depicted in Figure 5.4 and will be described in the following sections.

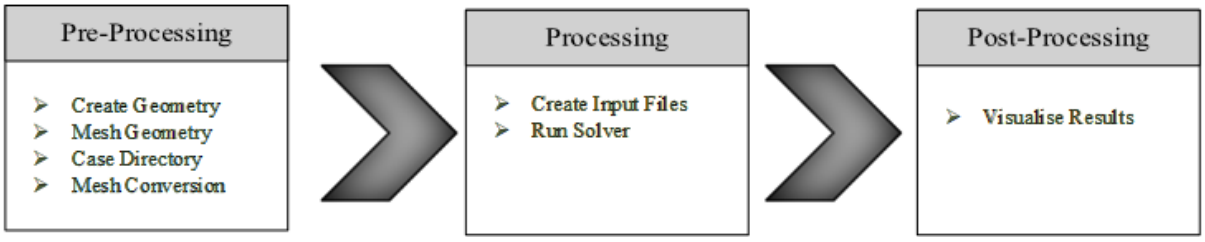
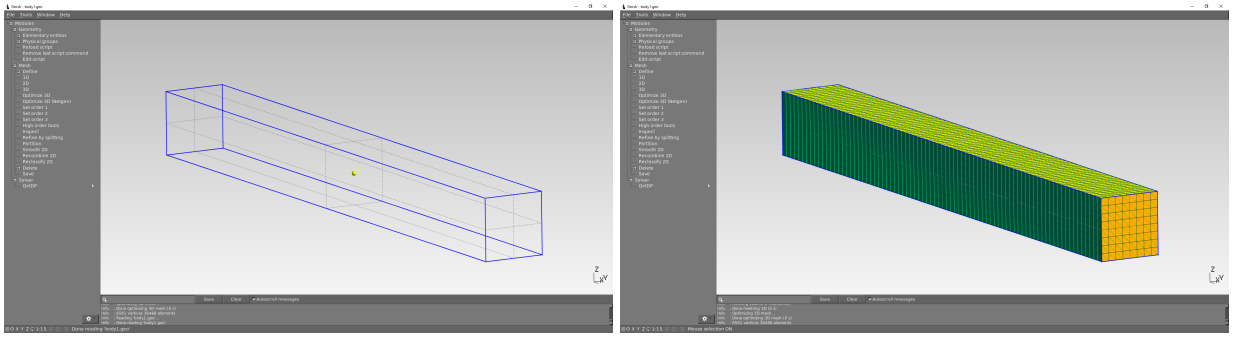


Figure 5.4: Simulation Workflow

5.4.1 Pre-Processing

The first stage is known as pre-processing, where the problem is setup by defining the geometry, generating the mesh, creating the case directory and converting the mesh to the OpenFOAM format. To solve this example using the newly developed OpenFOAM solver the geometry and mesh must first be created. This can be achieved through many different software packages, but for this example the geometry was created and mesh generated in open source software gmsh [145]. To create the geometry in gmsh, first the domain can be created as two geometry scripts based on the two bodies, the example gmsh scripts for each geometry are shown in Listing B.1 for Body 1 and B.2 for Body 2. In these geometry scripts the geometry is first defined with the "box" command, then the domain is discretised using a structured mesh by defining the number of cells along its length, width and height using the "transfinite" command. For this example, a mesh that consisted of 80 cells along the bar length and 8 across the width and height. Last in these scripts, the `"Physical Volume"` name was assigned[4] to correspond with the region name to be assigned in OpenFOAM. The example geometry and mesh are shown in Figure 5.5.

[4]A similar command `"Physical Surface"` can be used to assign the boundary patch names for the boundary conditions at this stage, in this scenario `"free"` and `"contact"`.



(a) Geometry (b) Mesh

Figure 5.5: Example pre-processing of Body 1 in gmsh

With the geometry and mesh parameters now defined, each body can now be discretised using the gmsh mesh generator for 3D geometries, resulting in two meshes that consist of 30720 cells and 6561 nodes that can be saved using the naming convention as body1.msh and body2.msh respectively within the multi-region case directory defined similar to that in Figure 5.1b. The next step is to convert the mesh files into the OpenFOAM format within the case directory, this is easily achieved by using the `"gmshToFoam"` utility with the `"-region"` modifier. In this scenario however, the OpenFOAM mesh conversion was unreliable when converting the boundary conditions if defined using the `"Physical Surface"` command in gmsh[5]. To circumvent this issue, a custom utility script was created and is presented in Listing B.3. This utility converts the mesh files previously created, then redefines the boundary patches by using the `"autoPatch"` and `"createPatch"` utilities, requiring the `"createPatchDict"` files[6] presented in Listings B.4 and B.5 for Body 1 and Body 2 respectively. By running this custom utility the OpenFOAM mesh can be created and boundary conditions correctly applied for rectangular bar impact scenarios only.

5.4.2 Processing

The next part of the typical simulation workflow is known as processing or solving, where the application is run. In order to run the solver, the relevant input files must be defined. First the initial conditions can be set for both bodies by modifying the `"lm.orig"` files, presented in Listing C.1 and C.2, located in the sub-directories of the initial time directory `"0"`. Next the constant directory needs to be setup, first the `"regionProperties"` file must be defined as presented in Listing C.3 in order to define the region names required by the solver and should match the region specific sub-

[5]This issue is associated with the `"transfinite"` commands therefore only occurs for structured meshes using this command.

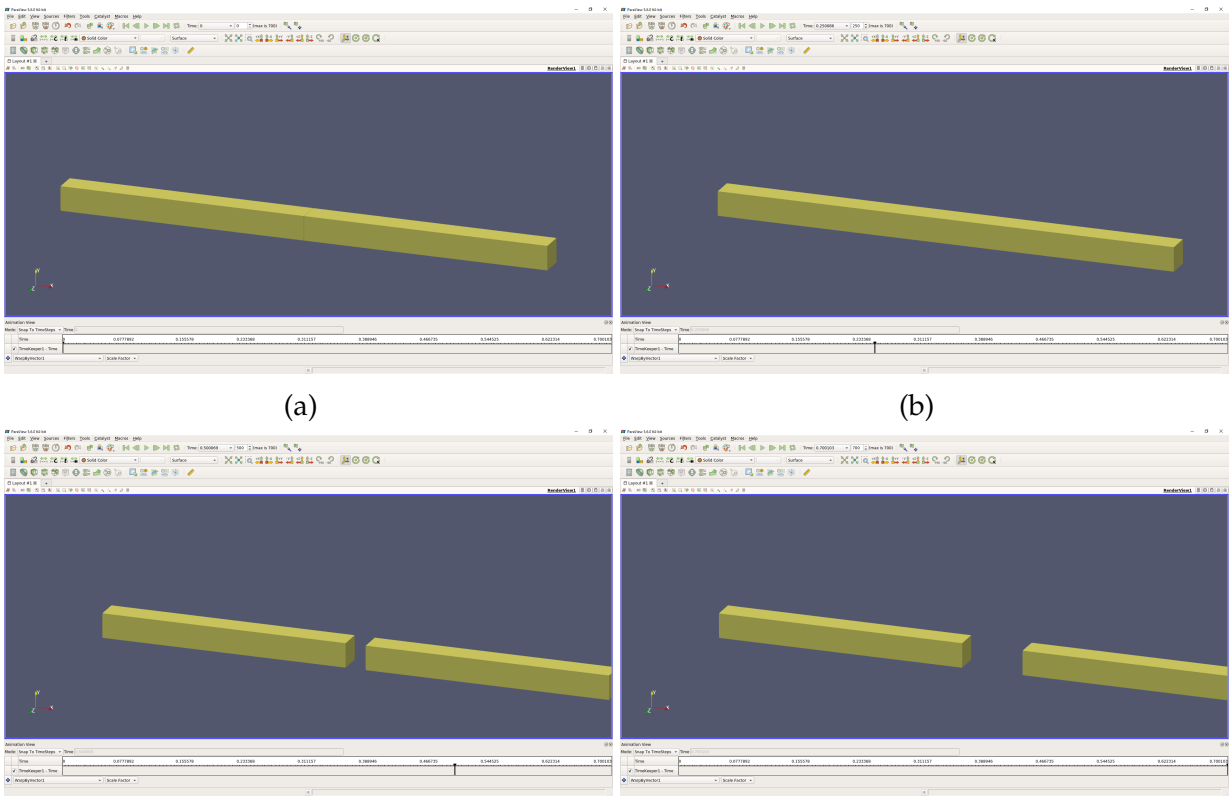
[6]These files must be stored in their region specific sub-directories of the `"system"` directory.

directories used within the case directory. Next the material properties for each region must be defined as presented in Listing C.4. For this scenario, both bodies are identical therefore only one file is presented, however the solver is capable of handling two completely different material models if defined. Lastly, the system directory needs to be setup with the `"controlDict"`, `"fvSolution"` and `"fvSchemes"` presented in Listing C.5, C.6 and C.7 respectively. The `"controlDict"` file defines all the system parameters such as the solver name `"solidVertexMBFoam"`, start time of 0s, end time of 0.7s and CFL number of 0.3. The `"fvSolution"` file must be in each sub-directory and defines the solid dynamic system selections such as solving the additional conservation equations and inclusion of the slope limiter. The `"fvSchemes"` file must be defined in each sub-directory however, this file is not used by the contact solver as it is the file used to select the cell-centred scheme discretisations and is mandatory as part of the standard OpenFOAM case structure. With the case directory now fully defined, as shown in Figure 5.6 the contact solver application can be run.

To run the contact solver, the `"run"` script must be executed through the console from the case directory that was setup during the pre-processing stage. When input, the script listed in Listing C.8 will run and the solver will commence solving the contact scenario previously defined. Various statistics will be displayed in the console and saved in a `"log.solidVertexMBFoam"` for later viewing or debugging. The console will display current progress of the simulation with output results saved in relevant time step directories. Once complete the results can be examined by post-processing.

5.4.3 Post-Processing

With the simulation complete the next stage is to visualise the results known as post-processing. In this research, paraView was used to visualise the results from OpenFOAM by using the paraFoam utility which converts the OpenFOAM format results to a readable format for paraView, an example of the results is shown in Figure 5.7. The local time evolution of the contact nodes were then extracted and plotted in MATLAB for comparison purposes with the one and two dimensional MATLAB codes. The results from this example are shown in Chapter 6.



(a) (b)

(c) (d)

Figure 5.7: Example post-processing of simulation in paraView (a) at $t = 0$s (b) at $t = 0.25$s (c) at $t = 0.5$s (d) at $t = 0.7$s

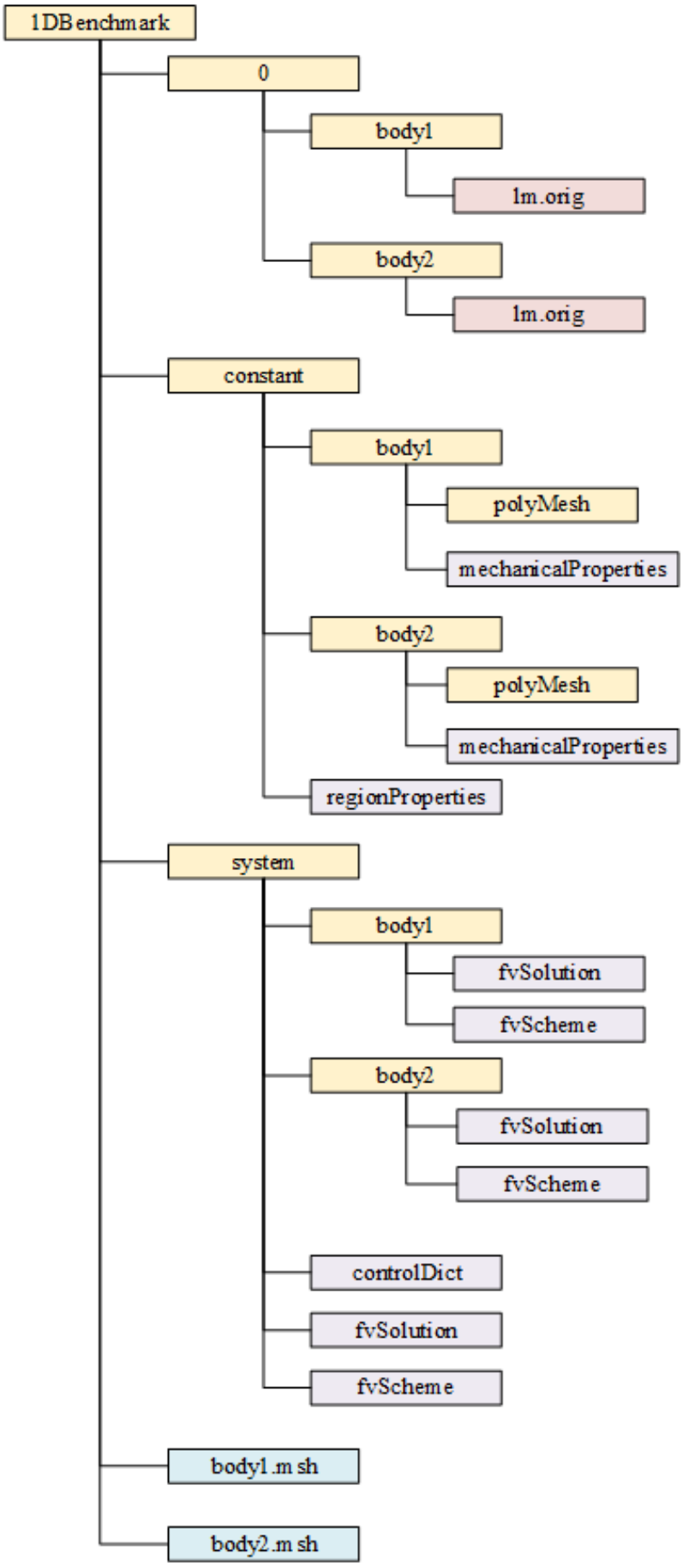


Figure 5.6: Case Directory Example

Part IV

Numerical Results

Chapter 6

One-Dimensional Problems

6.1 Preliminaries

In this chapter, a series of one-dimensional benchmark problems are presented in order to establish the accuracy, consistency and stability of the proposed method. This is achieved through comparisons of the exact solutions and commercial software for four example problems. The first problem considered is that of the impact between two identical linear elastic bars, in Section 6.2, followed by the second problem considering the impact of two dissimilar linear elastic bars, in Section 6.3. These two examples are based on typical manufactured problems for dynamic contact numerical methods as presented in several journal articles [45, 51, 148–154] with simple to derive exact solutions as presented previously in Section 3.4. The objective of these two examples are to benchmark the accuracy and consistency of the proposed method across one, two and three dimensional implementations within MATLAB and OpenFOAM. Next in Section 6.4 the Hugoniot shock model is first benchmarked against Abaqus/Explicit [16] followed by two parametric studies to investigate the importance of using the Hugoniot shock model by examining the slope s and impact velocity. Lastly, in Section 6.5 the impact of two dissimilar Hugoniot bars is investigated and benchmarked against Abaqus/Explicit.

6.2 Two Identical Linear Elastic Bars

6.2.1 Problem Description

The first problem considered, consisted of two one-dimensional linear elastic bars with identical lengths $L = 10$m, as depicted in Figure 6.1. Bar One (left) was travelling at

0.1m/s towards Bar Two (right) which was at rest with an initial gap between the bars δ_n of 0.01m. The material properties for both bars are shown in Table 6.1. The objective of this problem was to benchmark the accuracy of the proposed method against the exact solution for the problem which was derived in Section 3.4 and results in Table 6.2. First the one-dimensional implementation will be investigated, followed by the multi-dimensional implementations of the proposed method.

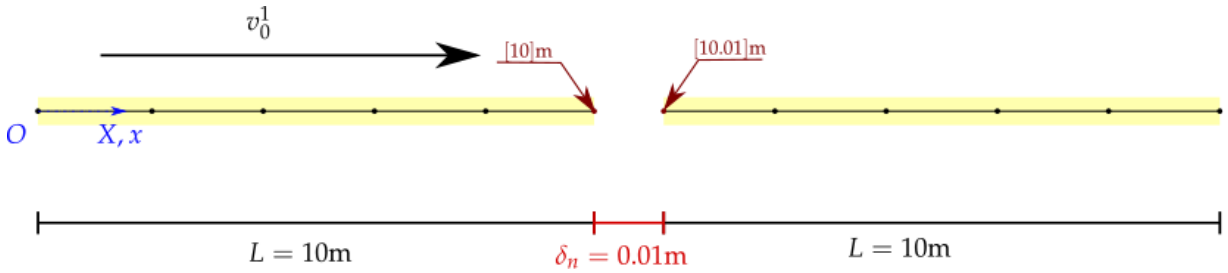


Figure 6.1: Two Identical Linear Elastic Bars - Problem description and geometry for Bar One (left) and Bar Two (right)

Table 6.1: Two Identical Linear Elastic Bars - Material parameters

Young's modulus	E_0	100	$[\mathrm{N/m^2}]$
Material density	ρ_0	0.01	$[\mathrm{kg/m^3}]$
Poisson's ratio	ν	0.0	
Lamé parameters	μ	50	$[\mathrm{N/m^2}]$
	λ	0	$[\mathrm{N/m^2}]$
Acoustic wave speed	c_p	100	$[\mathrm{m/s}]$

Table 6.2: Two Identical Linear Elastic Bars - Exact solution

Stage	Bar One		Bar Two		Time
	v_x^L [m/s]	P_{xX}^L [Pa]	v_x^R [m/s]	P_{xX}^R [Pa]	[s]
Prior to contact	0.1	0	0	0	$0 \leq 0.1$
In contact	0.05	-0.05	0.05	-0.05	$0.1 \leq 0.3$
After contact	0	0	0.1	0	> 0.3

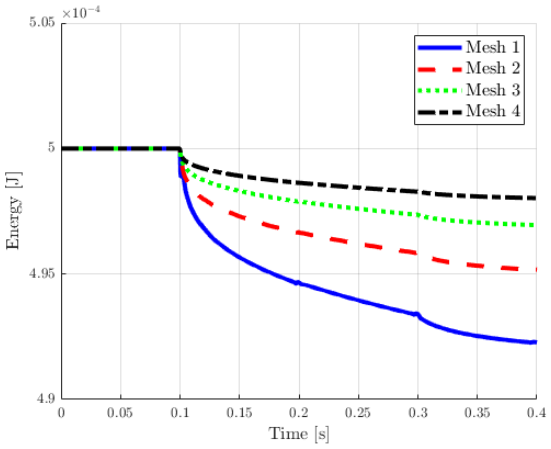
6.2.2 One-Dimensional Results

The proposed method was first implemented as a one-dimensional algorithm in MATLAB with results presented in this section. The problem was simulated using a CFL of 0.3 and in order to investigate the consistency of the method, spatially discretised

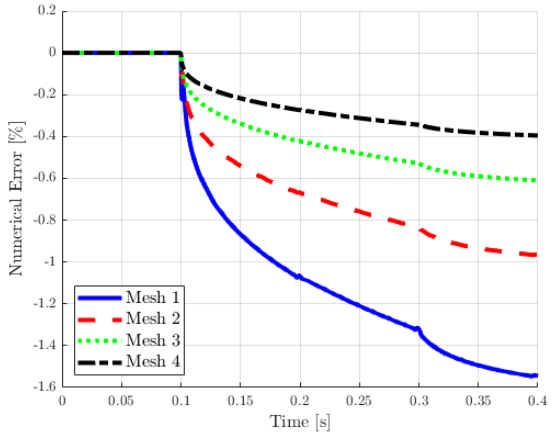
using the following meshes per bar: Mesh 1 consisting of 129 nodes and 128 elements; Mesh 2 consisting of 257 nodes and 256 elements; Mesh 3 consisting of 513 nodes and 512 elements; Mesh 4 consisting of 1025 nodes and 1024 elements.

The first objective to investigate, was the consistency of the proposed method and ensure that the method satisfies the second law of thermodynamics by comparing the global energy time histories for meshes 1 to 4 as shown in Figure 6.2. The consistency of the proposed method can be observed in Figure 6.2a and 6.2b where as the mesh density increases the numerical entropy (error) decreases and the energy conserved after contact ($t \geq 0.1$s) is improved. This demonstrates the satisfaction of the second law of thermodynamics ensuring long term stability. By examining the energy time histories in Figure 6.2c for Bar One and in Figure 6.2d for Bar Two, the expected bar behaviour from Section 3.4 can clearly be observed in terms of energy and can be described as follows. At $t = 0$, Bar One has purely kinetic energy as the bar is travelling with an initial velocity of 0.1m/s, while Bar Two has no energy associated with the bar, as the bar was at rest. As time evolves Bar One translates towards Bar Two until $t = 0.1$s when the bars come into contact and the pressure wave begins to propagate away from the contact region. While in contact, the pressure wave propagates along both bars producing elastic deformation as the kinetic energy is partial converted to potential energy while the remaining kinetic energy from Bar One is transferred to Bar Two as the bars travel together. At $t = 0.2$s the bars potential energy peaks as a result of the pressure wave of each bar reaching the free end and reflecting back to the contact region which results in contact separation occurring at $t = 0.3$s. After separation, it is clear that nearly all kinetic energy is transferred from Bar One to Bar Two, with approximately 0.4% (see Figure 6.2b) dissipated as numerical error for Mesh 4 and could be reduced with further mesh refinement. As a result Bar One is now at rest while Bar Two is travelling at approximately 0.1m/s.

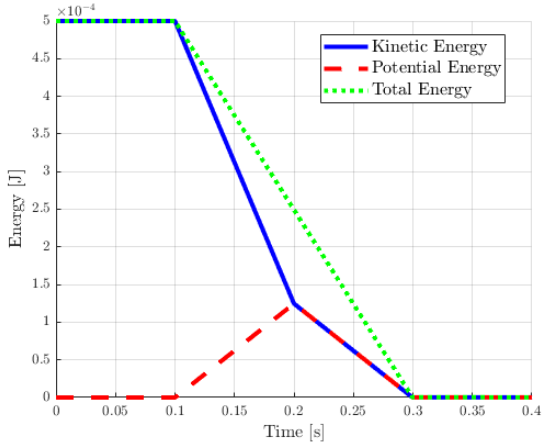
With the consistency of the proposed method clearly demonstrated for the one-dimensional problem, the next objective was to investigate the accuracy of the proposed method by comparing the results against the exact solution. Prior to comparing the results against the exact solution, a series of simulations were conducted to compare the results of different spatial approximations, these results are shown in Figure 6.3 comparing the proposed method using first order (constant), second order (linear) without limiter and second order (linear) with limiter against the exact solution and linear FEM without bulk viscosity. By observing Figure 6.3a and 6.3b it is clear that the linear FEM without bulk viscosity produces an accurate result in contact for the velocity time evolution with oscillations only at separation while in the axial stress time evolution oscillations are observed during contact and at separation demonstrating the need for ad-hoc reg-



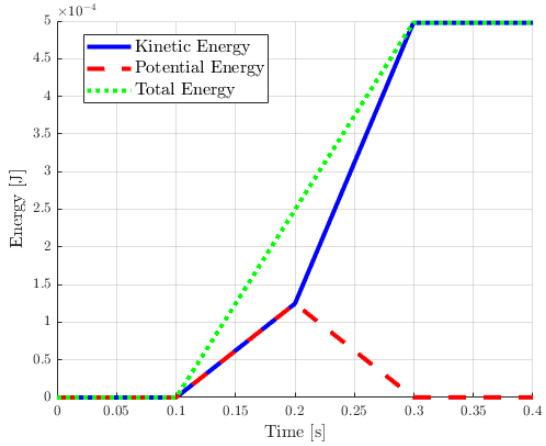
(a) Global total Energy - Mesh refinement



(b) Numerical dissipation - Mesh refinement



(c) Energy measures - Bar One (Mesh 4)

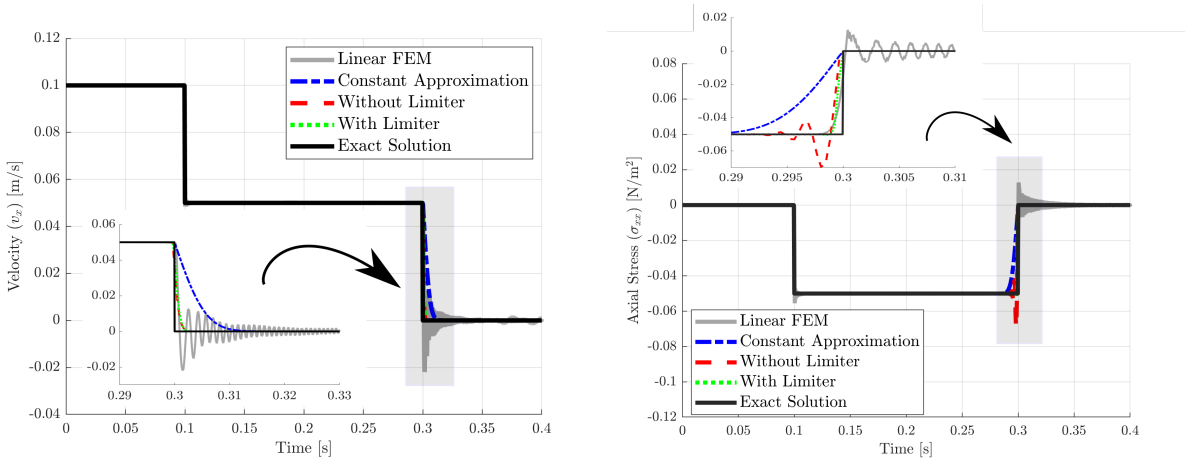


(d) Energy measures - Bar Two (Mesh 4)

Figure 6.2: Two Identical Linear Elastic Bars - Time evolution of global energy and dissipation

ularisation. The proposed method on the other hand agrees well with the exact solution across each spatial approximation, however there are slight discrepancies for each method. For the constant approximation, the local solution is smooth throughout the simulation for both velocity and axial stress time evolutions but is overly dissipative at separation in both results. This would be expected to improve for increasingly fine meshes, however this would also increase the computational cost of the method. While the linear approximation without a slope limiter, accurately matches the exact solution for velocity, however for the stress time history there is spurious oscillations in the region prior to separation. This could be an issue for more complex problems as this could not only affect the accuracy of the solution but the stability of the numerical simulation. Lastly the linear approximation with a slope limiter, accurately matches the exact solution for both velocity and axial stress. This method successfully retains the accuracy of the linear approximation while removing the spurious oscillations as a result this interpolation method will be used throughout this research for all numerical

examples using the proposed method. The investigation and analysis of appropriate limiters is out side the scope of this research. As discussed in Chapter 4, it is expected that drawing on the vast area of research into VCFVM slope limiters, any appropriate limiter could easily be implemented into the proposed method to improve the accuracy of the solutions presented here.



(a) Velocity v_x (b) Axial Stress σ_{xx}

Figure 6.3: Two Identical Linear Elastic Bars - Local time evolution at initial position $X = [10]$m comparing spatial approximations: linear FEM without bulk viscosity, constant (first order), linear (second order) without and with a limiter and the exact solution

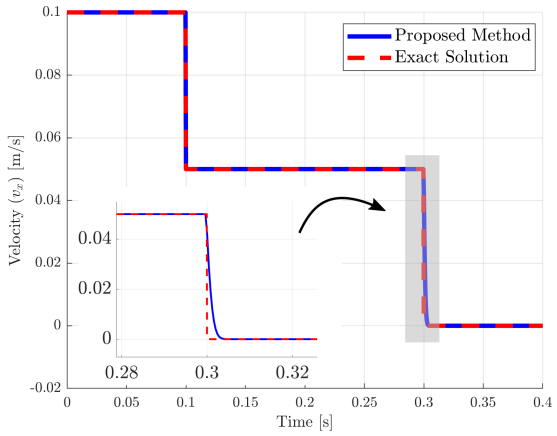
The last results presented for the identical bar scenario are the local time history of the contact nodes of both Bar One and Bar Two at initial positions $X = [10]$m and $X = [10.01]$m respectively, spatially discritised using Mesh 4 and solved by using linear approximation with a slope limiter. The results were then plotted against the exact solution in Figure 6.4. The proposed method matches closely with the exact solution with only a slight overshoot at separation for in the axial stress time evolution.

6.2.3 Multi-Dimensional Results

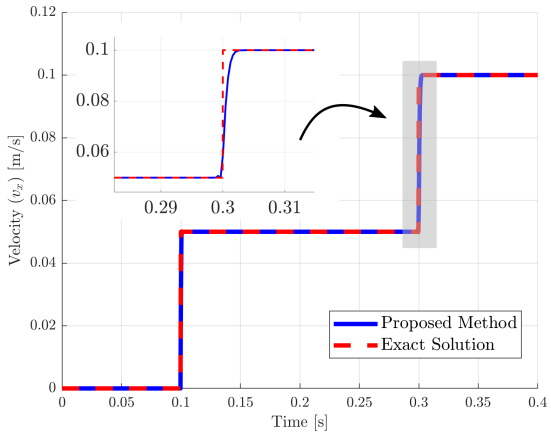
With the promising one-dimensional results presented in the previous section, the next objective was to investigate the two and three dimensional implementations of the proposed method for the same problem setup. The comparison of these implementations with the exact solution are shown in Figure 6.5 for the central contact node for each contact surface corresponding to initial position $\boldsymbol{X} = [10, 0, 0]^T$m and $\boldsymbol{X} = [10.01, 0, 0]^T$m for each bar respectively. The meshes used for each implementation are based on Mesh 4 of the one-dimensional implementation, by using the same number of elements along the length of the bar, 1024, and 2 elements across the height and width to produce a central contact node. The mesh information for each are as follows: in two-dimensions, a

triangular mesh consisting of 3075 nodes and 4096 elements per bar was used and in three-dimensions a tetrahedral mesh consisting of 9225 nodes and 24576 elements per bar.

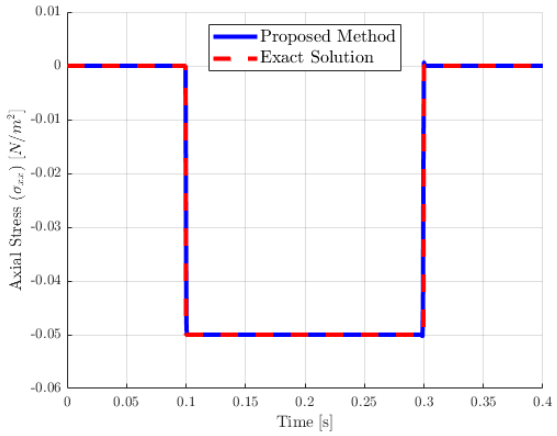
By analysing the local evolutions in Figure 6.5 it can clearly be observed that the proposed method across each implementation produces an accurate result matching closely with the exact solution. In terms of the two-dimensional results in MATLAB, the results match the exact solution perfectly for displacement and velocity with only a slight overshoot at separation in the axial stress evolution. For the OpenFOAM (three-dimensional) implementation the results match the exact solution perfectly for both displacement and velocity while the axial stress matches nearly perfectly with only a tiny overshoot at separation. The results from the multi-dimensional implementations successfully capture the contact accurately with nearly exact replication of the solution across each implementation.



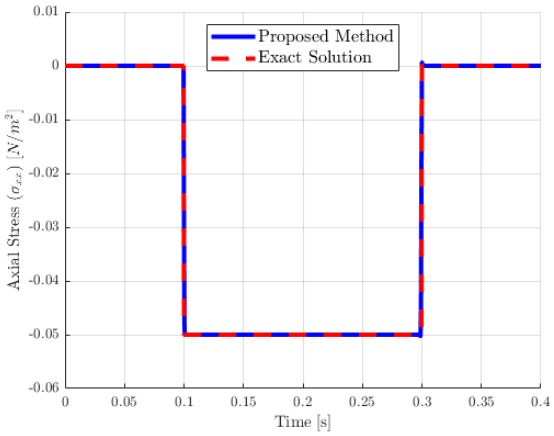
(a) Velocity v_x - Bar One



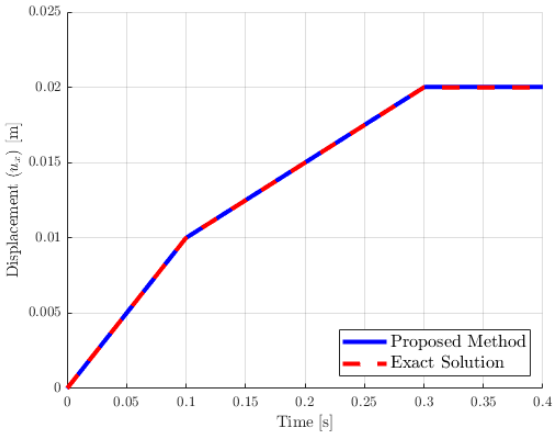
(b) Velocity v_x - Bar Two



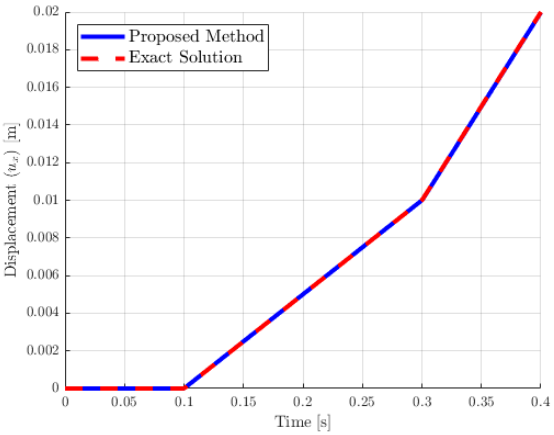
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two

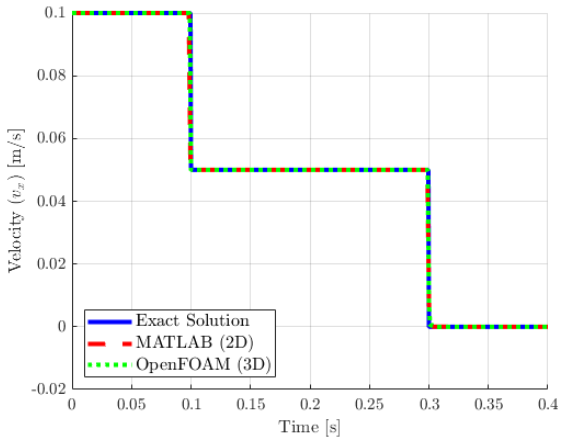


(e) Displacement u_x - Bar One

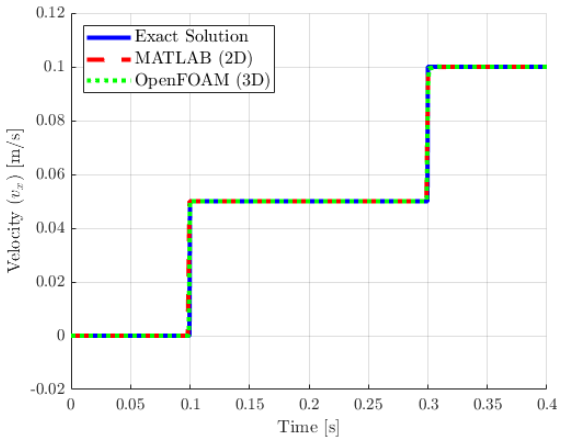


(f) Displacement u_x - Bar Two

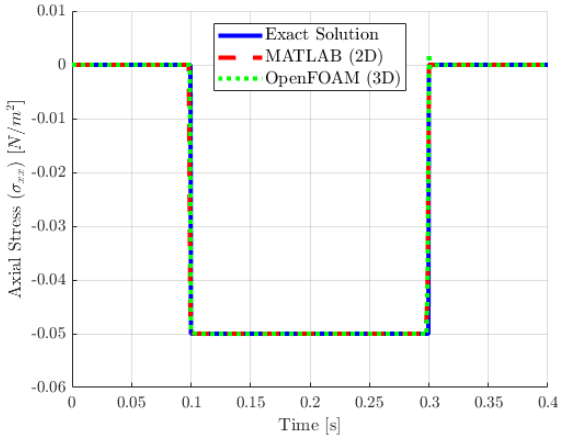
Figure 6.4: Two Identical Linear Elastic Bars - Comparison of local time evolutions for the proposed method and exact solution



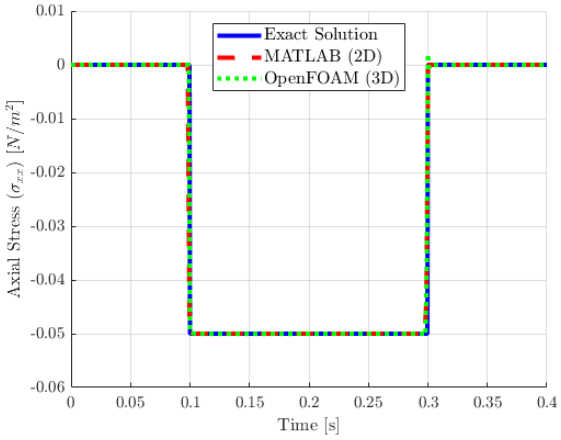
(a) Velocity v_x - Bar One



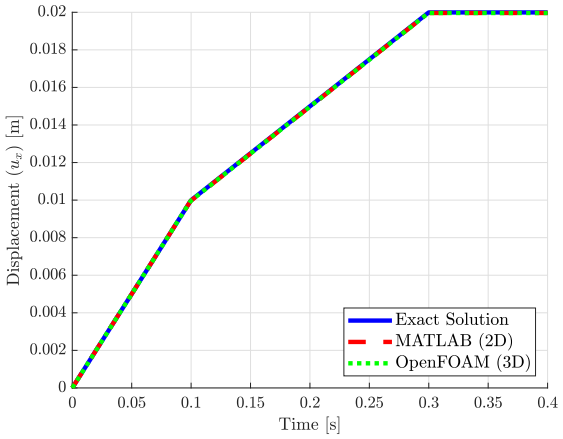
(b) Velocity v_x: Bar Two



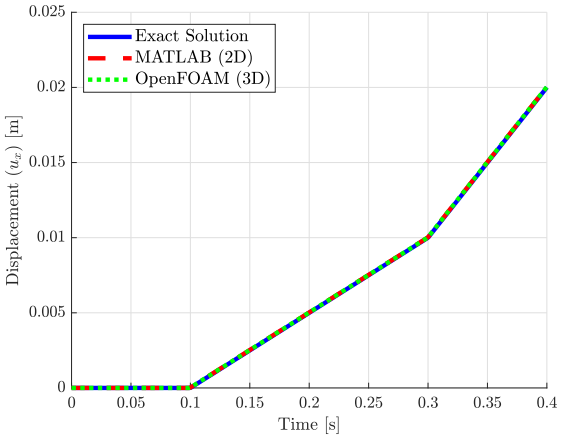
(c) Axial Stress (σ_{xx}) - Bar One



(d) Axial Stress σ_{xx} - Bar Two



(e) Displacement u_x - Bar One



(f) Displacement u_x - Bar Two

Figure 6.5: Two Identical Linear Elastic Bars - Comparison of local time evolution for the proposed method in MATLAB (Two-dimensional) and OpenFOAM (Three-dimensional) with the exact solution at initial position $\boldsymbol{X} = [10, 0, 0]^T$m and $\boldsymbol{X} = [10.01, 0, 0]^T$m for each bar respectively

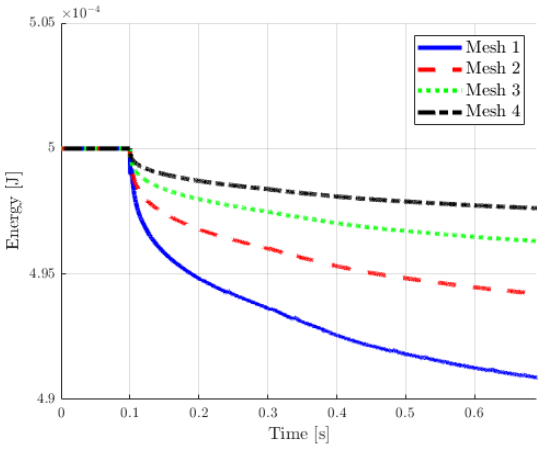
6.3 Two Dissimilar Linear Elastic Bars

The next one-dimensional problem considered consisted of a similar set up to the two identical linear elastic bar problem presented in the previous section (refer to Figure 6.1) with the exception that Bar One now is made from a softer material with a Young's Modulus of 49N/m^2 therefore the pressure wave speed $c_p = 70\text{m/s}$. The purpose of this problem was to investigate the proposed methods robustness when considering a contact interface consisting of two different materials.

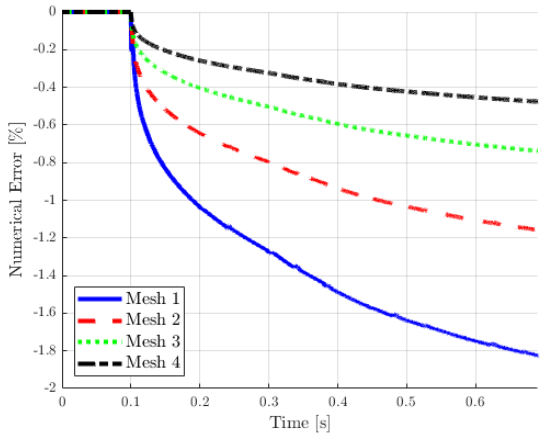
6.3.1 One-Dimensional Results

The first objective, once again, was to investigate the consistency of the proposed method through the mesh refinement of the energy time evolution as shown in Figure 6.6 using the same meshes as described in Section 6.2. This is then followed by the local time evolution of the contact nodes of both Bar One and Bar Two at initial positions $X = [10]\text{m}$ and $X = [10.01]\text{m}$ respectively.

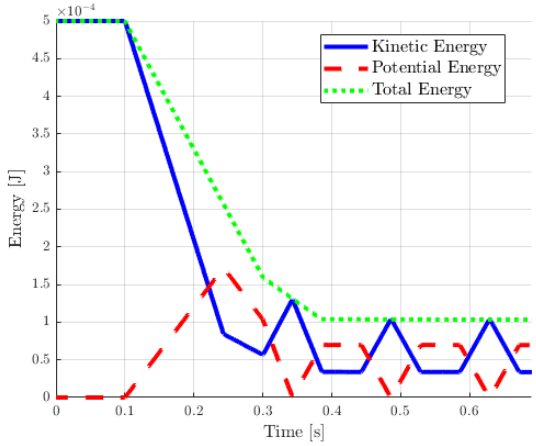
By observing Figure 6.6a and 6.6b it is clear that as the mesh density increases the accuracy of the proposed method improves similar to the results for the identical bar scenario as expected. The energy plots for each bar, Figure 6.6c and 6.6d highlight the importance of accurately modelling a contact interface for dissimilar materials as there is a significant difference in energy response in comparison to those presented in Figure 6.2 for the one-dimensional identical bars scenario, due to the change in material properties. This is evident in Figure 6.6c which shows the energy response for Bar One with the new Young's modulus of 49N/m^2. Until contact at $t = 0.1\text{s}$, Bar One has purely kinetic energy as the bar translates towards Bar Two which has no energy, as the bar is at rest. After contact the kinetic energy from Bar One is transferred to Bar Two as kinetic and potential energy in a similar fashion to that observed in Figure 6.2 as the material properties of Bar Two are unchanged. Bar One on the other hand has a larger proportion of kinetic energy converted to potential energy as the pressure wave propagates through the bar due to the lower Young's modulus. As a result the pressure wave speed is slower (70m/s) in Bar One than Bar Two (100m/s) which results in the pressure waves returning to the contact interface at different times, for Bar Two the pressure wave returns at $t = 0.3\text{s}$ while for Bar One the pressure wave returns at $t \approx 0.39\text{s}$. This can be observed clearly in the energy plots, first in Figure 6.6c at $t = 0.3\text{s}$ there is a sudden increase in kinetic energy as the pressure wave of Bar Two transfers energy across the contact interface while in Figure 6.6d a corresponding decrease in the gradient of the kinetic energy profile until the pressure wave of Bar One returns to the contact interface at $t \approx 0.39\text{s}$ resulting in separation of both bars. Unlike the case for



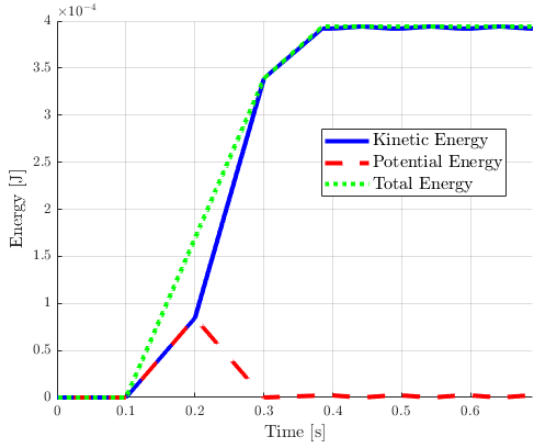
(a) Global total energy - Mesh refinement



(b) Numerical dissipation - Mesh refinement



(c) Energy measures - Bar One (Mesh 4)



(d) Energy measures - Bar Two (Mesh 4)

Figure 6.6: Two Dissimilar Linear Elastic Bars - Energy time evolution of comparing Mesh 1-4 for the proposed method

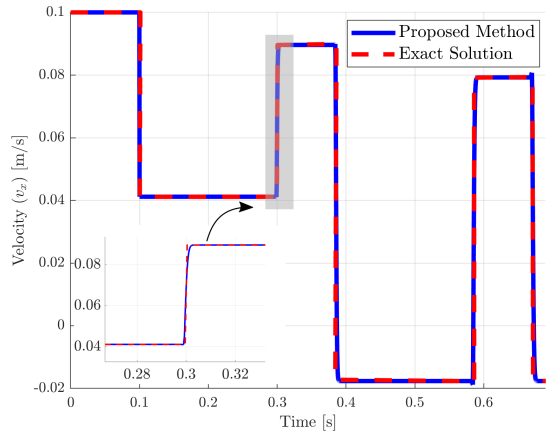
identical bars, not all energy is transferred from Bar One to Bar Two during contact, instead after separation Bar One retains a portion of both kinetic and potential energy which alternately oscillates as Bar One expands and contracts.

These observations are made even more apparent when investigating the local time evolution of the contacting nodes at the interface for velocity, axial stress and displacement in Figure 6.7 when compared with the exact solution. When comparing the results in Figure 6.7 it can be observed that the proposed method matched exactly with the displacement of both bars with near perfect agreement in both the velocity and axial stress results.

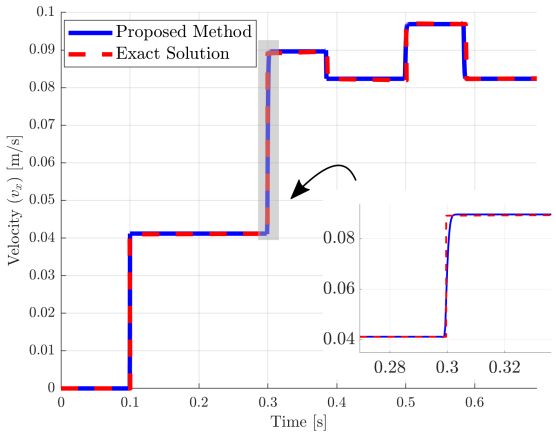
6.3.2 Multi-Dimensional Results

To further benchmark the multi-dimensional implementations, this problem was repeated and compared across each implementation in a similar manner to the identical

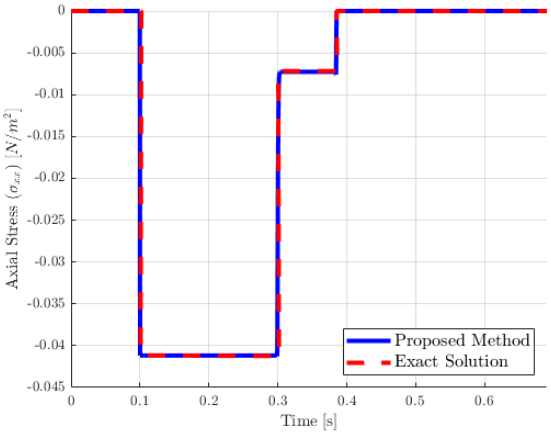
linear elastic bar example in Section 6.2.3. The local results taken at the contact points (initial position $\boldsymbol{X} = [10, 0, 0]^T$m and $\boldsymbol{X} = [10.01, 0, 0]^T$m for each bar respectively) from each set of results using the two and three-dimensional implementations are compared in Figure 6.8. From these results it is clear that the proposed method matches exactly across each implementation for both displacement and axial stress throughout the simulation. In the velocity time evolution however, it can be observed that the solution matches exactly throughout contact but there are overshoots at separation and during post-separation expansion and contraction of the softer material (Bar One) in Figure 6.8a. These overshoots are likely a result of the involutions that must be satisfied for the conservation of deformation gradient $\boldsymbol{F}$ and co-factor $\boldsymbol{H}$. In one-dimension these involutions are naturally satisfied, while in two and three dimensions they require special consideration by the numerical scheme as discussed in Section 4.3.1. This requires further investigation to improve the accuracy of the scheme for dissimilar materials.



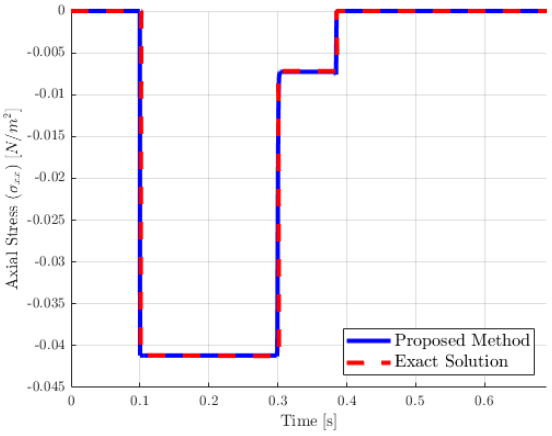
(a) Velocity v_x - Bar One



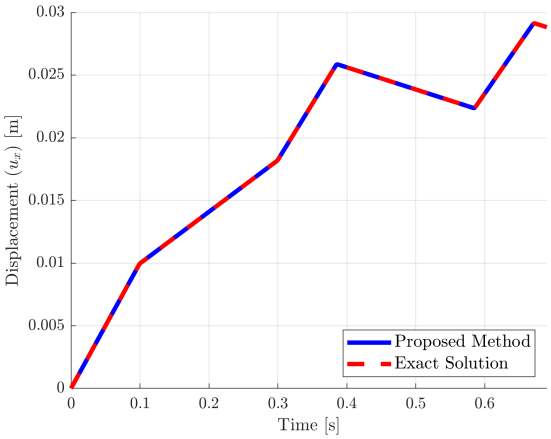
(b) Velocity v_x - Bar Two



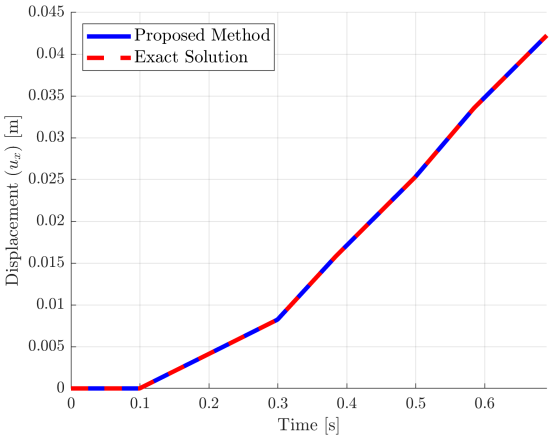
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two

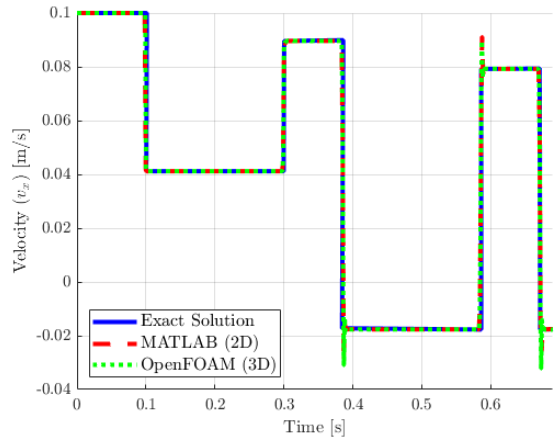


(e) Displacement u_x - Bar One

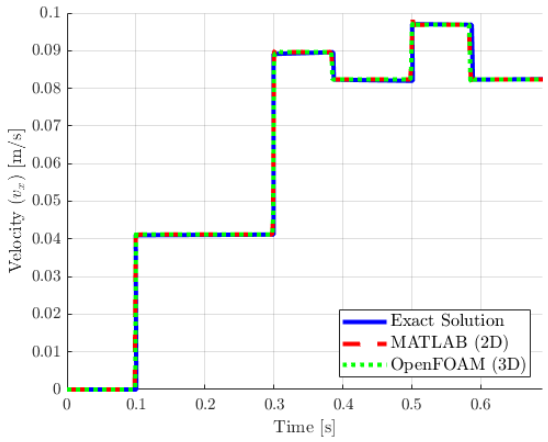


(f) Displacement u_x - Bar Two

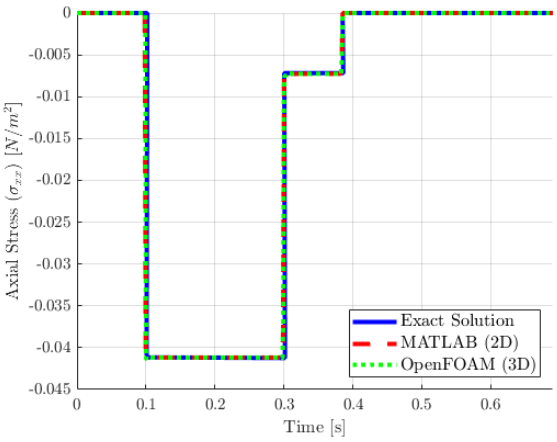
Figure 6.7: Two Dissimilar Linear Elastic Bars - Comparison of local time evolutions for the proposed method and exact solution at initial position $X = [10]$m and $X = [10.01]$m for each bar respectively



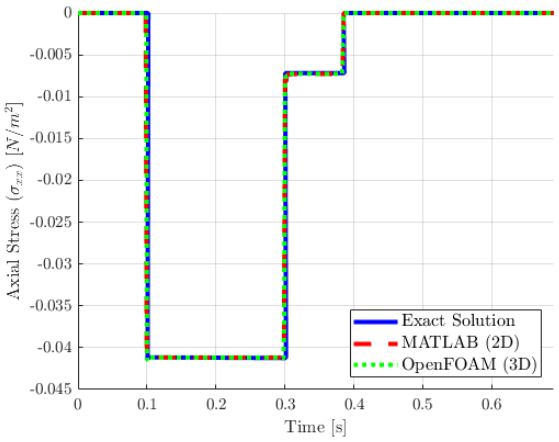
(a) Velocity v_x - Bar One



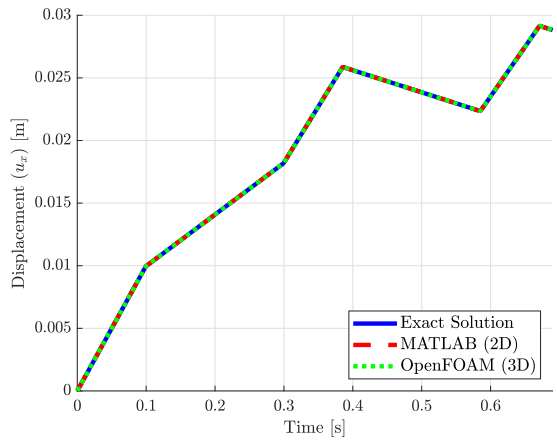
(b) Velocity v_x - Bar Two



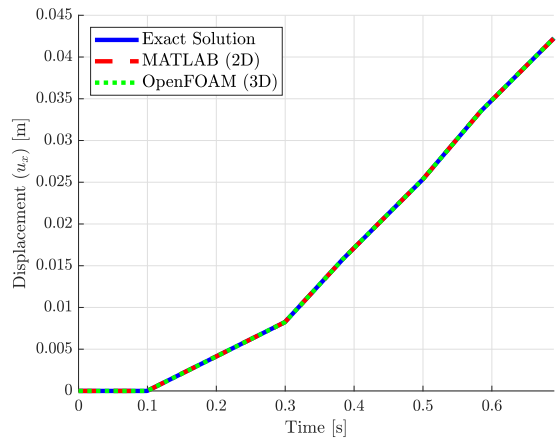
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two



(e) Displacement u_x - Bar One



(f) Displacement u_x - Bar Two

Figure 6.8: Two Dissimilar Linear Elastic Bars - Comparison of local time evolution for the proposed method in MATLAB (Two-dimensional) and OpenFOAM (Three-dimensional) with the exact solution at initial position $\boldsymbol{X} = [10, 0, 0]^T$m and $\boldsymbol{X} = [10.01, 0, 0]^T$m for each bar respectively

6.4 Two Identical Hugoniot Bars

6.4.1 Problem Description

The next problem investigated considers the impact between two one-dimensional bars similar to the benchmark case from Section 6.2, now modelled with Hugoniot pressure constitutive model, refer to Section 2.6.3. The problem set-up for this test case is shown in Figure 6.9 with material properties given in Table 6.3 where the initial velocity of Bar One v_0^1 is significantly increased from 0.1 m/s to 100 m/s (to match the acoustic material wave speed) and initial gap δ_n is increased from 0.01m to 10m to ensure contact occurs at 0.1s. The purpose of this test case is to 1) benchmark the Hugoniot pressure constitutive model against standard FEM approaches, 2) investigate and compare the acoustic Riemann solver and non-linear Riemann solver when using the Hugoniot pressure model, 3) investigate and compare the effect of the slope s of the shock wave speed to particle wave speed ratio for a selection of values $s = \{0, 0.25, 0.5, 0.75, 1\}$, 4) investigate and compare the effect of impact velocity for a selection of initial velocities $v_0^1 = \{0.1, 1, 10, 100\}$. All simulations were conducted with a CFL of 0.3.

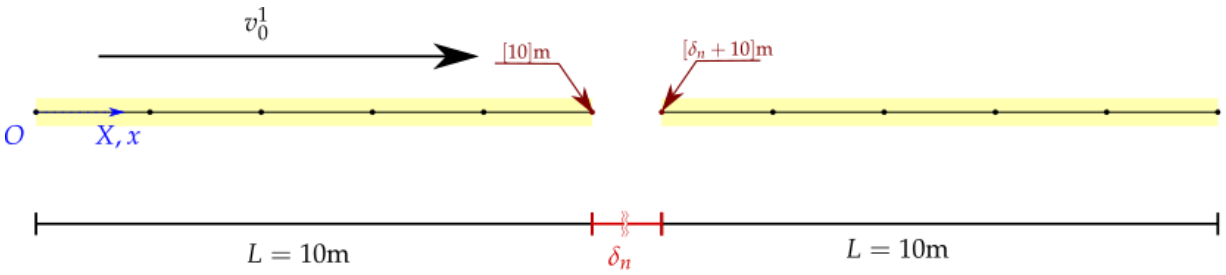


Figure 6.9: Two Identical Hugoniot Bars - Problem description and geometry for bar one (left) and bar two (right)

Table 6.3: Two Identical Hugoniot Bars - Material parameters

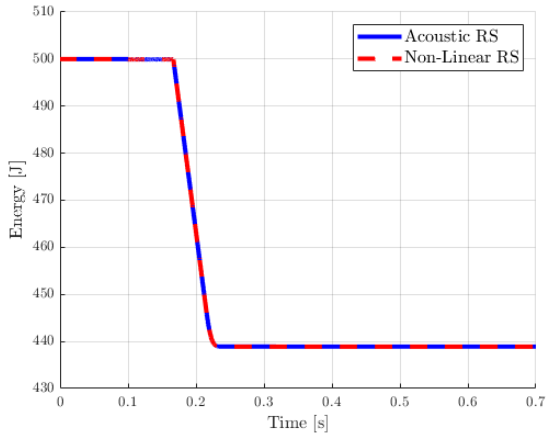
Young's modulus	E_0	100	$[\mathrm{N/m^2}]$
Material density	ρ_0	0.01	$[\mathrm{kg/m^3}]$
Poisson's ratio	ν	0.0	
Lamé parameters	μ	50	$[\mathrm{N/m^2}]$
	λ	0	$[\mathrm{N/m^2}]$
Acoustic wave speed	c_p	100	[m/s]
Shock Wave speed slope	s	$\{0, 0.25, 0.5, 0.75, 1\}$	

6.4.2 Numerical Results

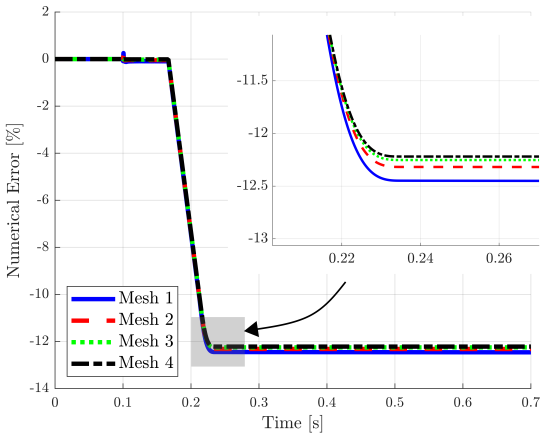
Benchmarking - Comparison with FEM

The first objective was to benchmark the solution against traditional FEMs for a large relative impact velocity, 100m/s, and $s = 1$. To do so the first step is to examine the consistency and accuracy of the global energy time histories by comparing two different methodologies for the proposed method. The two approaches considered here will be the acoustic Riemann Solver (RS) and non-linear Riemann solver, both modelled using the Hugoniot constitutive model. The acoustic Riemann solver will consider the proposed method applied with the acoustic Riemann solver (from Section 3.3.1) throughout the simulation for the calculation for the numerical fluxes. While the non-linear Riemann solver method will consider the proposed method applied with the non-linear Riemann solver (from Section 3.3.2) throughout the simulation for the calculation of the numerical fluxes. The global energy time histories comparing the two methods are shown in Figure 6.10a, when considering 1024 elements across the length of each bar (Mesh 4 from Section 6.2). It can be observed that the two different approaches agree very well, suggesting the different Riemann solvers have little influence on the overall solution in this scenario. While to prove consistency of the proposed method, the numerical dissipation for four meshes $\{128, 256, 512, 1024\}$ elements per bar are compared in Figure 6.10b which shows as number of elements increase the numerical dissipation decreases and tends to the steady state dissipation. Lastly, various energy measures are compared in Figure 6.10c and 6.10d for Bar One and Two respectively for Mesh 4. The kinetic energy from Bar One can clearly be observed to transfer to Bar Two after contact initiation at $t = 0.1$s through kinetic and potential energy with the energy loss equating to the numerical dissipation introduced to the system caused by contact.

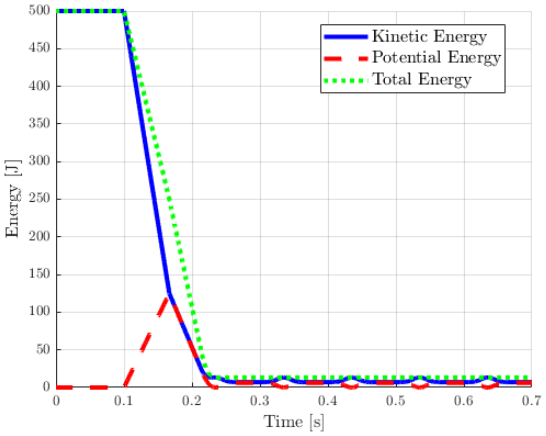
To further investigate the comparison of methods, the local time histories are examined in Figure 6.11 for initial position $X = [10]$m for Bar One and $X = [20]$m for Bar Two. The following results compare five methodologies, the two for the proposed method as mentioned previously, with the addition of a third proposed method termed inconsistent Hugoniot where the Hugoniot pressure model is applied throughout the simulation (despite the model being invalid when in tension and after separation, refer to Section 2.6.3). These results are then compared with two traditional FEMs, linear FEM and mean dilation which were simulated in Abaqus/Explicit discretised appropriately for the two methods using 2D plane strain elements with equivalent meshes. The artificial numerical dissipation (linear bulk viscosity) was removed from the linear FEM to demonstrate its significance, this is clearly visualised in the velocity and axial stress histories (Figures 6.11a - 6.11d) which show spurious oscillations throughout the linear FEM solution after initial contact. No other method shows the same spurious oscilla-



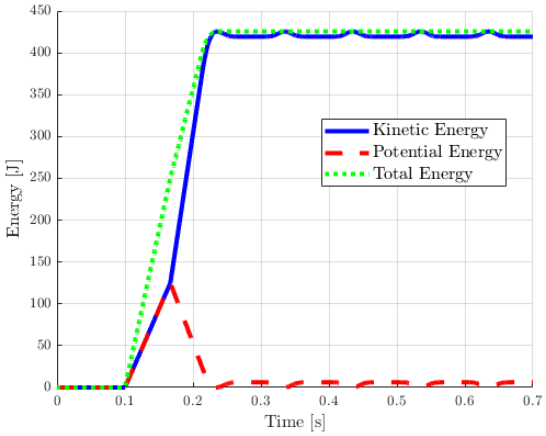
(a) Global total energy - Comparison of Riemann solvers



(b) Global dissipation - Mesh refinement



(c) Energy Measures - Bar One



(d) Energy Measures - Bar Two

Figure 6.10: Two Identical Hugoniot Bars - Time evolution of global energies and dissipations comparing different Riemann solvers and mesh refinements

tions with good agreement across all three variables. To more closely examine the local histories the linear FEM solution is removed, resulting in Figure 6.12. During contact the four methods match closely until separation occurs, where a slight difference in velocity and displacement histories is observed at initial separation and after complete separation in Figures 6.12a and 6.12b for velocity, and in Figures 6.12c and 6.12d for displacement while the axial stress histories match nearly exactly in Figures 6.12c and 6.12d. The results from the proposed method using acoustic and non-linear Riemann solvers match nearly exactly while the mean dilatation method has a slightly slower separation which coincides with the inconsistent Hugoniot method. This suggests the mean dilation method presented here does not accurately model the separation as the Hugoniot model is used in computing the compression and tension pressure waves resulting in a slower separation and lower magnitude post-separation reflection waves as the bars contract and expand in comparison with the proposed method.

It is therefore concluded that, the proposed method using the Hugoniot pressure model is successfully benchmarked for both acoustic and non-linear Riemann solvers compared with the mean dilatation method when in contact (compression) as the results match exactly. For separation on the other hand, the mean dilatation method serves as an approximation while the proposed method is valid for both compression and tension pressure waves. By comparing the acoustic and non-linear Riemann solver approaches it is observed that for this simple one-dimensional problem, no significant difference is observed throughout the simulation, this is likely due to the Hugoniot pressure constitutive model dominating the solution using both solvers. This suggests that for internal wave propagation an acoustic Riemann solver is sufficient to model the shock wave propagation. For the boundary however, in particular for contact, further investigation is required for multi-dimensional scenarios where the normal pressure plays a role in the computation of tangential components such as angled or frictional contact.

Comparison of Shock Wave Slope

To further investigate the importance of the linear relationship between the shock wave speed and the particle wave speed a parametric study was conducted using the non-linear Riemann solver for the values of the slope $s = \{0, 0.25, 0.5, 0.75, 1\}$. The comparison of the local time histories at $X = [10]$m for Bar One (first column) and $X = [10]$m for Bar Two (second column) are shown in Figure 6.13.

From these results, the value of slope s plays a significant role in each key variable, velocity, stress and displacement. In terms of velocity in Figures 6.13a and 6.13b, it can be observed that there is no effect prior to or during contact, with all simulations obtaining a 50m/s contact velocity at $t = 0.1$ s. The effect of s is observed in the separation time and magnitude, as the value of s increases the separation occurs earlier with a decrease in shock gradient, for example at $s = 0$ separation begins at $t = 0.3$s with an instantaneous release and full transfer of kinetic energy (velocity) from Bar One to Bar Two, while at $s = 1$ separation, in terms of velocity, begins at $t \approx 0.24$s and fully separates at $t \approx 0.27$s with the majority of kinetic energy transferred to Bar Two and Bar One retaining a small proportion of kinetic energy (velocity). In terms of axial stress in Figure 6.13c and 6.13d, it can be observed that for increasing values of s the axial stress during contact increases from -50Pa for $s = 0$ to -75Pa for $s = 1$ and the earlier separation can also be observed, in terms of stress, with separation beginning at $t \approx 0.21$s and finishing at $t \approx 0.24$s. Finally, in terms of displacement in Figures 6.13e and 6.13f, the observations from the velocity time histories are again evident, where the value of s only affects the separation time and post separation displacement. These observations are a direct result of the high relative impact velocity combined with the Hugoniot

pressure model producing an increase in shock wave speed emanating from the contact region. In this scenario the ratio of the jump in velocity $[\![v_n]\!]$ to acoustic wave speed c_p is equal to 1 therefore the change in shock wave speed is a direct result of the change in material parameter s and can be calculated analytically from Equation (2.44) for each value of s as $U_p = \{100, 125, 150, 175, 200\}$ m/s. This clearly indicates that the higher value of s results in a higher shock wave speed and stress magnitude, therefore faster wave propagation and results in a faster separation however for low values of impact velocity, such as 0.1m/s the shock wave speed are $\{100, 100.25, 100.5, 100.75, 101\}$ for each value of s respectively. The impact of slope s on the solution is minor as the shock wave speed is approximately the acoustic wave speed. To further investigate these results, the influence of impact velocity will now be demonstrated.

Comparison of Impact Velocity

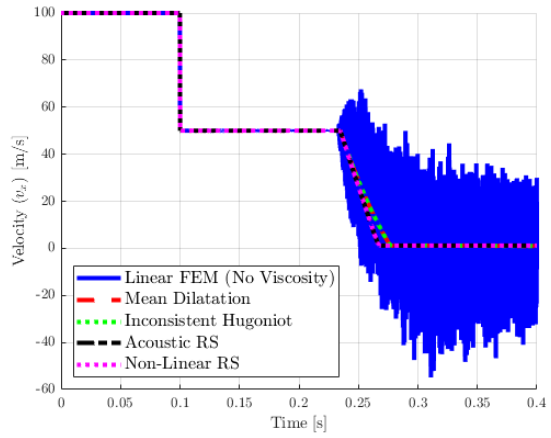
To investigate the role of impact velocity, a parametric study was conducted using the non-linear Riemann solver for the values of initial velocity $v_0^1 = \{0.1, 1, 10, 100\}$m/s. For each velocity, the initial gap was set to $\delta_n = \{0.01, 0.1, 1, 10\}$m to ensure initial contact at $t = 0.1$s. The comparison of the local time histories using a logarithmic Y axis are shown in Figure 6.14 for axial stress and displacement.

By first examining the axial stress in Figures 6.14a and 6.14b, it is observed that at initial contact $t = 0.1$s the stress magnitude increases significantly. The axial stress then remains constant for each initial velocity respectively at $\{-0.05, -0.5, -5.25 - 75\}$ Pa before separating at different times consistent with previous results for different values of slope s. For the two slower impact velocities of 0.1 and 1m/s, it can be observed that the bars separate instantaneously at $t = 0.3$s which matches exactly with the analytical solution for the linear elastic case in Section 6.2, even when considering a Hugoniot pressure model with $s = 1$. While for the two faster impact velocities 10m/s and 100m/s, an earlier separation is observed, with a gradual separation. For $v_0^1 = 10$m/s, separation begins at $t = 0.28$s and finishes at $t = 0.29$s while for $v_0^1 = 100$m/s, separation begins at $t \approx 0.21$s and finishes at $t \approx 0.24$s. This gradual separation is also observed in the displacement results from Figures 6.14c and 6.14d. These observation are explained by examining the speed of wave propagation for each scenario i.e. the shock wave speed, Equation (2.44), which are calculated as $U_p = \{100.1, 101, 110, 200\}$m/s. In the first two scenarios $U_p \approx c_p$ therefore these scenarios are dominated by the acoustic speed of the material. As the impact velocity increases the difference in shock wave speed to acoustic wave speed increases significantly, demonstrating that after initial impact the compressive shock wave travels significantly faster than the acoustic wave speed of the material to the free end, before reflection back to the contact interface thus resulting in an earlier separation. This suggests that for low relative velocity impacts

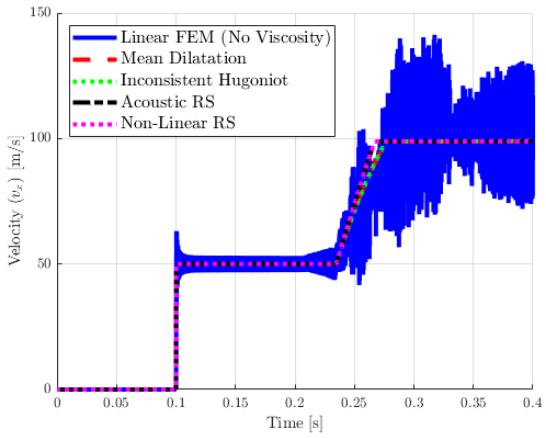
i.e. when the acoustic wave speed of the material is significantly larger than the impact velocity, the Hugoniot constitutive model is not required to model the shock produced by contact accurately. However as the impact velocity increases and it approaches the acoustic wave speed of the material the Hugoniot constitutive model is required to model the shock wave propagation accurately. These observations draw similarities to the Mach number from fluid dynamics and can be expressed equivalently as a ratio of the particle velocity $[\![v_n]\!]$ to acoustic wave speed c_p from Equation (2.44) as:

$$M = \frac{[\![v_n]\!]}{c_p}. \tag{6.1}$$

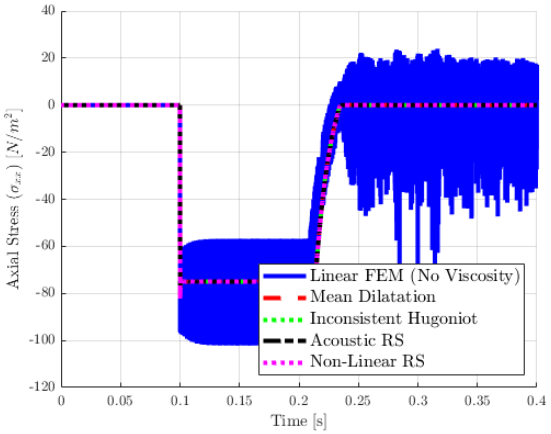
This ratio proves useful to determine when to use the Hugoniot constitutive model, for each impact velocity the ratio is calculated to be $\{0.001, 0.01, 0.1, 1\}$. It can be observed that the higher the ratio the greater the effect on the wave propagation therefore it can be concluded that at low ratios ($M < 0.01$) the Hugoniot model is not required while if the ratio increases towards transonic ($M = 0.8 - 1.2$) or supersonic ($M > 1.2$) shock wave speeds the use of the Hugoniot pressure model is essential to model the material response accurately.



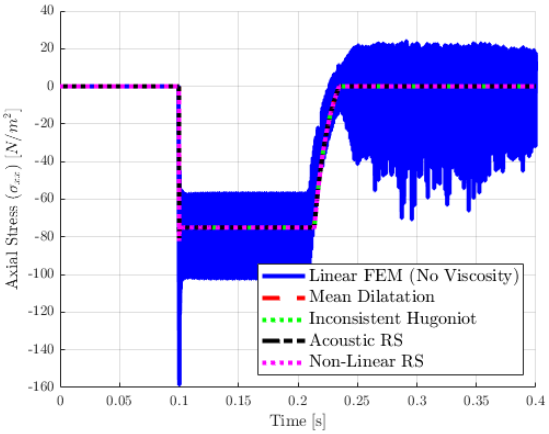
(a) Velocity v_x - Bar One



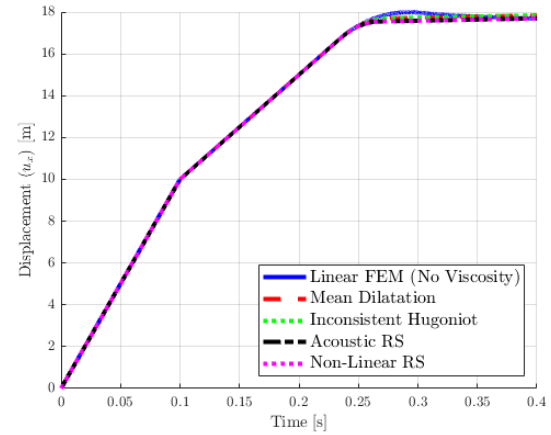
(b) Velocity v_x - Bar Two



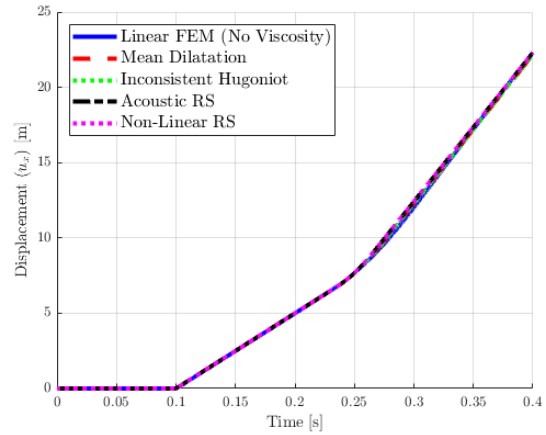
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two

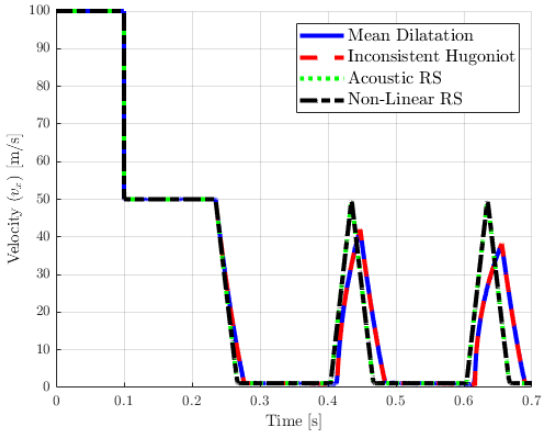


(e) Displacement u_x - Bar One

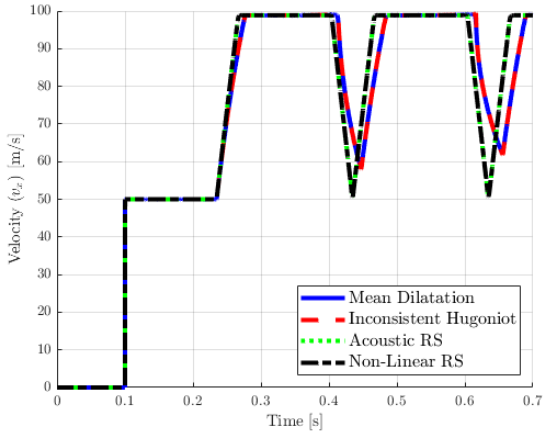


(f) Displacement u_x - Bar Two

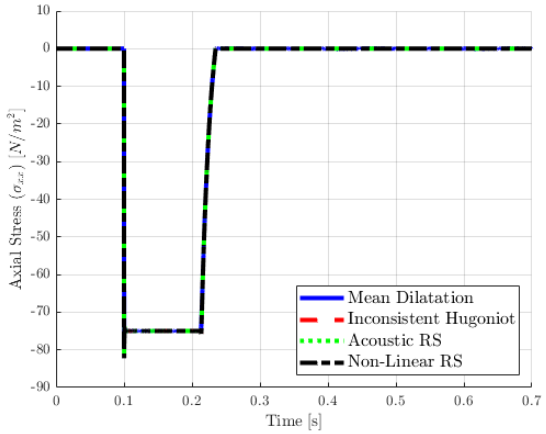
Figure 6.11: Two Identical Hugoniot Bars - Time evolution monitored at $X = [10]$m (first column) and $X = [20]$m (second column) comparing linear FEM without bulk viscosity, mean dilatation with bulk viscosity, inconsistent Hugoniot constitutive model, acoustic Riemann Solver and non-linear Riemann Solver



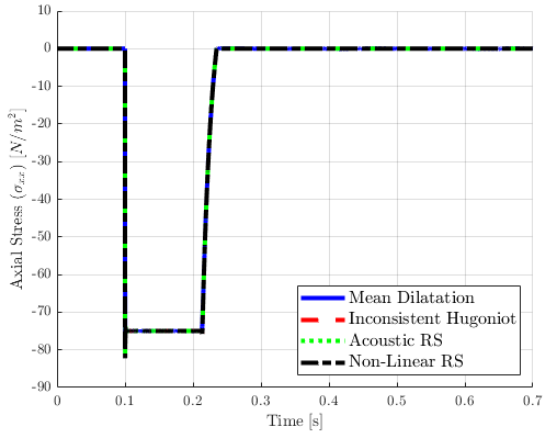
(a) Velocity v_x - Bar One



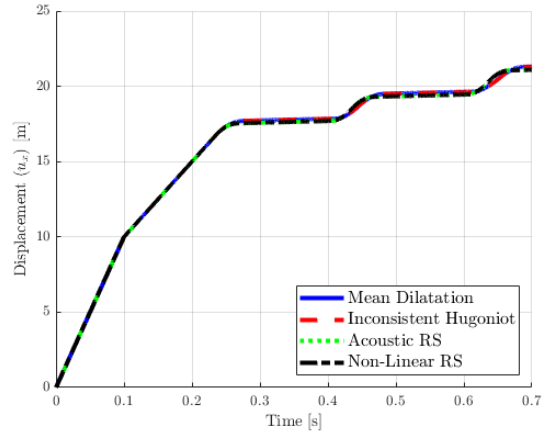
(b) Velocity v_x - Bar Two



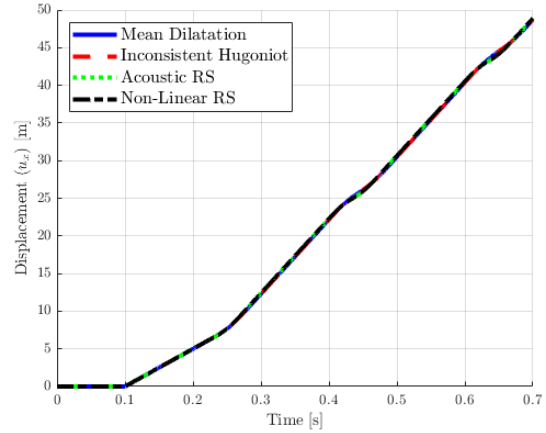
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two

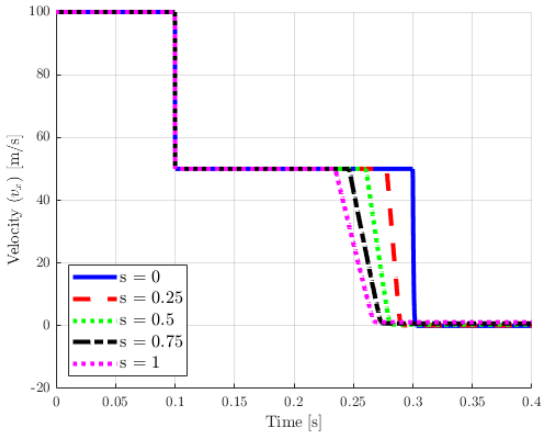


(e) Displacement u_x - Bar One

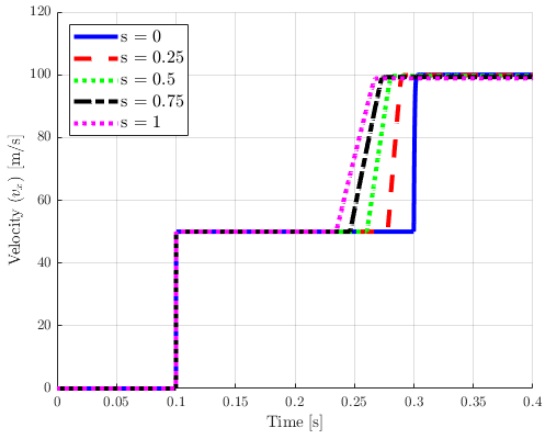


(f) Displacement u_x - Bar Two

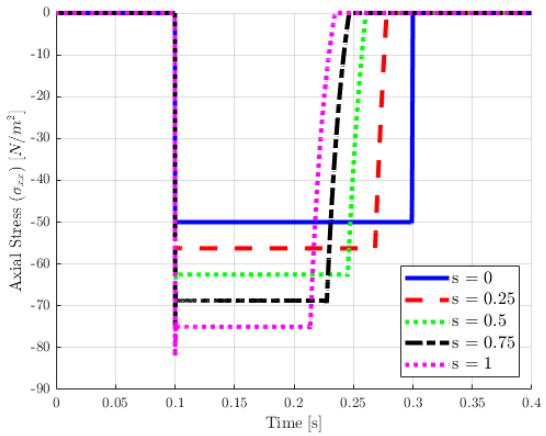
Figure 6.12: Two Identical Hugoniot Bars - Time evolution monitored at $X = [10]$m (first column) and $X = [20]$m (second column) comparing mean dilatation with bulk viscosity, inconsistent Hugoniot constitutive model, acoustic Riemann Solver and non-linear Riemann Solver



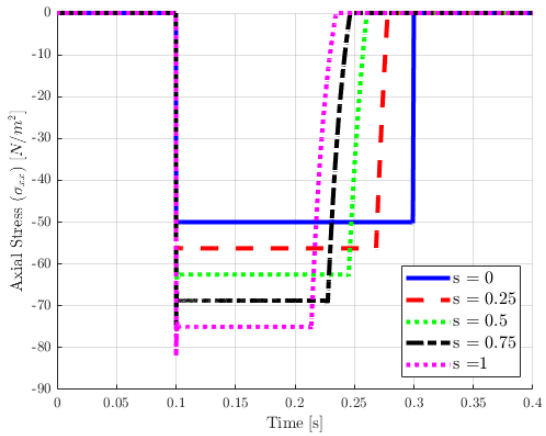
(a) Velocity v_x - Bar One



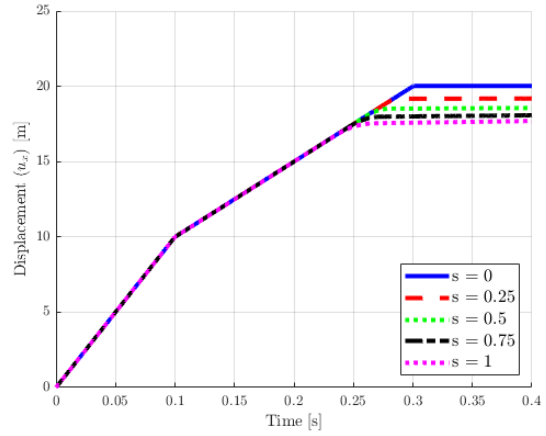
(b) Velocity v_x - Bar Two



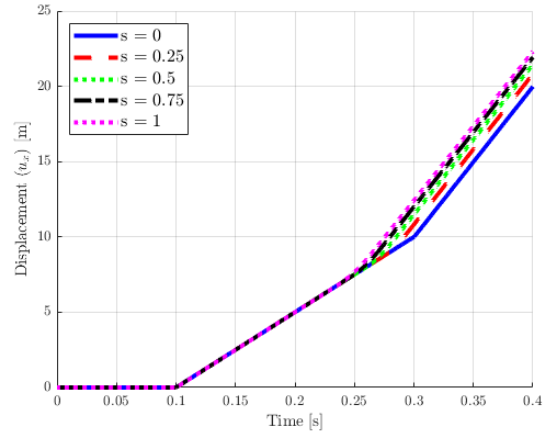
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two

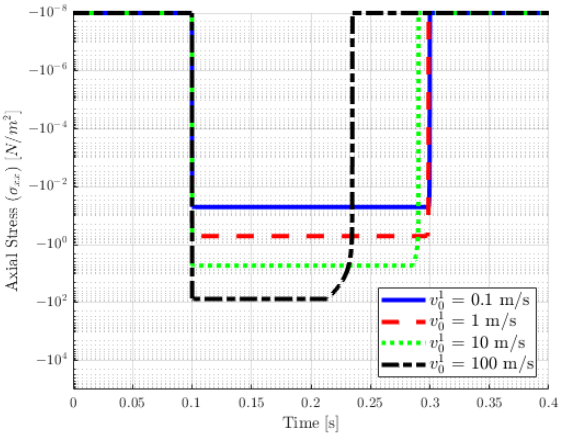


(e) Displacement u_x - Bar One

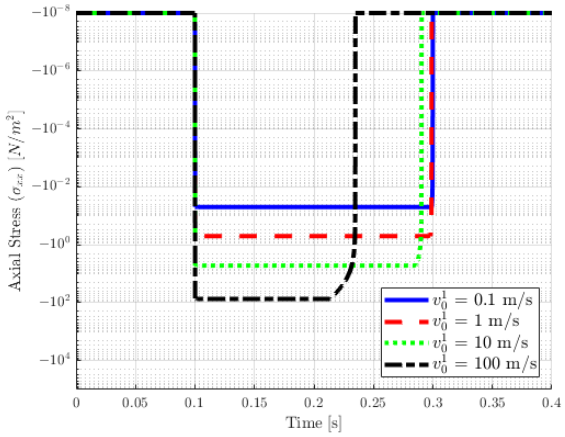


(f) Displacement u_x - Bar Two

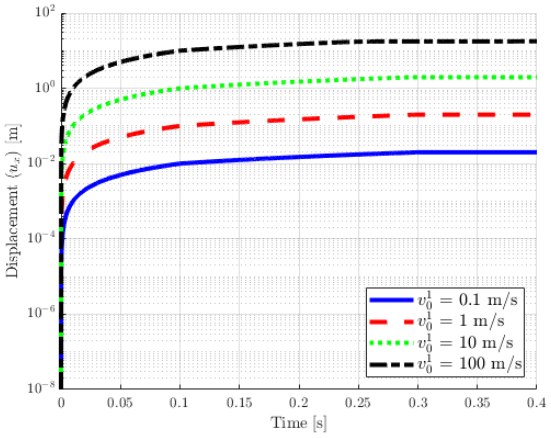
Figure 6.13: Two Identical Hugoniot Bars - Time evolution monitored at $X = [10]$m (first column) and $X = [20]$m (second column) comparing different values of shock wave slope $s = \{0, 0.25, 0.5, 0.75, 1\}$



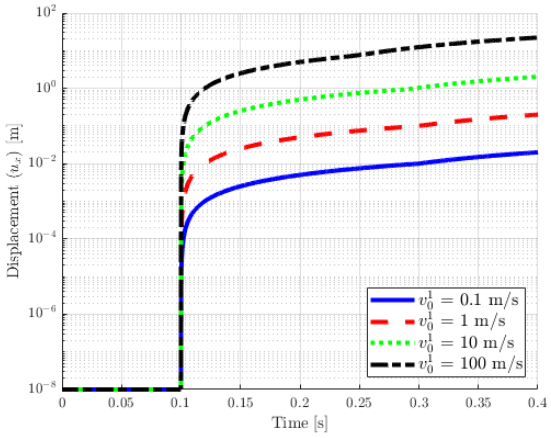
(a) Axial Stress σ_{xx} - Bar One



(b) Axial Stress σ_{xx} - Bar Two



(c) Displacement u_x - Bar One



(d) Displacement u_x - Bar Two

Figure 6.14: Two Identical Hugoniot Bars - Time evolution monitored at $X = [10]$m (first column) and $X = \{10.01, 10.1, 11, 20\}$m (second column) comparing different values of impact velocity $v_0^1 = \{0.1, 1, 10, 100\}$

6.5 Two Dissimilar Hugoniot Bars

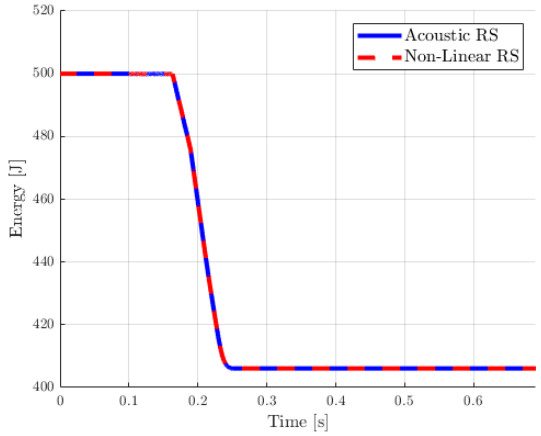
6.5.1 Problem Description

The next problem considered the impact of two one-dimensional Hugoniot bars, similar to the problem set-up from Figure 6.9 with Bar Two now with a reference density of 0.02 $\mathrm{kg\,m^{-3}}$ instead of 0.01 $\mathrm{kg\,m^{-3}}$. The objective of the problem is to benchmark the proposed method against standard FEMs for a dissimilar contact interface. Comparison after separation will not be considered (after $t = 0.3$s) in this section since the Hugoniot model when applied to traditional FEMs, as discussed in Section 6.4, is invalid as it is applied in both compression and tension. All simulations for the proposed method were conducted with a CFL of 0.3.

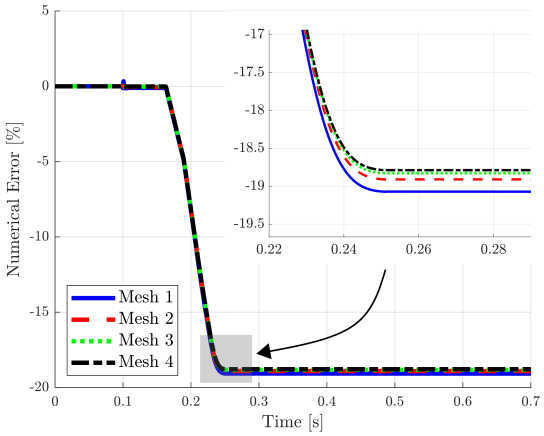
6.5.2 Numerical Results

As with previous problems, the first step was to compare the energy profiles to ensure consistency as shown in Figure 6.15 for various energy time evolutions. First, shown in Figure 6.15a and 6.15b are the comparison of methods for total energy and dissipation respectively, comparing the proposed method using an acoustic Riemann solver and non-linear Riemann solver. From these two figures it is observed that the energy and dissipation using the proposed method match exactly for the two different Riemann solvers, as observed in the identical Hugoniot bar problem in Section 6.4. At initial contact, $t = 0.1$s, the total energy for the proposed methods remain nearly constant at $\approx$ 500 J until separation begins to occur at $t \approx 0.17$s the energies then gradually decrease to $\approx$ 406J. This is also observed in the dissipation, Figure 6.15b. In Figure 6.15c is the dissipation time evolution comparing the four meshes consisting of $\{128, 256, 512, 1024\}$ elements per bar, the consistency of the proposed method is observed as the number of elements increases the result tends to a steady state solution. Lastly, in Figure 6.15d and Figure 6.15e are the time evolutions of different energy measures (total, kinetic and potential) using Mesh 4 for Bar One and Bar Two respectively. The effect of the increased density of Bar Two can clearly be observed by comparing these results to those in Figure 6.10d and Figure 6.10e. Next, the local time histories at the contact nodes, position $X = [10]$m and $X = [20]$m for Bar One and Two respectively, are compared for the four methods in Figure 6.16. By examining these results, clear agreement is observed across all three variables, velocity, axial stress and displacement for all four methodologies. Slight overshoot is observed in both velocity and axial stress time histories for each method at initial contact, consistent with previous results. The primary observable difference between methods is observed in the velocity profiles, Figure 6.16a and 6.16b, as the bars begin to separate. Again, as ob-

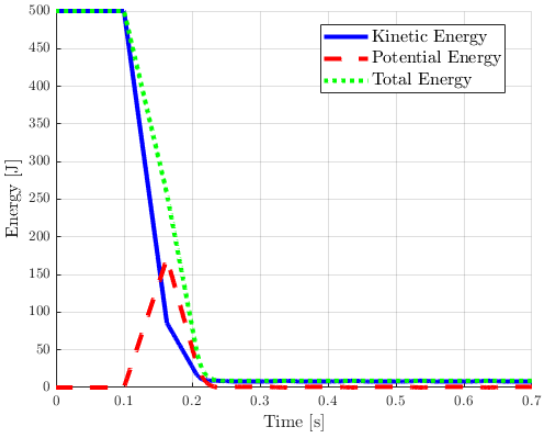
served previously in Section 6.4, this is likely caused by the different constitutive model being applied at separation as the FEM and Inconsistent Hugoniot proposed method match exactly. While the two Riemann solver methods with valid Hugoniot model also match exactly.



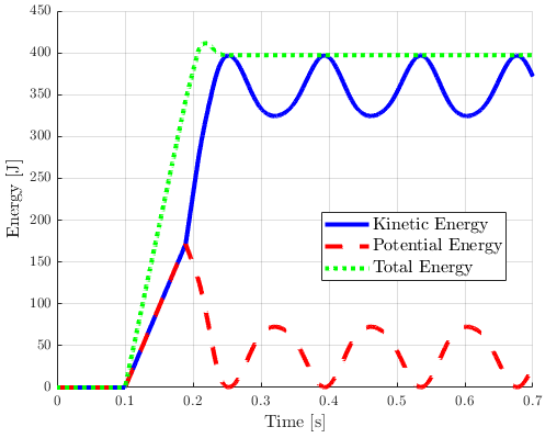
(a) Global total energy - Comparison of methods



(b) Global dissipation - Mesh refinement

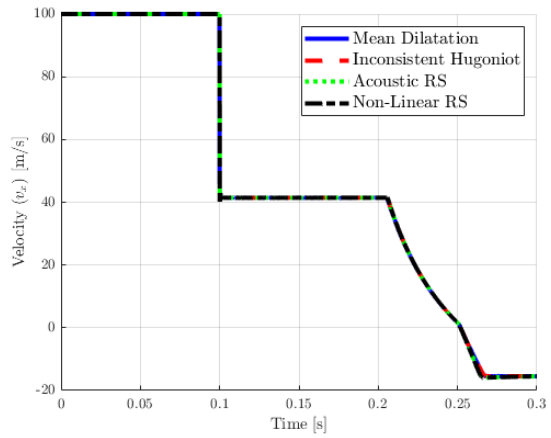


(c) Energy measures - Bar One

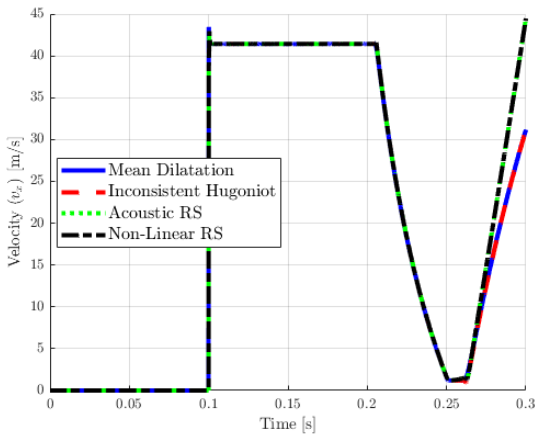


(d) Energy measures - Bar Two

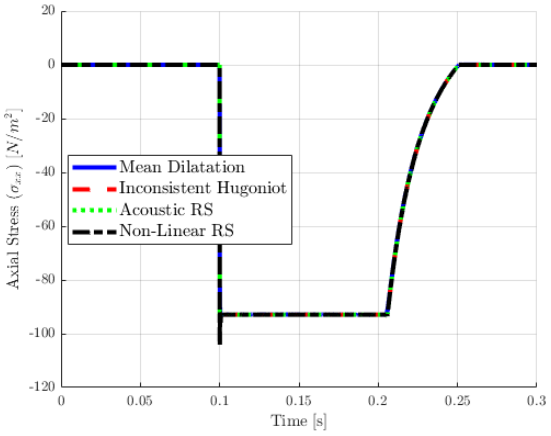
Figure 6.15: Two Dissimilar Hugoniot Bars - Time evolution of global energies and dissipations comparing different Riemann solvers and mesh refinements



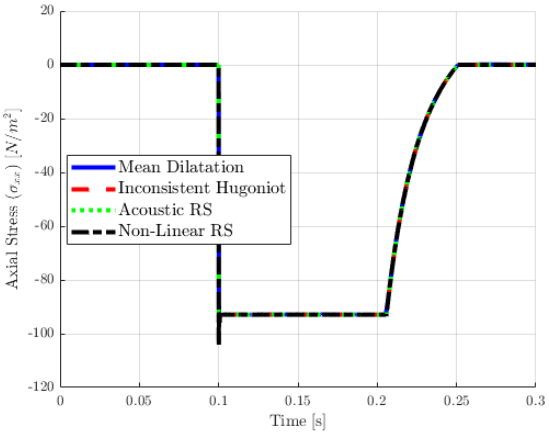
(a) Velocity v_x - Bar One



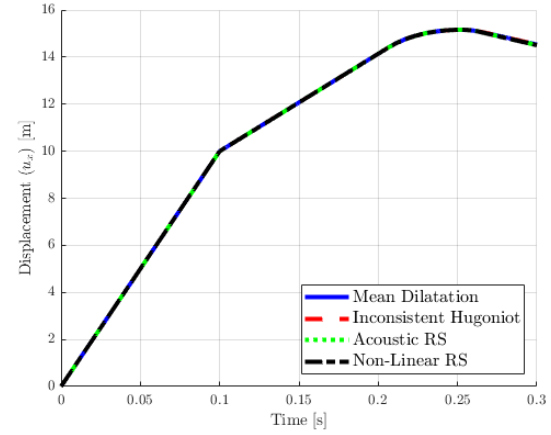
(b) Velocity v_x - Bar Two



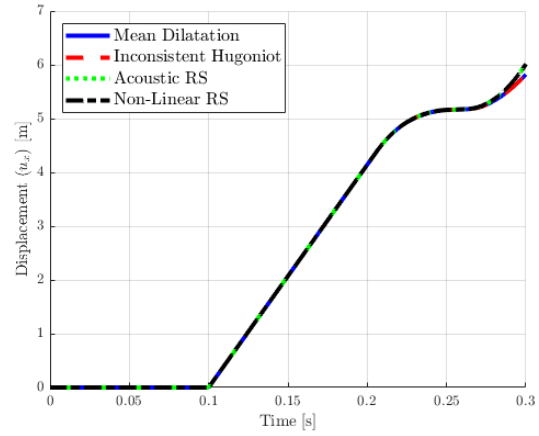
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two



(e) Displacement u_x - Bar One



(f) Displacement u_x - Bar Two

Figure 6.16: Two Dissimilar Hugoniot Bars - Time evolution monitored at $X = [10]$m (first column) and $X = [20]$m (second column) comparing mean dilatation, inconsistent Hugoniot constitutive model, acoustic Riemann solver and non-linear Riemann solver

Chapter 7

Multi-Dimensional Problems

7.1 Preliminaries

In this Chapter, a series of multi-dimensional problems addressing different objectives are investigated for six different multi-dimensional scenarios. First in Section 7.2, the impact between curved surfaces is investigated through a two-dimensional problem consisting of two identical compressible rings. The primary objective of this problem was to ensure no spurious modes exist in the simulation for the proposed method which is common in linear FEM. The second objective was to investigate the presence of pressure checker boarding through the problem presented in Section 7.3. This two-dimensional problem considered the impact of two identical nearly incompressible bars which typically exhibit pressure checker boarding when simulated using linear FEM. The third objective was to investigate pressure checker boarding in more complex material models, in particular von-Mises plasticity with isotropic hardening. First, in Section 7.4 the classic plasticity benchmark problem known as the Taylor bar impact [155] and will be used to demonstrate the accuracy and stability of the three-dimensional implementation through comparisons against published results for various computational methods. Then in Section 7.5 a torus floor impact problem is investigated for both elastic and plastic material models. The next problem investigated, in Section 7.6, is the three-dimensional impact of two nearly incompressible bars with non-matching contact interfaces. Finally to demonstrate a practical application, a simplified car wall impact is investigated in Section 7.7.

7.2 Compressible Ring Impact

7.2.1 Problem Description

The first objective was to explore more complex two dimensional geometry, in this case a curved surface, and compare with standard FEMs using commercial software Abaqus/Explicit [16]. This problem considered two identical compressible rings travelling at equal but opposite initial velocity $\boldsymbol{v_0}$ of $\pm[0.59, 0]^T$m/s with an initial gap δ_n of 8mm as depicted in Figure 7.1. The rings were modelled using a hyperelastic neo-Hookean constitutive model with material properties presented in Table 7.1.

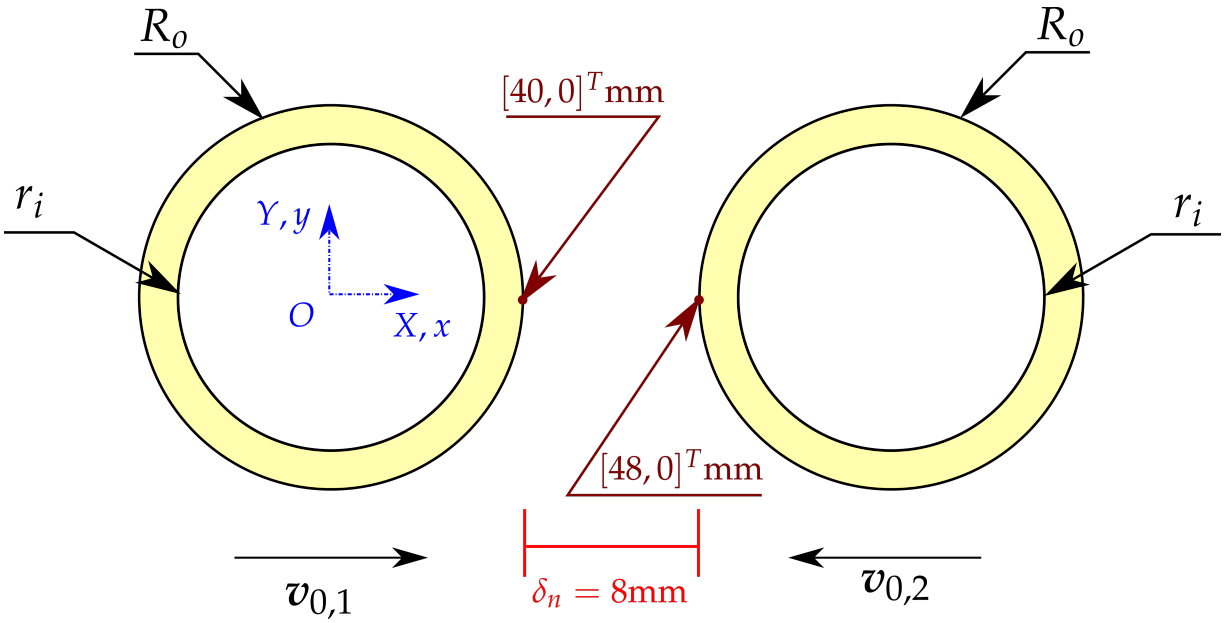


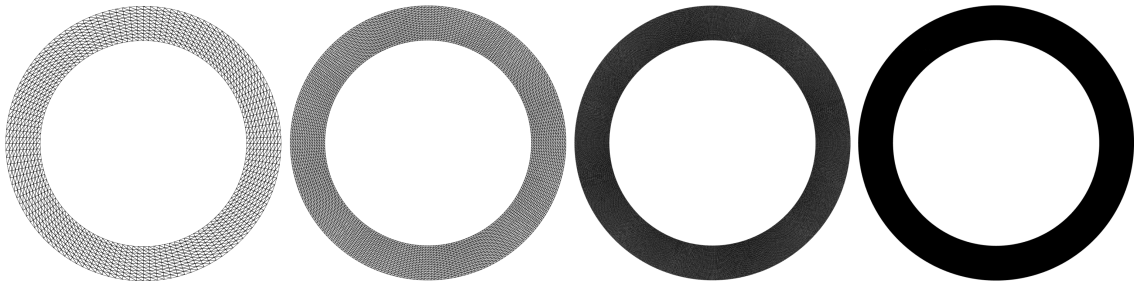
Figure 7.1: Compressible Ring Impact - Problem description

Table 7.1: Compressible Ring Impact - Material parameters

Young's modulus	E_0	1×10^6	$[\mathrm{N/m^2}]$
Material density	ρ_0	1000	$[\mathrm{kg/m^3}]$
Poisson's ratio	ν	0.4	
Lamé parameters	μ	0.35714	$[\mathrm{MN/m^2}]$
	λ	1.42857	$[\mathrm{MN/m^2}]$

This problem could have been simulated as a symmetrical problem but to explore the proposed multi-body contact algorithm both rings were simulated to investigate the accuracy of the proposed method. As with the one-dimensional scenarios, the first step was to ensure consistency of the proposed method by conducting a mesh sensitivity study, using a CFL of 0.3 the four meshes investigated are as shown in Figure 7.2

with mesh information as follows per ring: Mesh 1 had 1364 nodes and 2480 elements, Mesh 2 had 5292 nodes and 10080 elements, Mesh 3 had 20828 nodes and 81280 elements, Mesh 4 had 82620 nodes and 163200 elements. The results from the proposed method could then be compared against the results from two different standard FEMs, in particular linear FEM using linear triangular elements and mean dilatation FEM using quadrilateral elements. The meshes used were as follows: the linear FEM mesh consisted of 82944 nodes and 163840 elements per ring and the mean dilatation FEM consisted of 82944 nodes and 81920 bi-linear quadrilateral elements per ring.

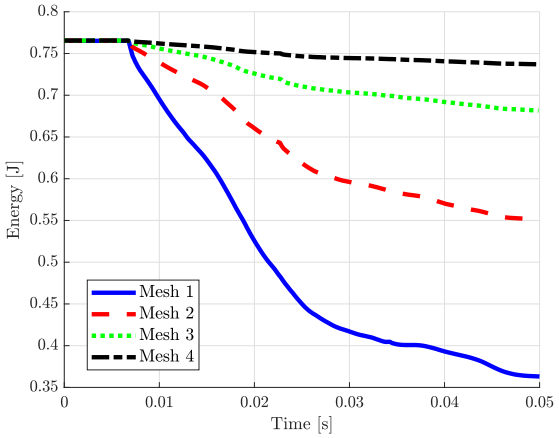


(a) (b) (c) (d)

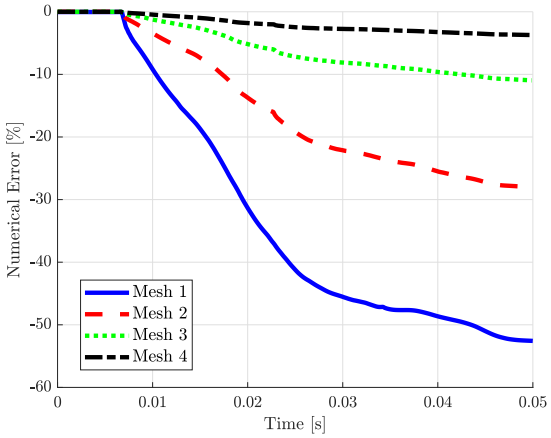
Figure 7.2: Compressible Ring Impact - (a) Mesh 1 - 1364 nodes and 2480 elements (b) Mesh 2 - 5292 nodes and 10080 elements (c) Mesh 3 - 20828 nodes and 81280 elements (d) Mesh 4 - 82620 nodes and 163200 elements

7.2.2 Numerical Results

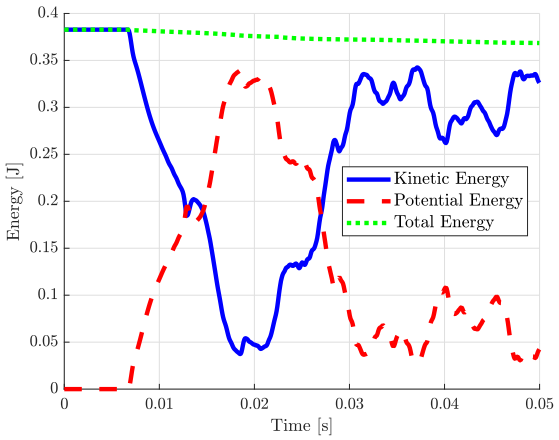
The first results presented are from the mesh sensitivity study as shown in Figure 7.3, the consistency of the proposed method is again observed now in two dimensions. The global pressure contour snapshots are then shown in Figure 7.4 and compared with mean dilatation FEM. From the pressure contours it is clear that the solution using the proposed method is globally stable and matches closely with the solution from mean dilatation FEM. The next results presented are the local time evolution (velocity v_x and axial stress σ_{xx}) of Ring One at initial position $\boldsymbol{X} = [40, 0]^T$mm for the mesh refinement of the proposed method (Mesh 1 - 4) compared with mean dilatation FEM as a reference solution. Figure 7.5 shows that as the mesh density of the proposed method using a triangular mesh increases, the local results tend towards the solution using the mean dilatation FEM, demonstrating the accuracy of the proposed scheme. This is further proven through Figure 7.6 which shows the local result evolution for the proposed method using a triangular mesh (Mesh 4), linear FEM and mean dilatation FEM. It is clear that all three methods match closely for the local velocity and displacement evolution, however the linear FEM for axial stress oscillates significantly throughout the solution even during separation which violates the contact conditions while the proposed method does not while using a similar triangular mesh.



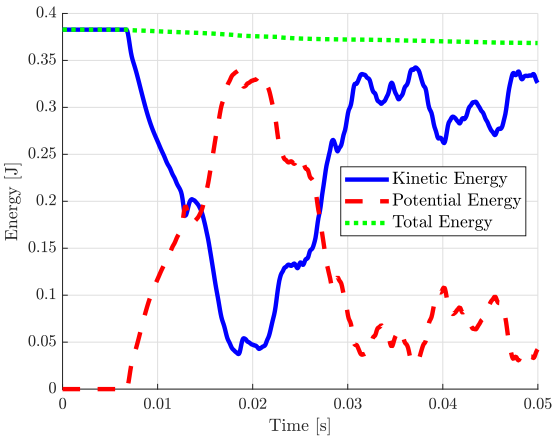
(a) Total system energy - Mesh refinement



(b) Numerical error - Mesh refinement



(c) Energy history - Ring One (Mesh 4)



(d) Energy history - Ring Two (Mesh 4)

Figure 7.3: Compressible Ring Impact - Energy time history comparing Mesh 1-4 for the proposed method

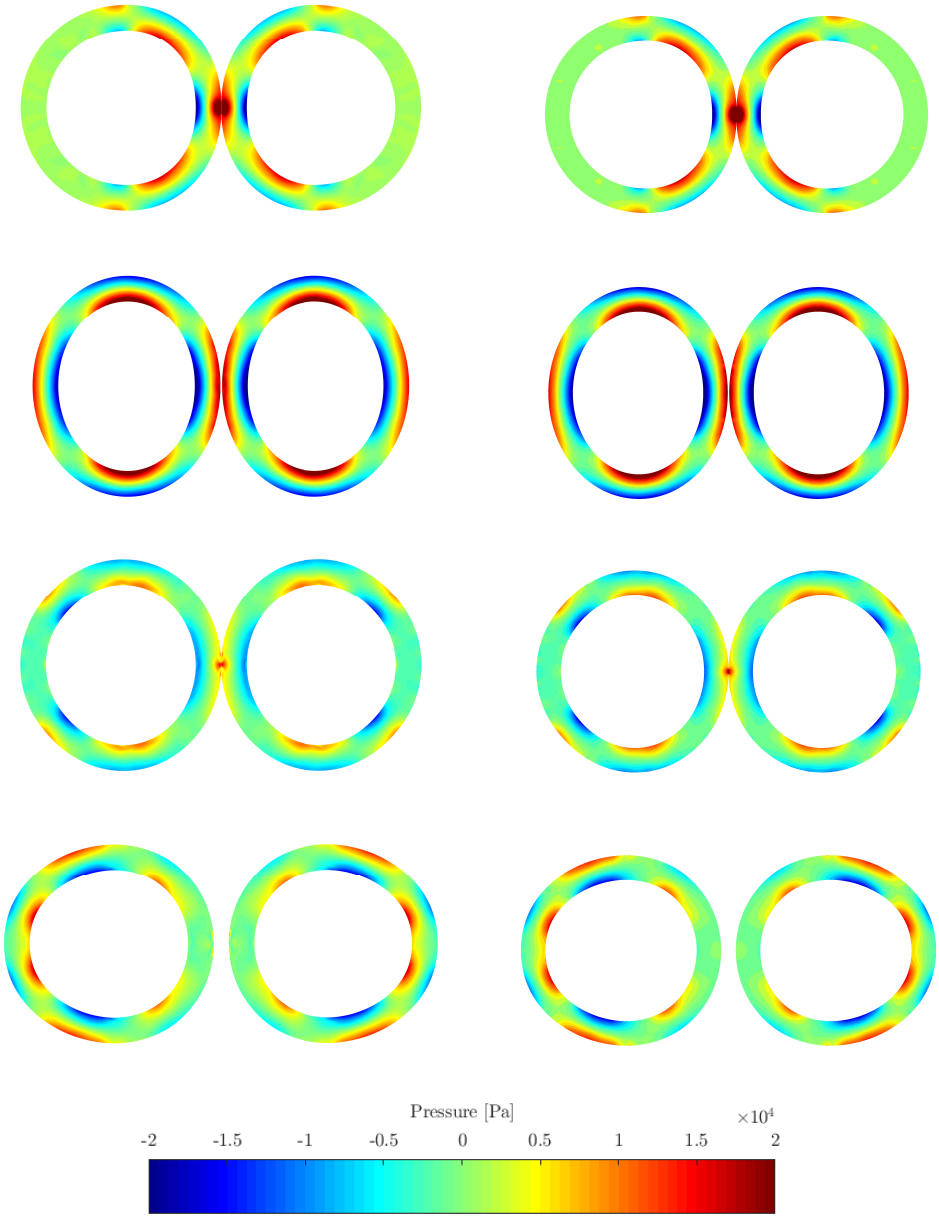
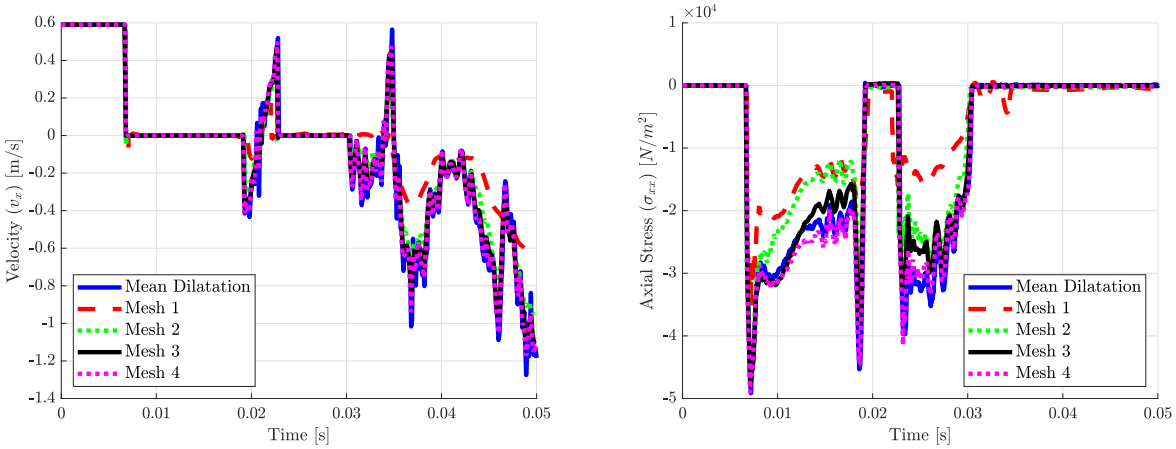


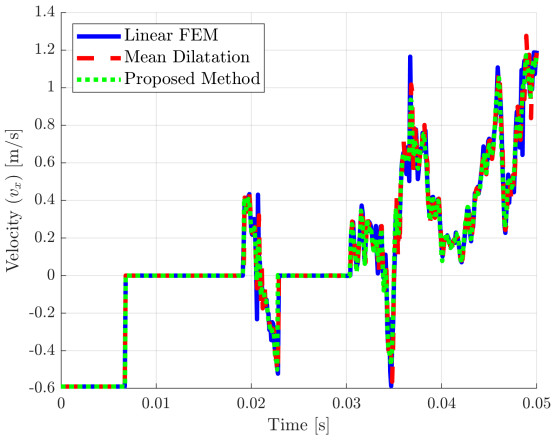
Figure 7.4: Compressible Ring Impact - Global pressure contour snapshots for the proposed method Mesh 4 which uses triangular elements (left) and mean dilatation which uses bi-linear quadrilateral elements (right) at 10ms, 20ms, 30ms and 40ms



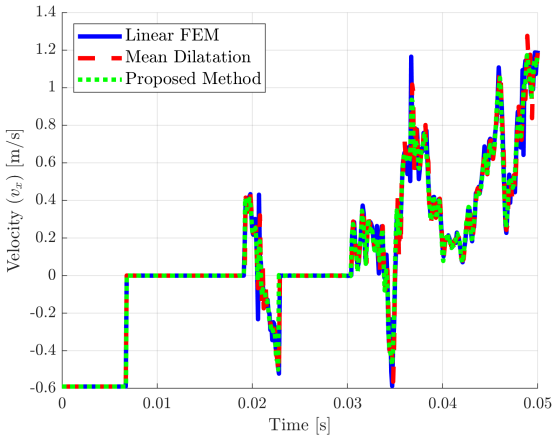
(a) Velocity v_x - Mesh Refinement

(b) Axial stress σ_{xx} - Mesh Refinement

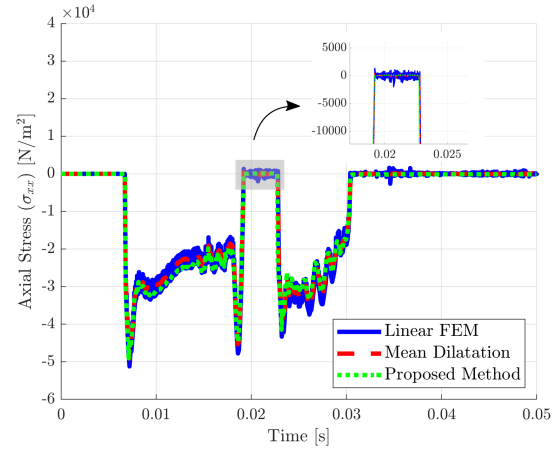
Figure 7.5: Compressible Ring Impact - Comparison of local time evolution of Ring One taken at $\boldsymbol{X} = [40, 0]^T$mm for the proposed method Mesh 1-4 and mean dilatation FEM using bi-linear quadrilateral elements



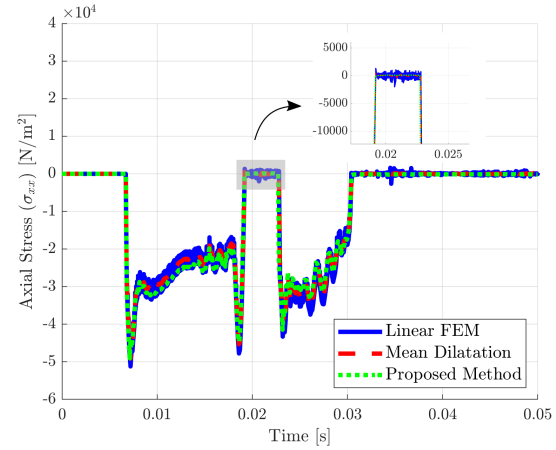
(a) Velocity v_x - Ring One



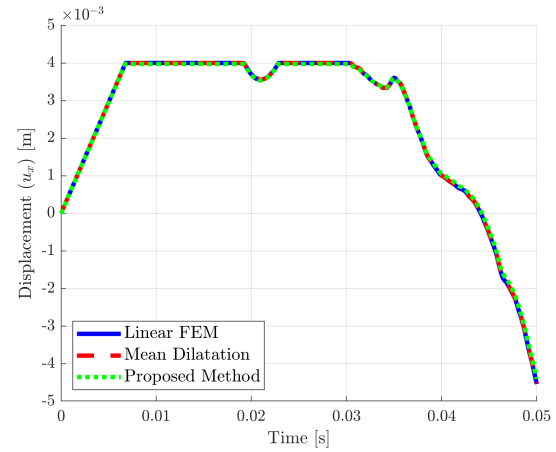
(b) Velocity v_x - Ring Two



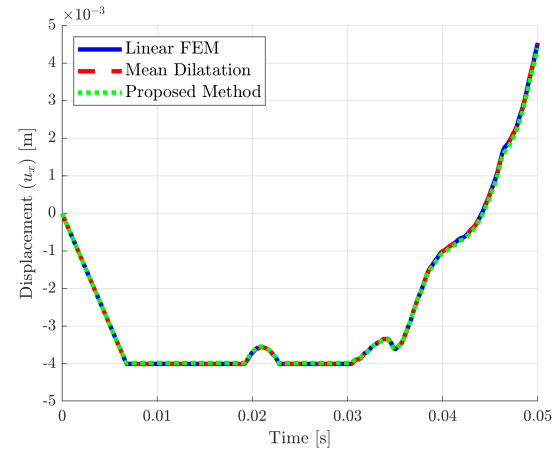
(c) Axial stress σ_{xx} - Ring One



(d) Axial stress σ_{xx} - Ring Two



(e) Displacement u_x - Ring One



(f) Displacement u_x - Ring Two

Figure 7.6: Compressible Ring Impact - Comparison of local results of Ring One taken at $\boldsymbol{X} = [40, 0]^T$mm and Ring Two at $\boldsymbol{X} = [48, 0]^T$mm for the proposed method, linear FEM using triangular mesh and mean dilatation using bi-linear quadrilateral mesh

7.3 Nearly Incompressible Bar Impact

7.3.1 Problem Description

The next problem investigated, considered the impact of two two-dimensional nearly incompressible rectangular bars travelling at equal but opposite velocity, $\boldsymbol{v_0} = [\pm 50, 0]^T \text{m/s}$, as depicted in Figure 7.7 with initial gap δ_n of 8mm, width w of 6.4mm and length L of 32.4mm for each bar. The bars were modelled using a hyperelastic neo-Hookean constitutive model with material properties of each bar shown in Table 7.2. The objective of this problem was to investigate the proposed method's robustness when simulating nearly incompressible materials by comparing the solution with linear FEM and mean dilatation FEM using Abaqus/Explicit [16]. This problem was simulated using a CFL of 0.3 and the meshes had the following mesh information per bar: Mesh 1 had 369 nodes and 640 elements, Mesh 2 had 1377 nodes and 2560 elements, Mesh 3 had 5313 nodes and 10240 elements and Mesh 4 had 20865 nodes and 40960 elements.

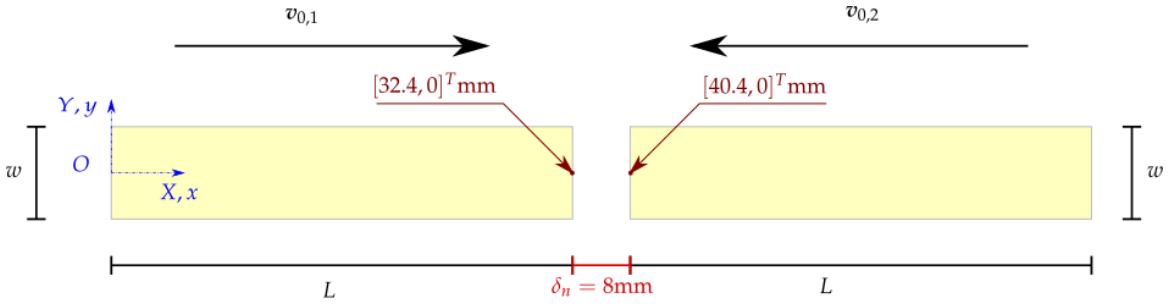


Figure 7.7: Nearly Incompressible Bar Impact - Problem description

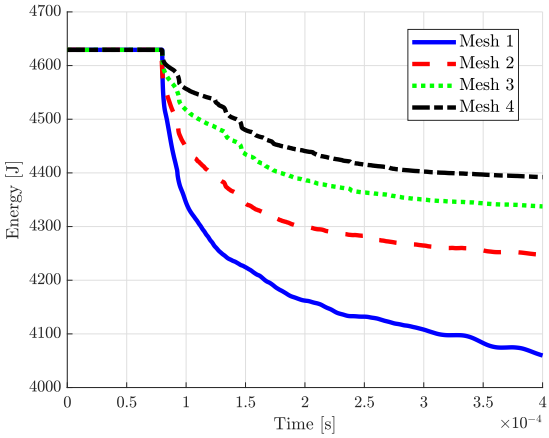
Table 7.2: Nearly Incompressible Bar Impact - Material parameters

Young's modulus	E_0	5.85×10^8	$[\text{N/m}^2]$
Material density	ρ_0	8930	$[\text{kg/m}^3]$
Poisson's ratio	ν	0.495	
Lamé parameters	μ	0.19565	$[\text{GN/m}^2]$
	λ	19.3696	$[\text{GN/m}^2]$

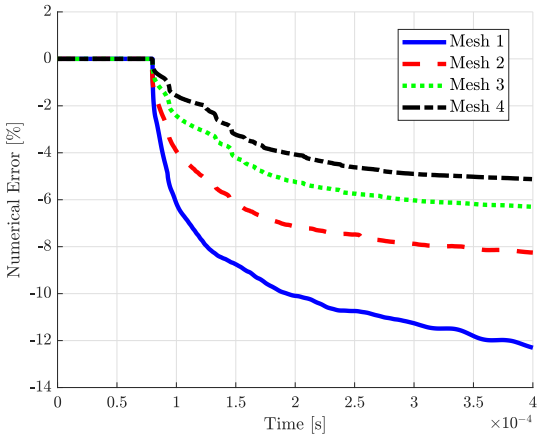
7.3.2 Numerical Results

As with previous examples presented in this research the first step was to prove the consistency of the proposed method for this problem by examining the energy evolution and numerical error, the consistency of the proposed method is again clearly observed in Figure 7.8. This is reiterated in Figure 7.9, which shows the meshes and the improvement in the pressure contour resolution for each mesh at $t = 90\mu$s. An important

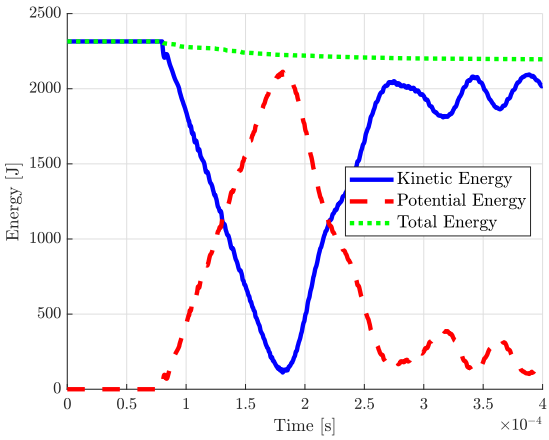
observation from Figure 7.9 is that for the proposed method using a triangular mesh, a smooth pressure contour is obtained for each mesh when modelling nearly incompressible materials. In comparison as presented in Figure 7.10, when using a standard linear FEM using a triangular mesh, pressure checker boarding is observed, while the proposed method matches closely with the mean dilatation FEM solution using bi-linear quadrilateral elements. For completeness the time evolution of the pressure contour plots for the proposed method is presented in Figure 7.12.



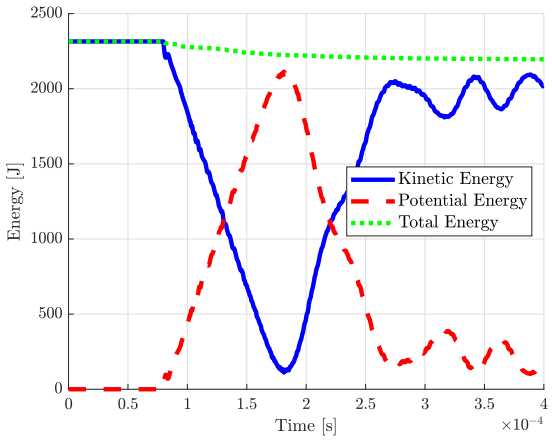
(a) Total System Energy - Mesh Refinement



(b) Numerical Error Mesh Refinement



(c) Energy - Bar One (Mesh 4)



(d) Energy - Bar Two (Mesh 4)

Figure 7.8: Nearly Incompressible Bar Impact - Energy time history of nearly incompressible bar impact comparing Mesh 1-4 for the proposed method

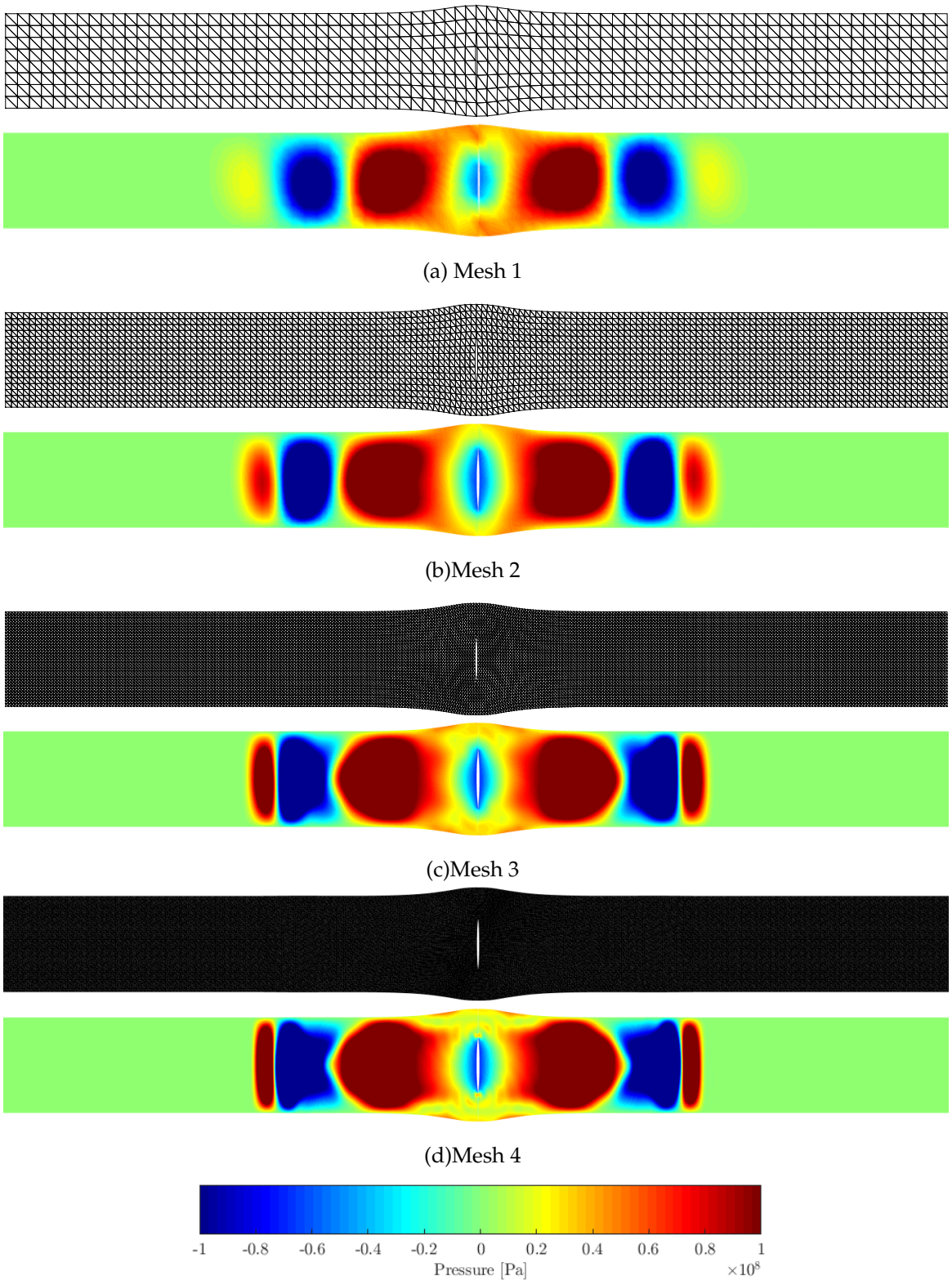


Figure 7.9: Nearly Incompressible Bar Impact - Comparison of pressure contour plots of the nearly incompressible bar impact at 90μs for Mesh 1-4

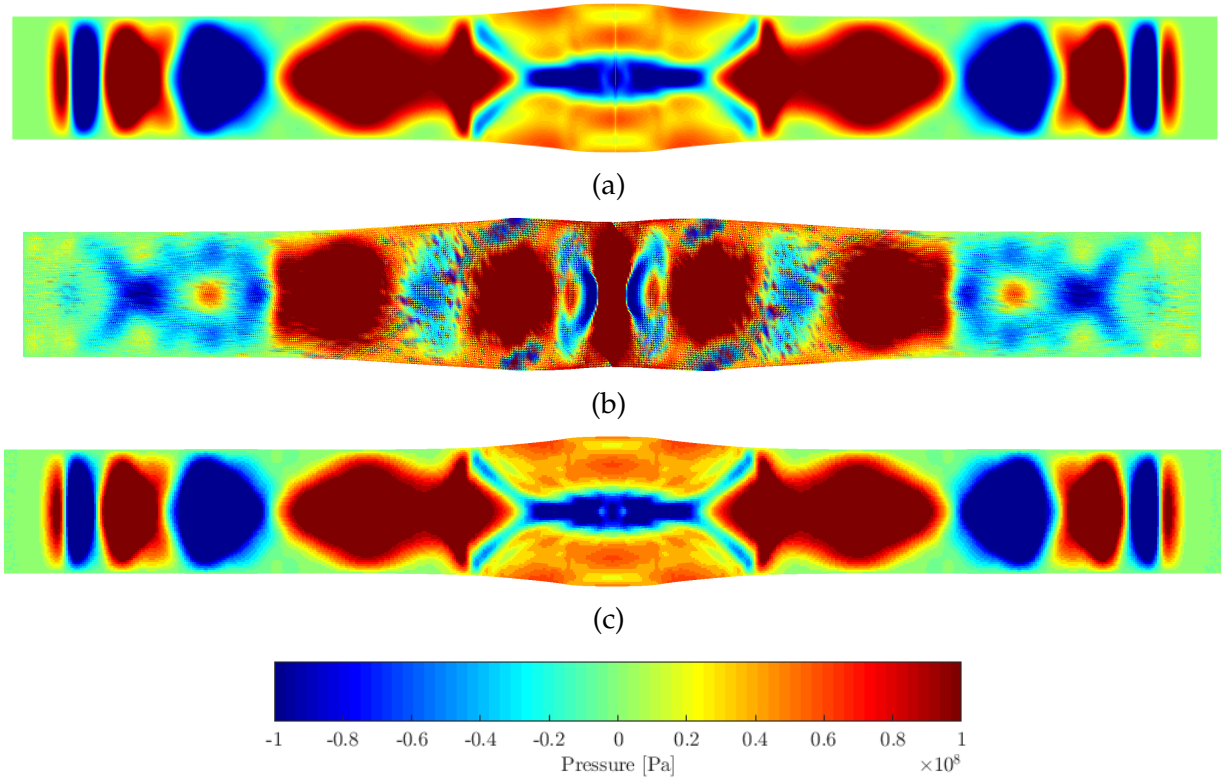
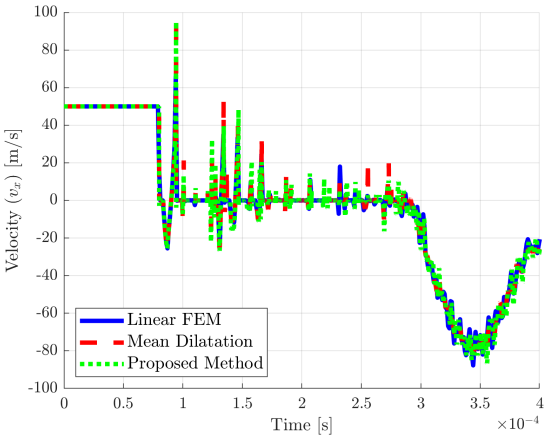
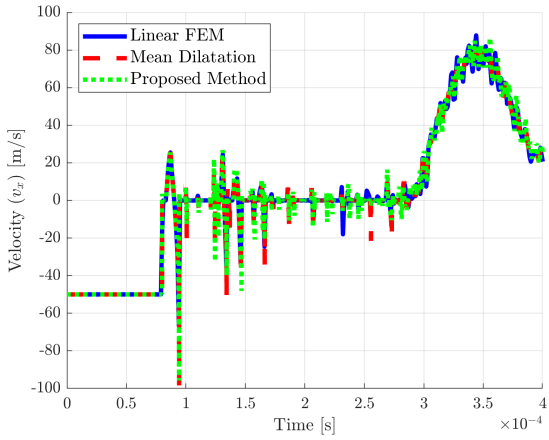


Figure 7.10: Nearly Incompressible Bar Impact - Comparison of pressure contour snapshots of the nearly incompressible bar impact at 100μs (a) proposed method using linear triangular mesh (Mesh 4) (b) Linear FEM using a triangular mesh (Mesh 4) (c) Mean dilatation using a bi-linear quadrilateral mesh (20865 nodes, 20480 elements per bar)

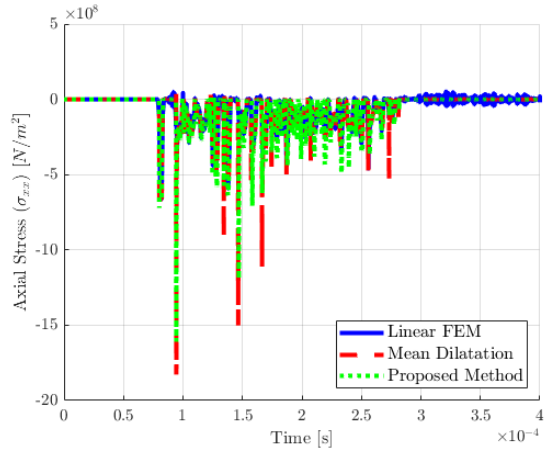
To investigate the comparison of the three different methods further the local time histories of the velocity, axial stress and displacement were plotted at initial position $\boldsymbol{X} = [32.4, 0]^T$mm for Bar One and $\boldsymbol{X} = [40.4, 0]^T$mm for Bar Two (i.e the centre of the contact interface), the results of which are presented in Figure 7.11. From the local time histories for the velocity and displacement it can be observed that all three methods agree well with some local oscillations in the solution during contact due to separation and re-initiation of contact as the bars deform significantly. When comparing the axial stress it can be observed that the standard linear FEM violates the contact conditions with oscillations after separation, as seen in Section 7.2.



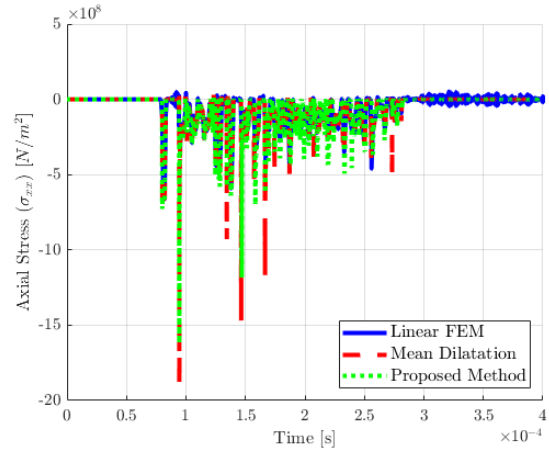
(a) Velocity v_x - Bar One



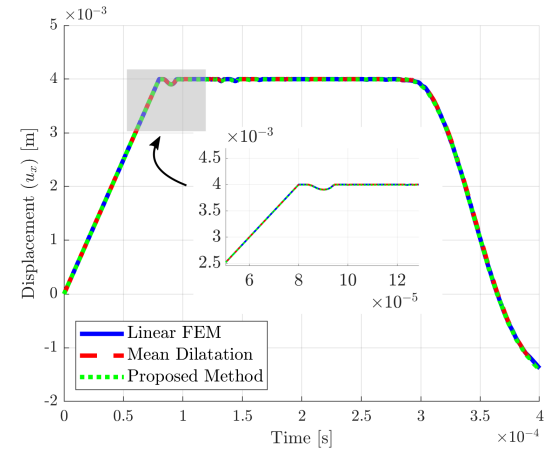
(b) Velocity v_x - Bar Two



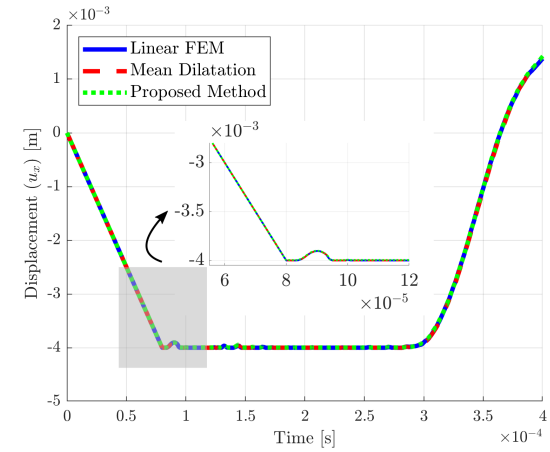
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two



(e) Displacement u_x - Bar One



(f) Displacement u_x - Bar Two

Figure 7.11: Nearly Incompressible Bar Impact - Comparison of local time histories at position $X = [32.4, 0]^T$mm for Bar One and $X = [40.4, 0]^T$mm for Bar Two

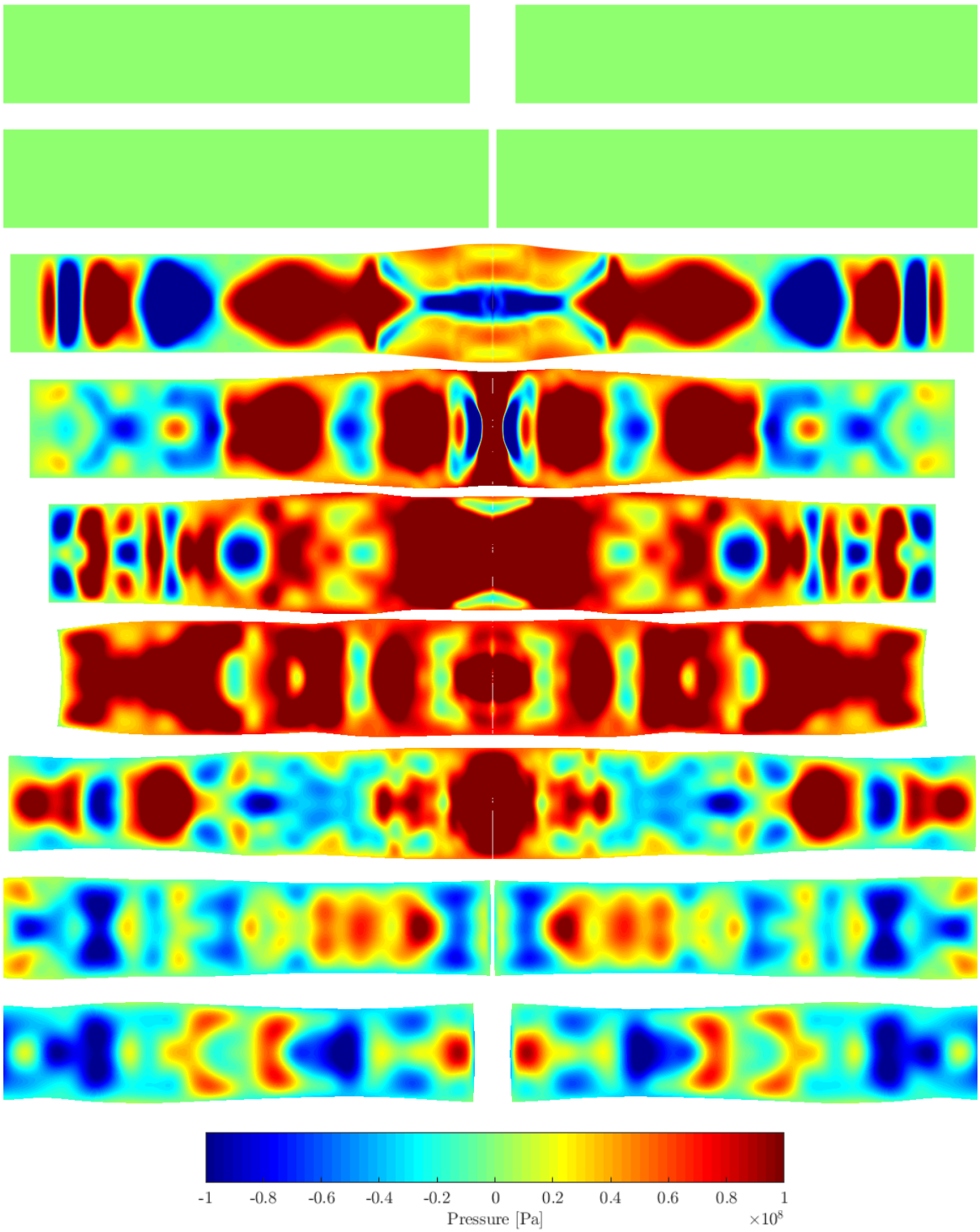


Figure 7.12: Nearly Incompressible Bar Impact - Pressure contour time evolution for Mesh 4 at 50 μs, 75 μs, 100 μs, 125 μs, 150μs, 200μs,250μs, 300μs and 325μs

7.4 Three-Dimensional Taylor Bar Impact

7.4.1 Problem Description

The next problem considered was that of a three-dimensional bar impacting a rigid surface. This problem is typically known as the Taylor bar impact [155] and is used as a benchmark test case for plasticity to ensure the absence of locking. The bar has length L of 32.4 mm, diameter D of 6.4 mm and initial velocity $\boldsymbol{v_0}$ of $[0, -227, 0]^T$ m/s as shown in Figure 7.13. The purpose of this problem is to benchmark single body contact using the purpose built OpenFOAM solver developed as part of this research against previously published results for various computational methods, including the previous vertex centred solver using the first order framework presented in [114] which was implemented in MATLAB. To compare these results the bar was modelled as a quarter of the domain using appropriate symmetric boundary conditions and considered in contact at $t = 0$ μs with material properties shown in Table 7.3.

Table 7.3: Taylor Bar Impact - Material parameters

Constitutive model		von-Mises Plasticity	
Young's modulus	E_0	117	$[\mathrm{GN/m^2}]$
Material density	ρ_0	8930	$[\mathrm{kg/m^3}]$
Poisson's ratio	ν	0.35	
Lamé parameters	μ	43.333	$[\mathrm{GN/m^2}]$
	λ	101.111	$[\mathrm{GN/m^2}]$
Yield stress	$\sigma_{y,0}$	0.4	$[\mathrm{GN/m^2}]$
Hardening parameter	H	0.1	$[\mathrm{GN/m^2}]$
Friction coefficient	k	0	

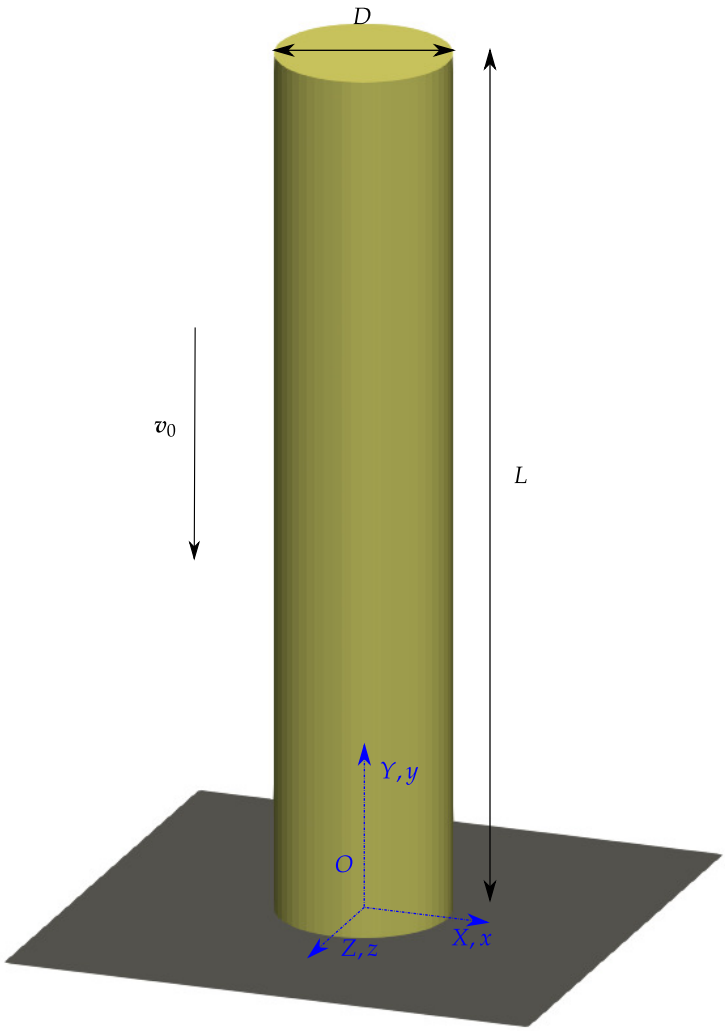


Figure 7.13: Taylor Bar Impact - Problem description

The problem was simulated for three different structured tetrahedral meshes to investigate the mesh sensitivity and to compare against other computational methods. These meshes were as follows: Mesh 1 had 1887 nodes and 8100 elements; Mesh 2 had 9191 nodes and 45000 elements; Mesh 3 had 25519 nodes and 132300 elements. All simulations were conducted with a CFL number of 0.3.

7.4.2 Numerical Results

Figure 7.14 shows the radius evolution for initial position $\boldsymbol{X} = [0, 0, 3.2]^T$mm for each mesh using the proposed method implemented in OpenFOAM. It is clear that as the mesh density increases the results tend towards the expected final position of approximately 7mm [155]. This final position can be compared with that of other computational methods summarised in Table 7.4.

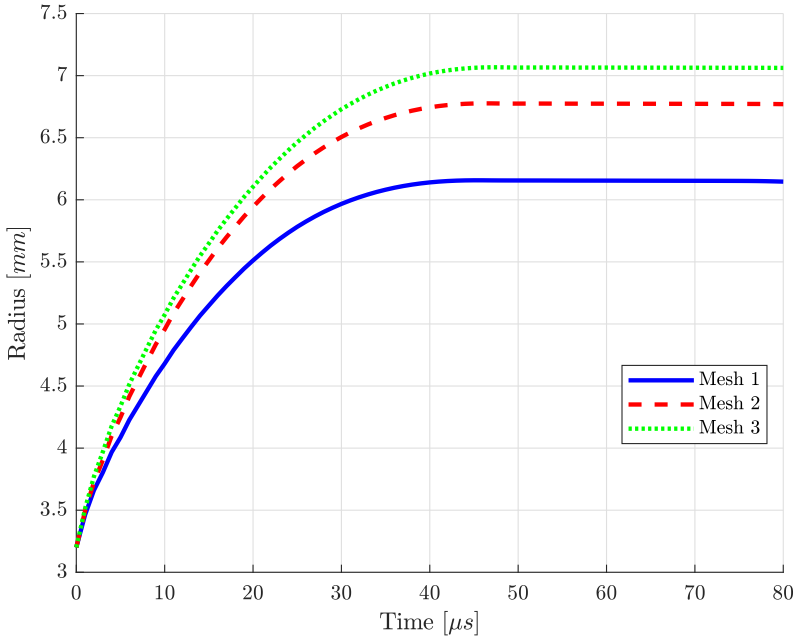
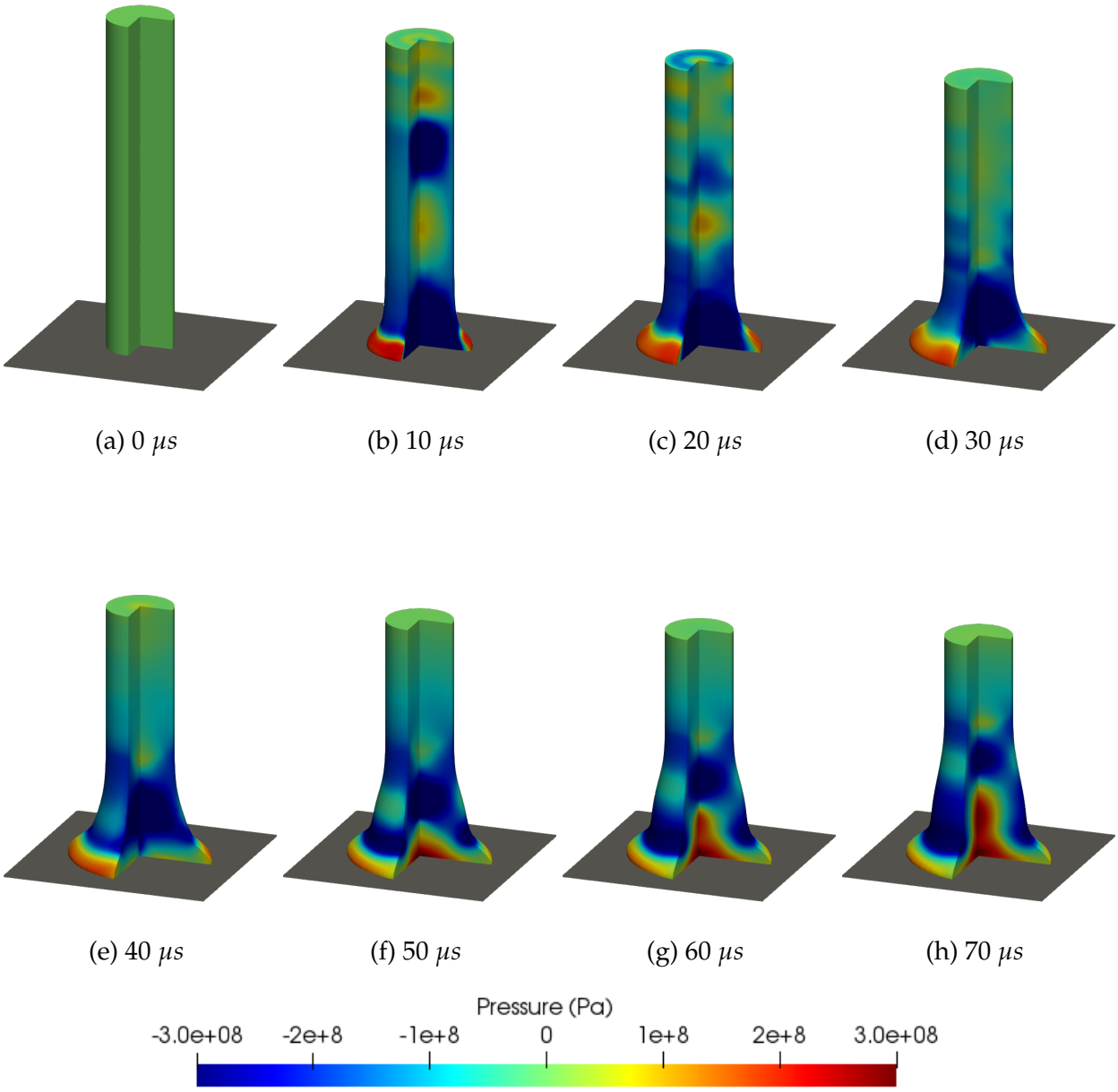


Figure 7.14: Taylor Bar Impact - Final Radius at initial position $\boldsymbol{X} = [0, 0, 3.2]^T$mm

Table 7.4: Taylor Bar Impact - Summary of Final Radius for Various Computational Methods

Method	**Mesh Type**	**Final Radius** $[mm]$
Standard FEM [34]	Tetrahedral	5.55
Standard FEM [34]	Hexahedral	6.95
Average Nodal Pressure FEM [34]	Tetrahedral	6.99
First Order Framework CCFVM [112]	Hexahedral	$6.88 - 7.11$
First Order Framework VCFVM (MATLAB) [114]	Tetrahedral	≈ 7.1
First Order Framework VCFVM (OpenFOAM)	Tetrahedral	7.06

From the results in Table 7.4, it can be observed that the results using standard FEM with a tetrahedral mesh experiences locking resulting in a lower final radius compared with other methods. The proposed method in OpenFOAM, on the other hand, matches closely to those presented using the same methodology in MATLAB [114] and agrees well with previous results presented for locking free methodologies [34, 112]. To demonstrate clearly that the proposed method in OpenFOAM is locking free with a smooth pressure contour, the results for the time evolution are presented in Figure 7.15. Now that the proposed method has been benchmarked in OpenFOAM, more advanced problems can be considered.



(a) 0 μs (b) 10 μs (c) 20 μs (d) 30 μs

(e) 40 μs (f) 50 μs (g) 60 μs (h) 70 μs

Figure 7.15: Taylor Bar Impact - Pressure Time Evolution

7.5 Torus Floor Impact

7.5.1 Problem Description

The next problem considered a three-dimensional torus impacting a rigid surface as depicted in Figure 7.16. The torus had outer radius R_o of 40mm, inner radius r_i of 30mm and diameter d_0 of 1mm. The torus was travelling with initial velocity v_0 of $[1.18, 0, 0]^T$m/s towards the rigid surface with initial gap δ_n of 4mm. The problem was simulated using two different constitutive models, first was an elastic model, nearly Incompressible neo-Hookean, then the problem was repeated using a plastic constitutive model, von-Misies plasticity with isotropic hardening. The material properties for both cases are shown in Table 7.5. All simulations were conducted using a CFL number of 0.3.

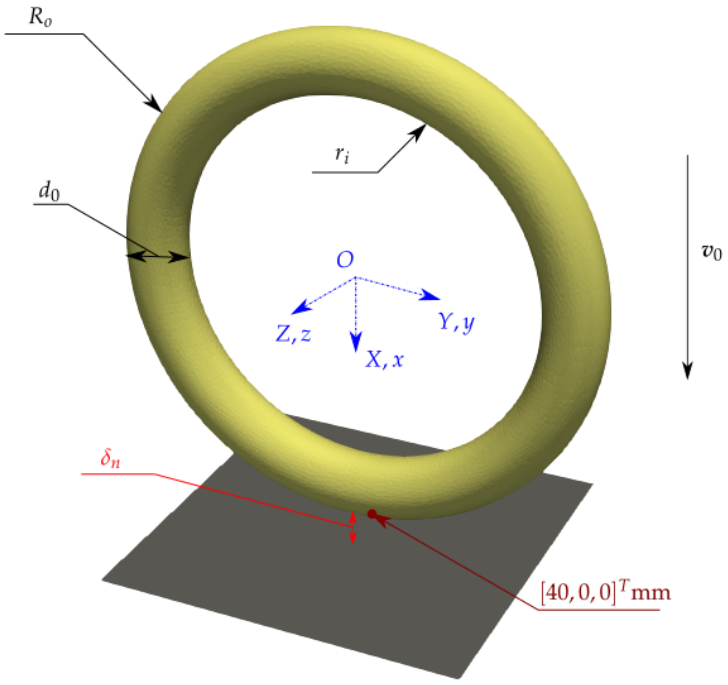


Figure 7.16: Torus Impact - Problem description

Table 7.5: Torus Impact - Material parameters

		Elastic torus	Plastic torus	
Constitutive Model		neo-Hookean	von-Mises Plasticity	
Young's Modulus	E_0	$1 \cdot 10^6$	$1 \cdot 10^6$	$[N/m^2]$
Material density	ρ_0	1000	1000	$[kg/m^3]$
Poisson Ratio	ν	0.45	0.45	
Lamé parameters	μ	0.34483	0.34483	$[MN/m^2]$
	λ	3.10345	3.10345	$[MN/m^2]$
Yield stress	$\sigma_{y,0}$	-	$1 \cdot 10^4$	$[N/m^2]$
Hardening parameter	H	-	10	$[N/m^2]$
Friction coefficient	k	0	0	

7.5.2 Numerical Results

Global energy solution

First, to demonstrate consistency of the purpose built OpenFOAM solver, mesh sensitivity study was conducted using the same meshes in both material models with the global energy evolutions shown in Figure 7.17. The meshes used are unstructured tetrahedral meshes where Mesh 1 consisted of 3545 nodes and 12439 elements, Mesh 2 consisted of 7606 nodes and 29748 elements, Mesh 3 consisted of 13497 nodes and 56955 elements and Mesh 4 consisted of 33744 nodes and 155300 elements.

From the results in Figure 7.17a-7.17d it is clear that as the number of nodes and elements increases the accuracy of the simulation in both elastic and plastic cases improves, demonstrated by the increase in total system energy and reduction in numerical error. By examining the results in Figures 7.17e and 7.17f the total system behaviour for both cases can be described. In both cases, at the start of the simulation all energy exists as kinetic energy as the torus is moving with the initial velocity until approximately 3.4ms where impact occurs. In the elastic case (Figure 7.17e) the kinetic energy is mostly transferred into elastic potential energy, as demonstrated by the red line until approximately 30ms when separation begins to occur and the potential energy is mostly converted back to kinetic energy as the torus bounces off the rigid surface. For the plastic case (Figure 7.17f) at impact the kinetic energy is partially converted into potential energy (red line) while most of the kinetic energy is transferred into plastic dissipation (black line) as a result the torus becomes nearly fully plastic and there is no separation during the 50ms simulation. This is reiterated by examining the pressure contour time evolutions in Figures 7.20 for the elastic torus and Figure 7.21 for the

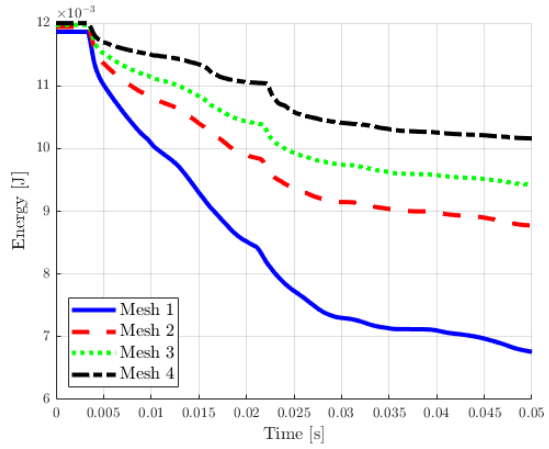
plastic torus.

Global pressure contour comparison

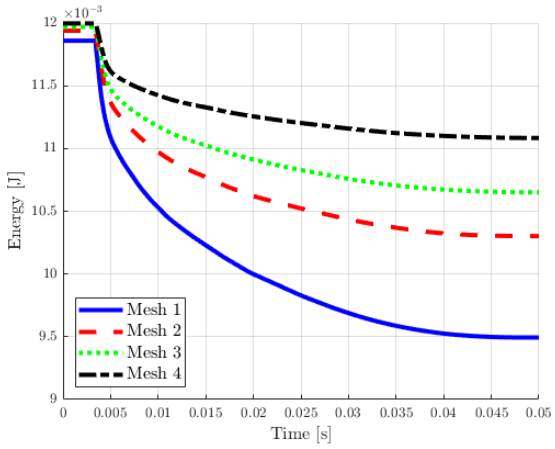
Figures 7.18 and 7.19 show the comparison of pressure contour snapshots for the three different methods. Each figure compares the proposed method using a tetrahedral mesh against the linear FEM with tetrahedral mesh (20427 Nodes, 101045 Elements) and mean dilatation FEM with hexahedral mesh (33793 Nodes, 29441 Elements). It is clear from these snapshots that the linear FEM experiences pressure checker boarding in both elastic and plastic cases while the proposed method does not, matching closely with the mean dilatation results. For completeness, Figures 7.20 and 7.21 show the time evolutions for the elastic and plastic scenarios of the pressure contours respectively.

Local Results

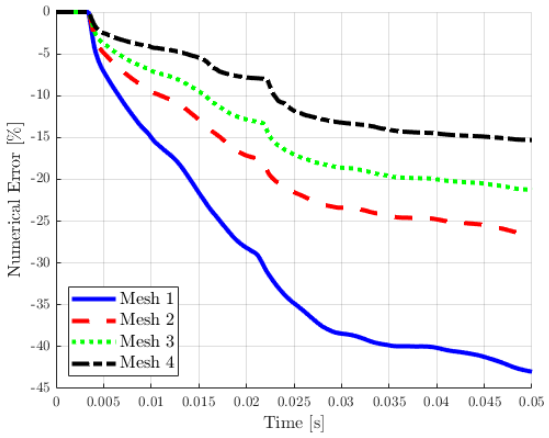
The last results examined in this section are the local results taken at initial position $\boldsymbol{X} = [40, 0, 0]^T$mm for each material model. The local results for the elastic and plastic torus simulations are presented in Figure 7.22. By examining the local results of both the elastic and plastic torus it can be observed that there is good agreement between the proposed method and the mean dilatation FEM solutions with no spurious oscillations while the linear FEM solution clearly has spurious oscillations in the axial stress solutions of both material models and experiences locking when examining the velocity (Figure 7.22b) and displacement (Figure 7.22d) time evolutions. While the proposed method shows good agreement with the mean dilatation FEM in the elastic case, the solutions differ slightly suggesting the proposed method solution has not yet converged requiring mesh refinement.



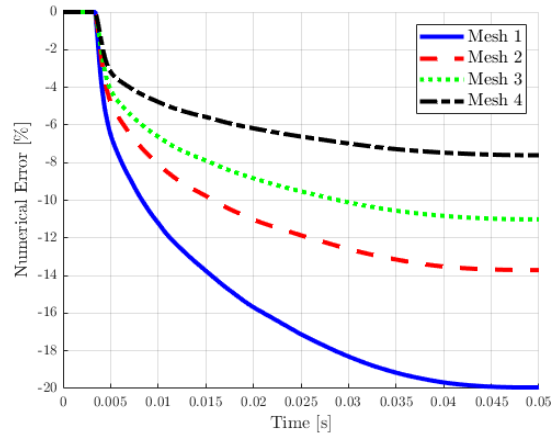
(a) Total system energy - Mesh refinement



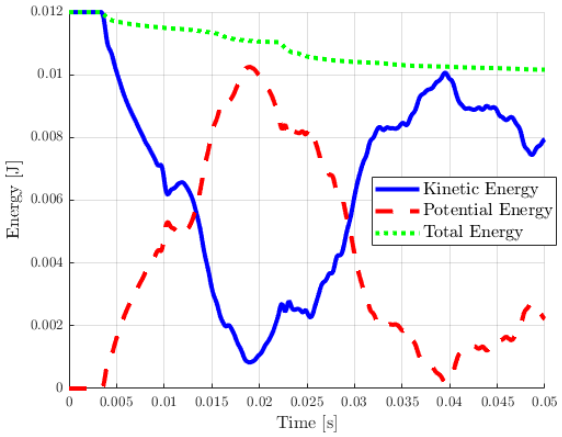
(b) Total system energy - Mesh refinement



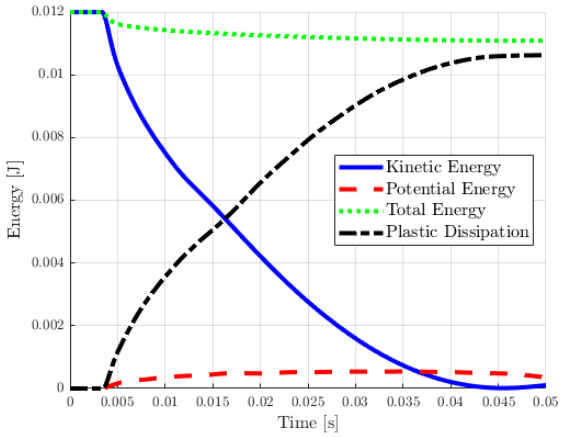
(c) Numerical error - Mesh refinement



(d) Numerical error - Mesh refinement



(e) Energy history (Mesh 4)



(f) Energy history (Mesh 4)

Figure 7.17: Torus Floor Impact - Elastic (left) and plastic (right) torus impact global energy evolutions

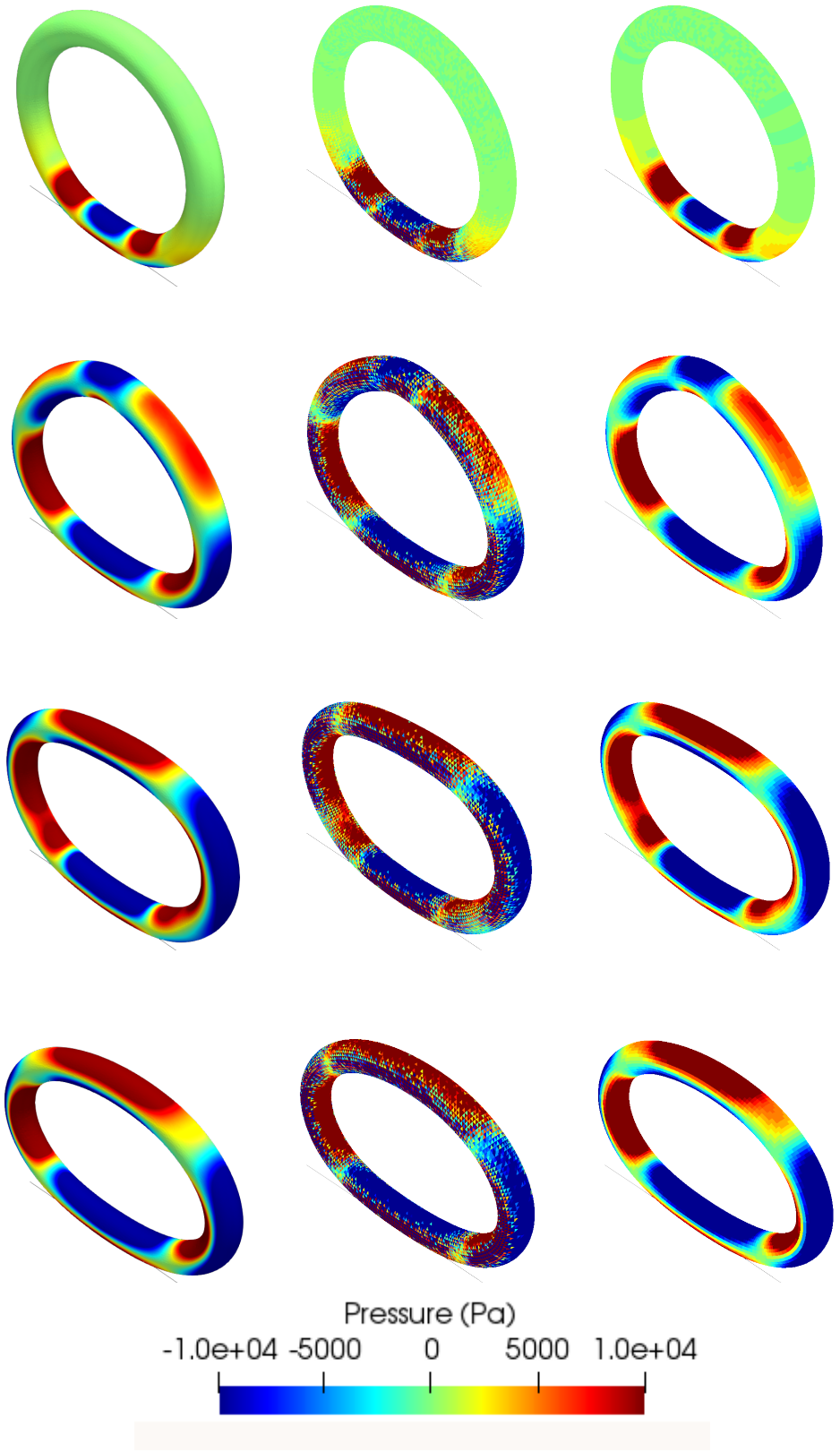


Figure 7.18: Torus Impact - Time evolution comparison of the elastic torus for the proposed method with a tetrahedral mesh (left), linear FEM with a tetrahedral mesh (centre) and mean dilatation FEM with a hexahedral mesh (right) at 0.5ms, 1ms, 1.5ms, 2ms

Figure 7.19: Torus Floor Impact - Time evolution comparison of the plastic torus for the proposed method with a tetrahedral mesh (left), linear FEM with a tetrahedral mesh (centre) and mean dilatation FEM with a hexahedral mesh (right) at 0.5ms, 1ms, 1.5ms, 2ms

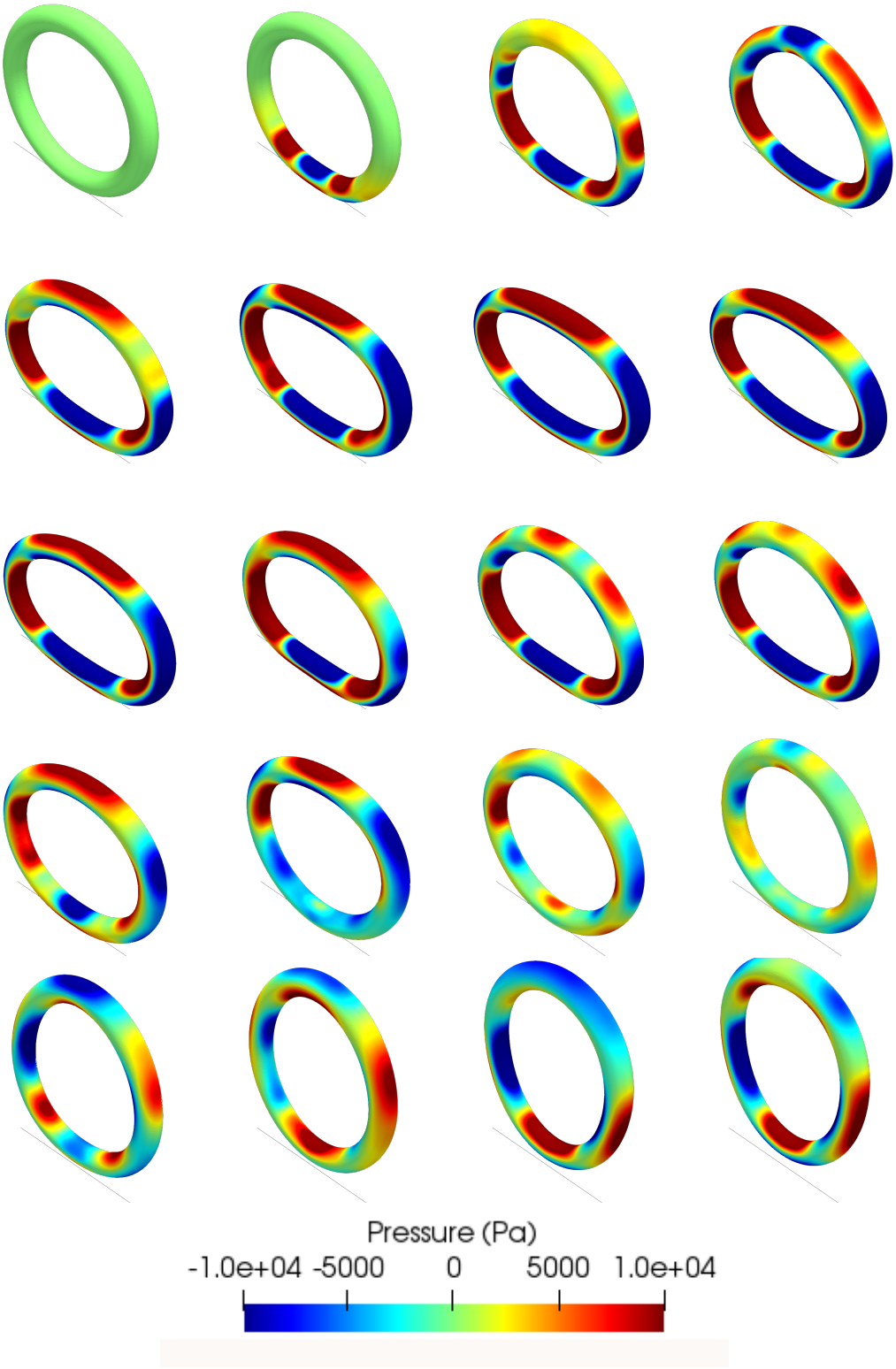


Figure 7.20: Torus Impact - Time evolution of the elastic torus for the proposed method with a tetrahedral mesh at 2.5ms, 5ms, ..., 47.5ms, 50ms

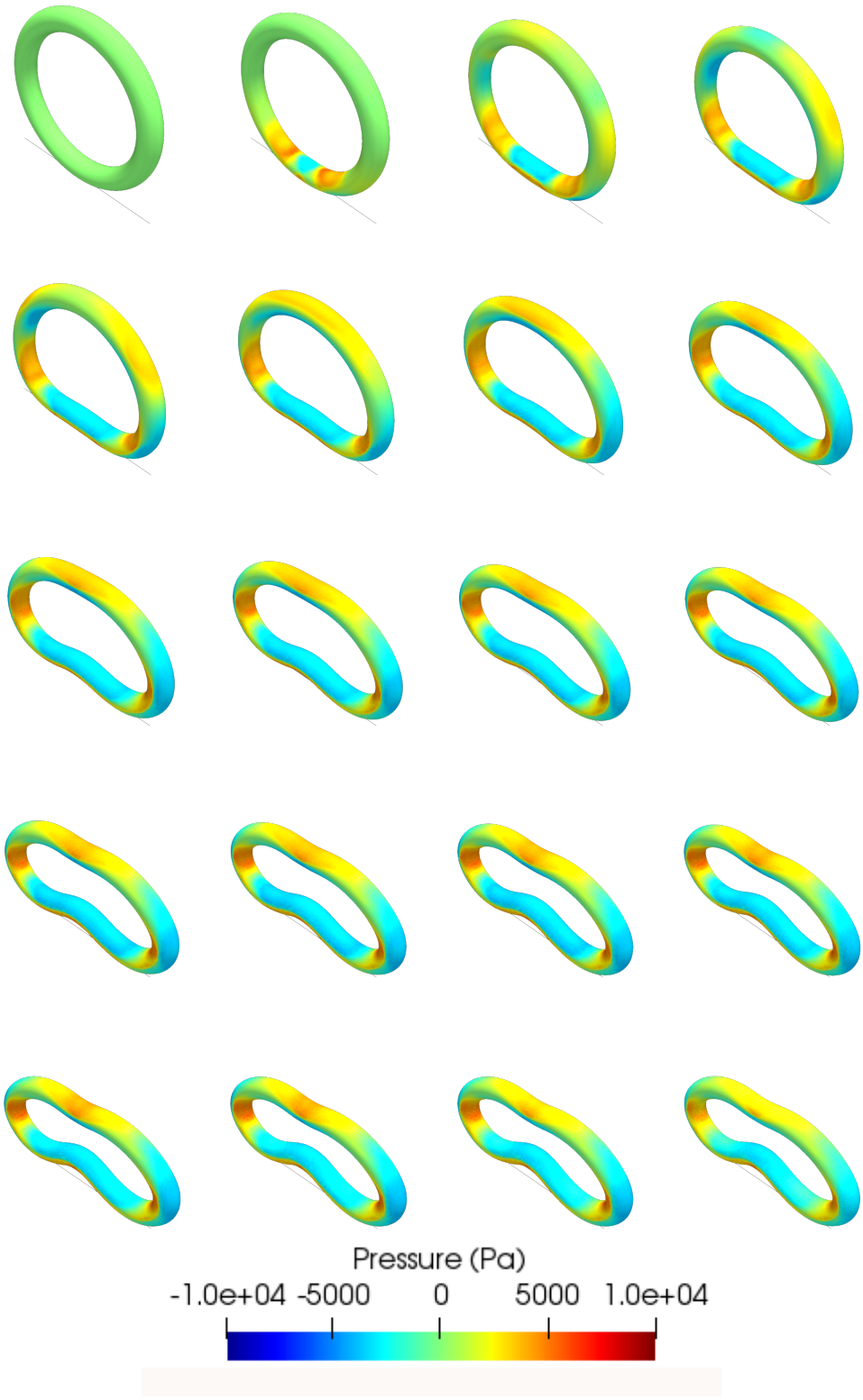
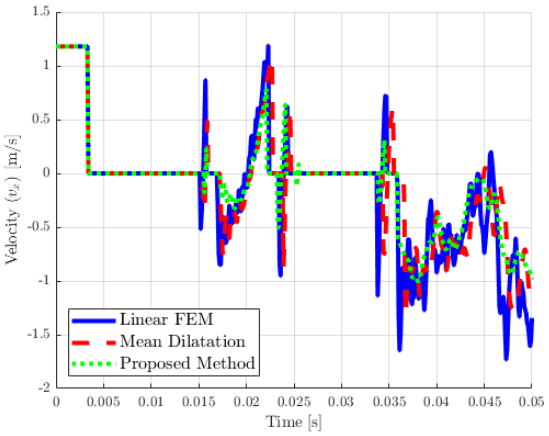
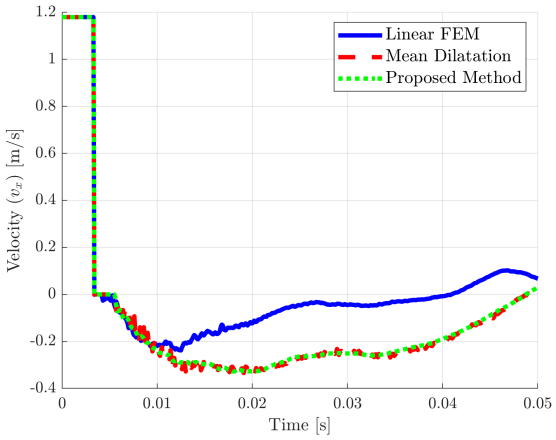


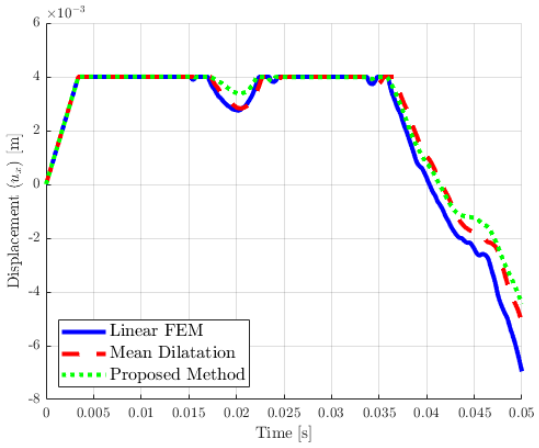
Figure 7.21: Torus Floor Impact - Time evolution of the plastic torus for the proposed method with a tetrahedral mesh at 2.5ms, 5ms, ..., 47.5ms, 50ms



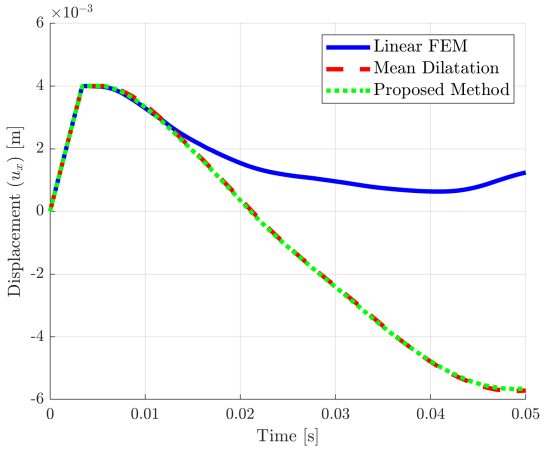
(a) Velocity v_x



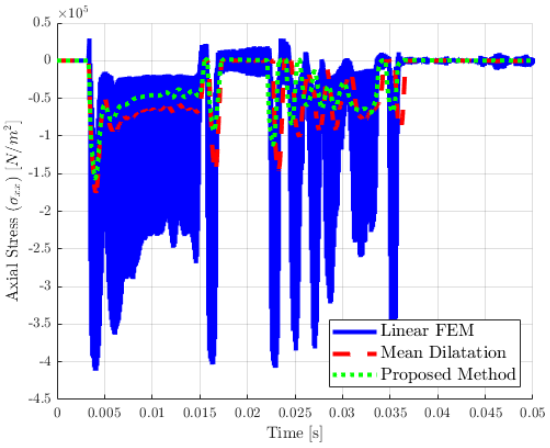
(b) Velocity v_x



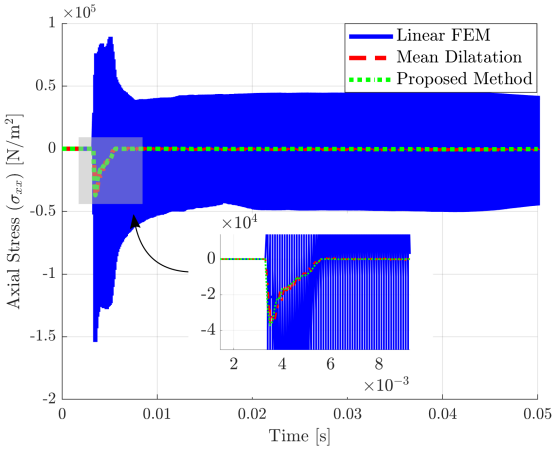
(c) Displacement u_x



(d) Displacement u_x



(e) Axial stress σ_{xx}



(f) Axial stress σ_{xx}

Figure 7.22: Torus Floor Impact - Comparison of local results of elastic torus (left) and plastic torus (right) taken at $\boldsymbol{X} = [40, 0, 0]^T$mm for the proposed method (tetrahedral mesh), Linear FEM (tetrahedral mesh) and mean dilatation (hexahedral mesh)

7.6 Non-Matching Nearly Incompressible Bar Impact

7.6.1 Problem Description

This problem considered the impact of two three-dimensional rectangular bars as depicted in Figure 7.23, similar to that presented in Section 7.3 for two dimensions. Both bars have width w and height h of 3.2mm with a length L of 32.4mm. The bars were travelling with initial velocity $\boldsymbol{v}_0$ of $[50, 0, 0]^T$m/s and $[-50, 0, 0]^T$m/s respectively with initial gap δ_n of 0.4mm. The objective of this problem was to investigate the accuracy of the proposed method when considering a non-matching contact interface. All simulations were conducted using a CFL number of 0.3.

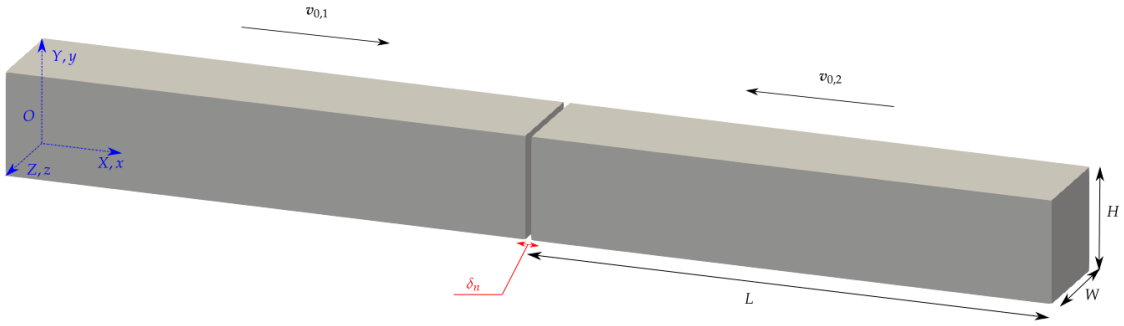


Figure 7.23: Non-Matching Nearly Incompressible Bars - Problem description

The problem was simulated for two scenarios, the first scenario was where both bars had identical meshes resulting in a matching (conforming) contact interface and then the second scenario was where one bar had a different mesh resulting in a non-matching (non-conforming) contact interface. The material properties for the bars are in Table 7.6.

Table 7.6: Non-Matching Nearly Incompressible Bars - Material parameters

Constitutive Model		neo-Hookean	
Young's Modulus	E_0	$5.85 \cdot 10^8$	$[\mathrm{N/m^2}]$
Material density	ρ_0	8930	$[\mathrm{kg/m^3}]$
Poisson's ratio	ν	0.495	
Lamé parameters	μ	0.35714	$[\mathrm{MN/m^2}]$
	λ	1.42857	$[\mathrm{MN/m^2}]$
Friction coefficient	k	0	

To accurately investigate the accuracy of the non-matching contact interfaces, two sets

of mesh refinements were conducted. First, four matching meshes increasing in mesh density were analysed, these consist of structured tetrahedral meshes where the contact interface for Mesh 1 is 4 by 4 elements, Mesh 2 is 6 by 6 elements, Mesh 3 is 8 by 8 elements and Mesh 4 is 10 by 10 elements. Following this analysis four non-matching meshes were analysed where Bar One is a constant mesh size with contact interface of 10 by 10 elements, while Bar Two has increasing elements at the contact interface starting with Mesh 1 4 by 4 elements, Mesh 2 with 6 by 6 elements, Mesh 3 with 8 by 8 elements then Mesh 4 with 12 by 12 elements. The resulting number of nodes and elements for each bar as summarised in Table 7.7 for the matching meshes and Table 7.8 for the non-matching meshes with a comparison of the different meshes shown in Figure 7.24.

Table 7.7: Non-Matching Nearly Incompressible Bars - Matching contact interface mesh information

	Bar One		Bar Two	
	Nodes	Elements	Nodes	Elements
Mesh 1	525	1920	525	1920
Mesh 2	1519	6480	1519	6480
Mesh 3	3321	15360	3321	15360
Mesh 4	6171	30000	6171	30000

Table 7.8: Non-Matching Nearly Incompressible Bars - Non-matching contact interface mesh information

	Bar One		Bar Two	
	Nodes	Elements	Nodes	Elements
Mesh 1	6171	30000	525	1920
Mesh 2	6171	30000	1519	6480
Mesh 3	6171	30000	3321	15360
Mesh 4	6171	30000	10309	51840

7.6.2 Numerical Results

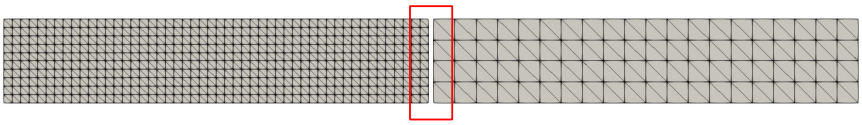
The global energy results for the matching contact interface using the meshes in Table 7.7 are shown in Figure 7.25. Similarly for the global energy results for a non-matching contact interface using the meshes in Table 7.8 are shown in Figure 7.26.

The results in both Figure 7.25 and 7.26 demonstrate the consistency of the proposed method as the mesh density increases the global energy improves and the numerical error decreases. From the individual energy plots for each bar, the simulation can clearly be described through the evolution of kinetic and potential energies. Both bodies initially are travelling at their respective initial velocity and are completely undeformed resulting in all energy existing as kinetic energy (blue). The bars then contact at approximately 5μs this shock results in significant elastic deformation as demonstrated by the transfer of kinetic energy to potential energy (red) producing an internal pressure wave. Once the pressure wave reflects off the free end of the bars and returns to the contact interface the bars separate at approximately 245μs, where a proportion of the elastic potential energy is converted back to kinetic energy and the bars will travel in opposite directions away from each other. From the figures it can be seen that not all elastic potential energy is transferred to kinetic energy at separation resulting in elastic potential and kinetic energy oscillations, as the bar expands and contracts after separation. Through comparing the results of the matching and non-matching meshes the results agree well for the various levels of conformity of the contact interfaces. To visually demonstrate this agreement, the pressure contour snapshots comparing matching contact interface Mesh 4 against each non-matching mesh is presented in Figure 7.27 at $t = 120\mu$s and in Figure 7.28 at $t = 260\mu$s.

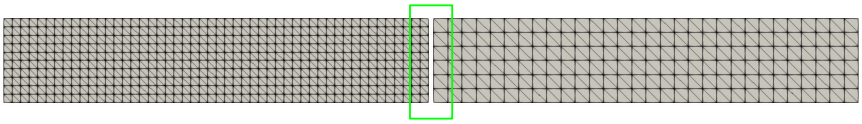
To further examine these results for non-matching meshes, the local nodal time evolutions taken at the centre of the contact interface in terms of global coordinates from Figure 7.23 for Bar One, $\boldsymbol{X} = [32.4, 3.2, 3.2]^T$mm and for Bar Two, $\boldsymbol{X} = [32.8, 3.2, 3.2]^T$mm were examined. The results for both bars are shown in Figure 7.29, comparing matching Mesh 4 against non-matching Mesh 1 to 4. By examining Figure 7.29 it can be observed that for the local results for Bar One, all five simulation results match closely. This was to be expected as the mesh for this bar is identical for all five simulations therefore the slight difference in results is produced purely as result of the mesh density of Bar Two interpolated to Bar One. This is clearly observed in the results for Bar Two as the non-matching Mesh 1 (red) results produce the results that differ the most from the other simulations due to this mesh having the lowest number of nodes and elements. While the matching Mesh 4 and non-matching Mesh 4 have results that match closely and only differ in resolution as these have a similar mesh density. To further examine the difference in the solutions for the various meshes, the results along the centreline of Bar Two[1] ($\boldsymbol{X} = [32.8, 1.6, 1.6]^T to \boldsymbol{X} = [65.2, 1.6, 1.6]^T$) at $t = 260\mu$s are shown in Figure 7.30 demonstrating clear convergence of the meshes with mesh refinement along the length of the bar. For completeness the time evolutions of the non-matching contact interface using Mesh 1 are shown in Figure 7.31. Lastly with the proposed methods demonstrat-

[1]Only Bar Two results are presented here as the Bar One mesh remains constant for each simulation.

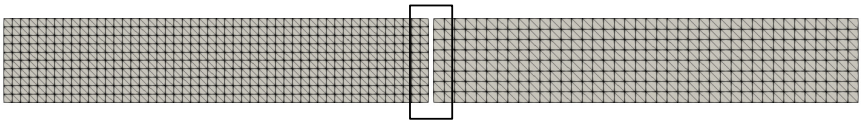
ing good agreement across various meshes, the global total energy from Figure 7.26 is compared in Figure 7.32 to an equivalent problem setup using linear FEM and mean dilatation. The linear FEM simulation used 30,778 elements and 6267 nodes for Bar One and 50,261 elements and 9976 nodes for Bar Two, while the mean dilatation simulation used 5,000 elements and 6171 nodes for Bar One and 8,640 elements and 10,309 nodes for Bar Two. It is observed that the total energy decreases over time and that the proposed method is more dissipative than mean dilatation with no spurious modes (pressure checker boarding) which are observed in the linear FEM pressure contour.



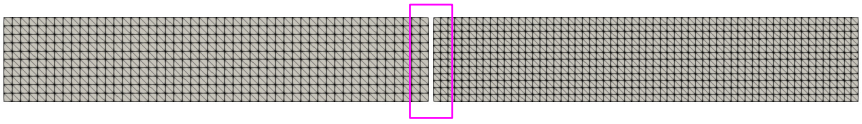
(a) Non-matching contact interface (Mesh 1)



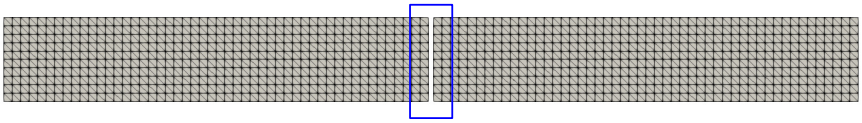
(b) Non-matching contact interface(Mesh 2)



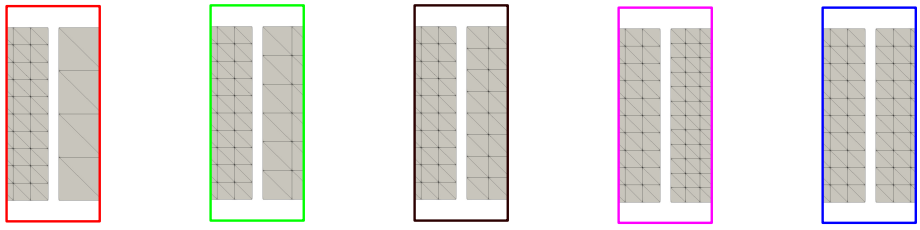
(c) Non-matching contact interface (Mesh 3)



(d) Non-matching contact interface (Mesh 4)

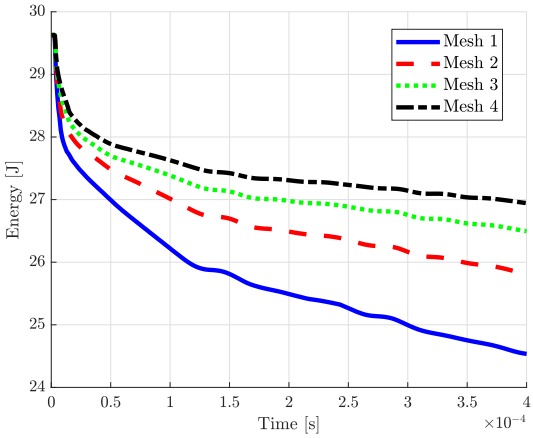


(e) Matching contact interface (Mesh 4)

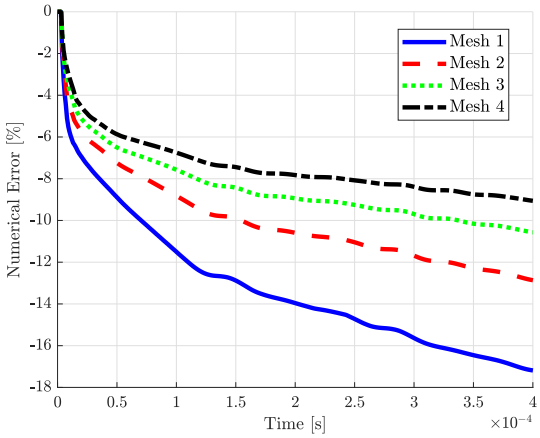


(f) Close up of each contact interface, from left to right: Non-matching contact interface (Mesh 1), Non-matching contact interface (Mesh 2), Non-matching contact interface (Mesh 3), Non-matching contact interface (Mesh 4), Matching contact interface (Mesh 4)

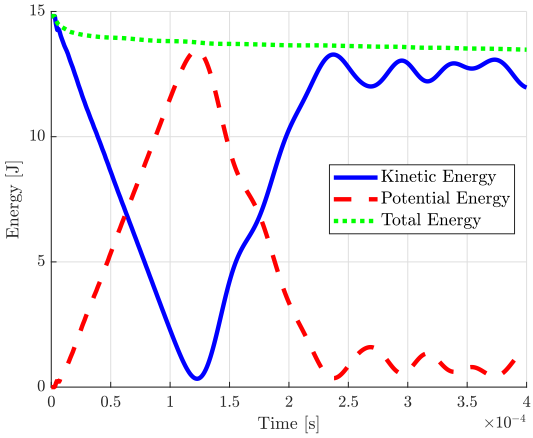
Figure 7.24: Non-Matching Nearly Incompressible Bars - Comparison of mesh contact interfaces



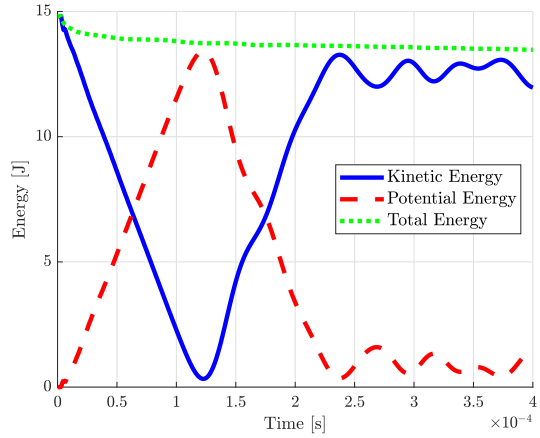
(a) Total system energy - Mesh refinement



(b) Numerical error - Mesh refinement

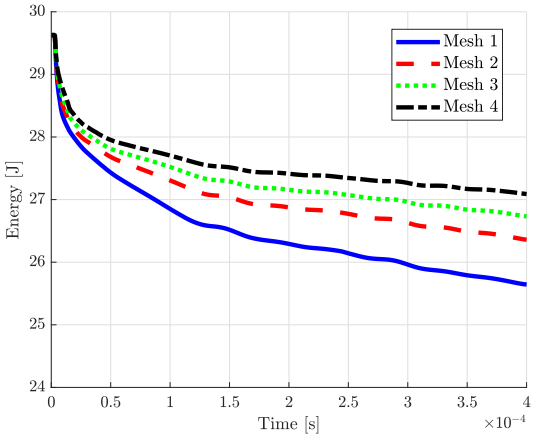


(c) Energy history - Bar One (Mesh 4)

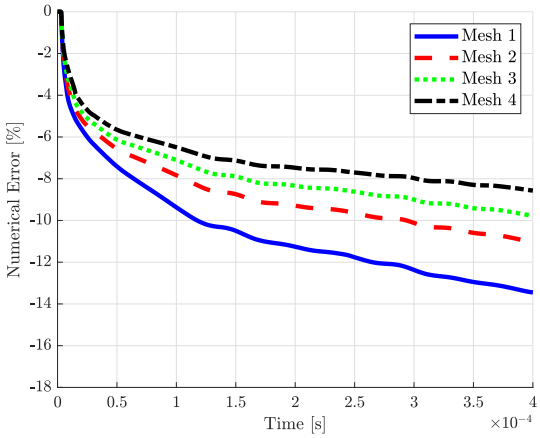


(d) Energy history - Bar Two (Mesh 4)

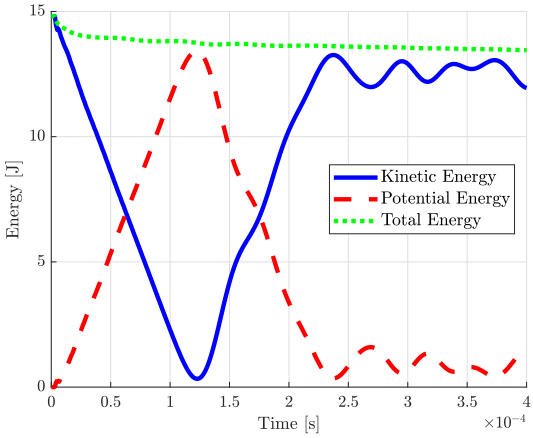
Figure 7.25: Non-Matching Nearly Incompressible Bars - Energy time history comparing matching contact interface Mesh 1-4 for the proposed method



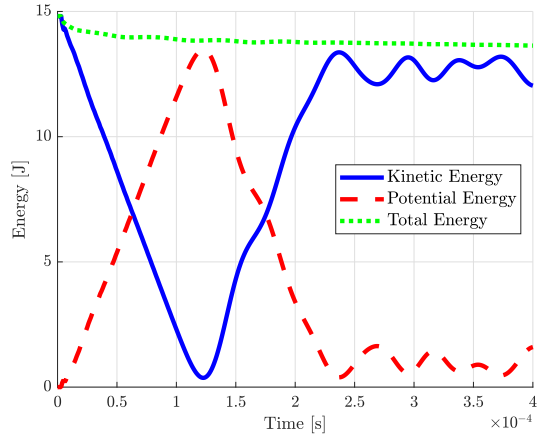
(a) Total system energy - Mesh refinement



(b) Numerical error - Mesh refinement



(c) Energy history - Bar One (Mesh 4)



(d) Energy history - Bar Two (Mesh 4)

Figure 7.26: Non-Matching Nearly Incompressible Bars - Energy time history comparing non-matching contact interface Mesh 1-4 for the proposed method

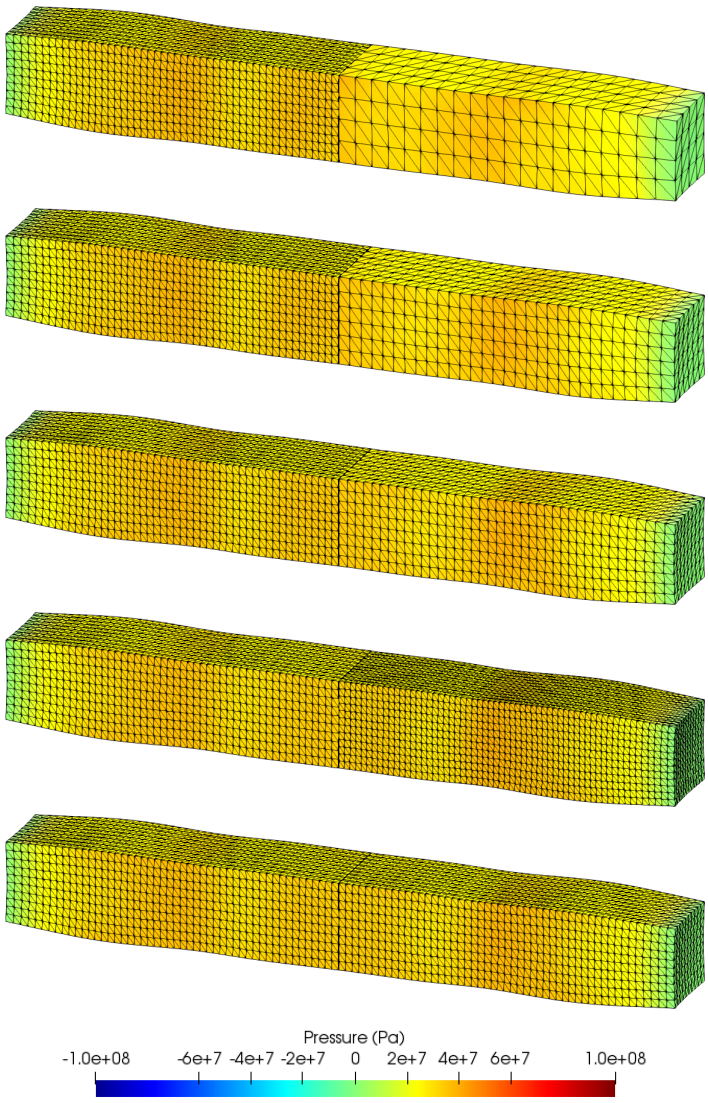


Figure 7.27: Non-Matching Nearly Incompressible Bars - Comparison of global pressure contours at 120μs for each mesh from top to bottom: non-matching Mesh 1, non-matching Mesh 2, non-matching Mesh 3, non-matching Mesh 4, matching Mesh 4

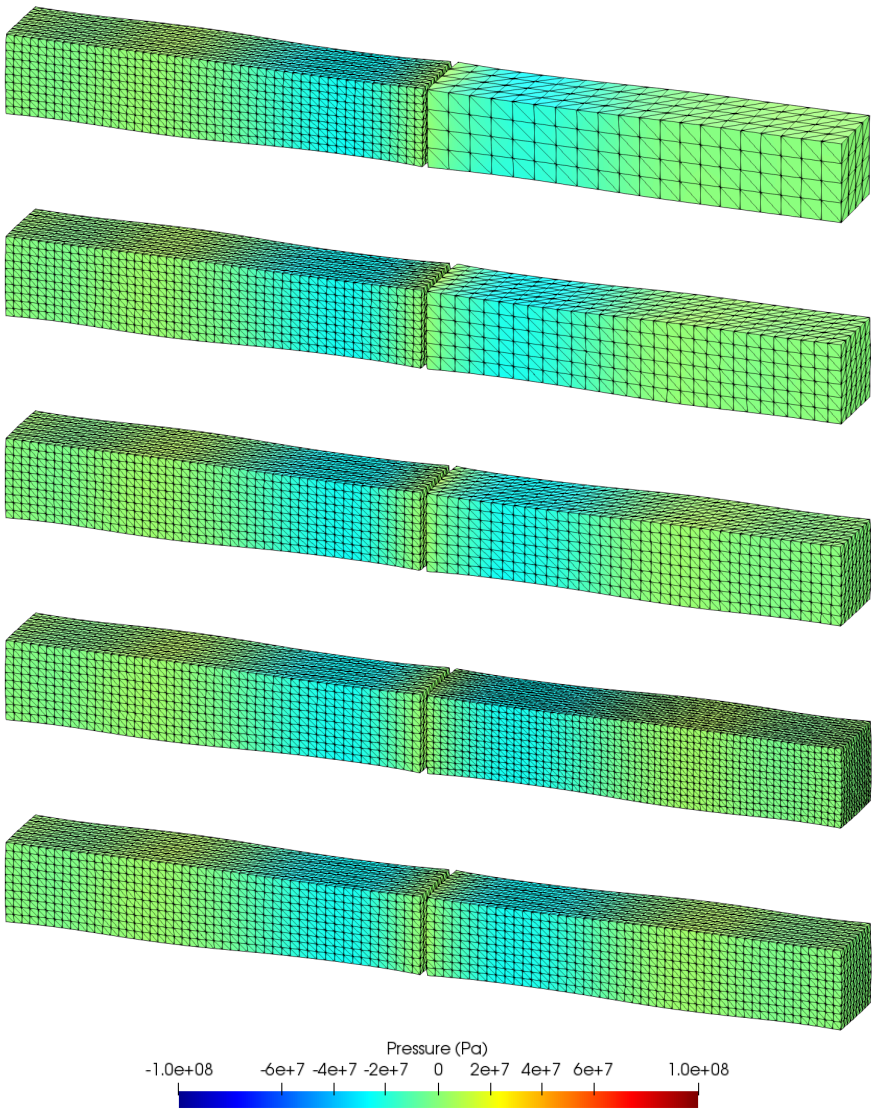
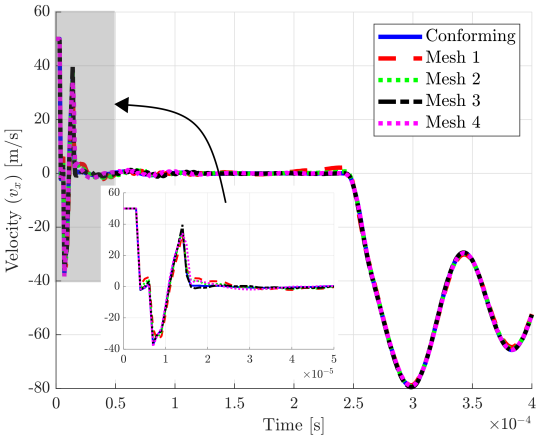
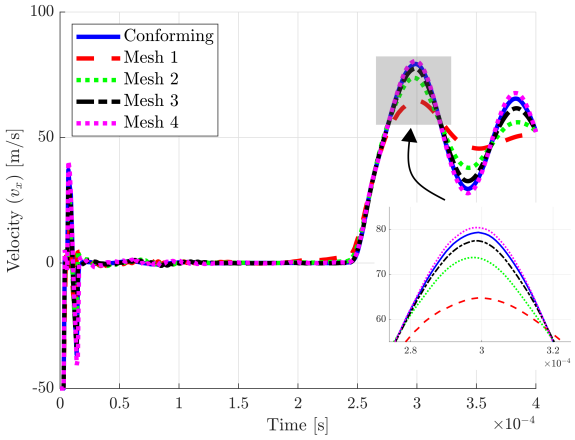


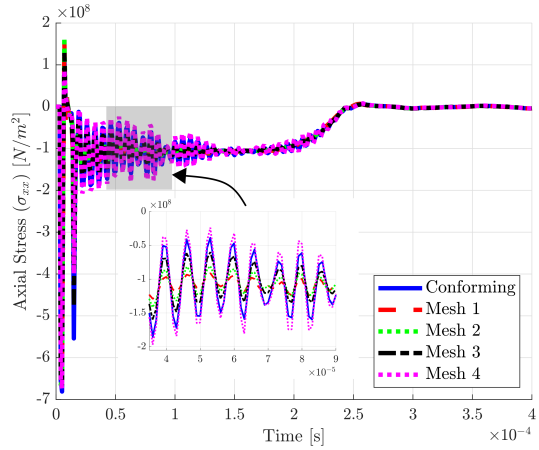
Figure 7.28: Non-Matching Nearly Incompressible Bars - Comparison of global pressure contours at 260μs for each mesh from top to bottom: non-matching Mesh 1, non-matching Mesh 2, non-matching Mesh 3, non-matching Mesh 4, matching Mesh 4



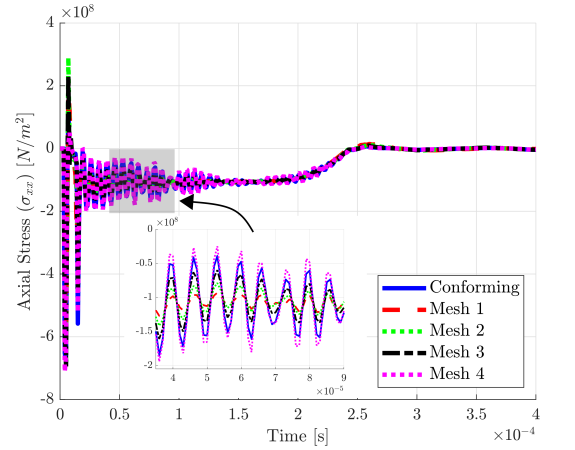
(a) Velocity v_x - Bar One



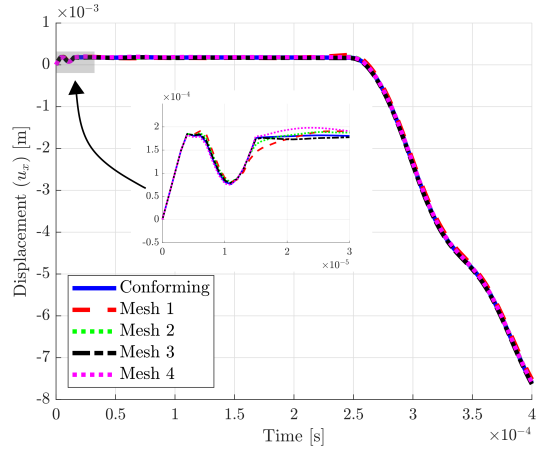
(b) Velocity v_x - Bar Two



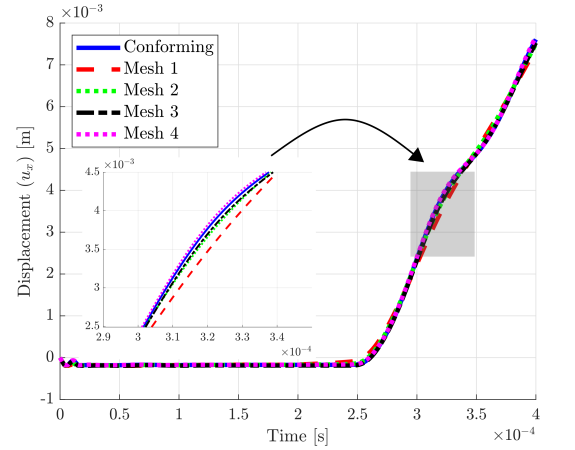
(c) Axial Stress σ_{xx} - Bar One



(d) Axial Stress σ_{xx} - Bar Two



(e) Axial Displacement u_x - Bar One



(f) Axial Displacement u_x - Bar Two

Figure 7.29: Non-Matching Nearly Incompressible Bars - Comparison of local results for matching Mesh 4 and non-matching Mesh 1 to 4

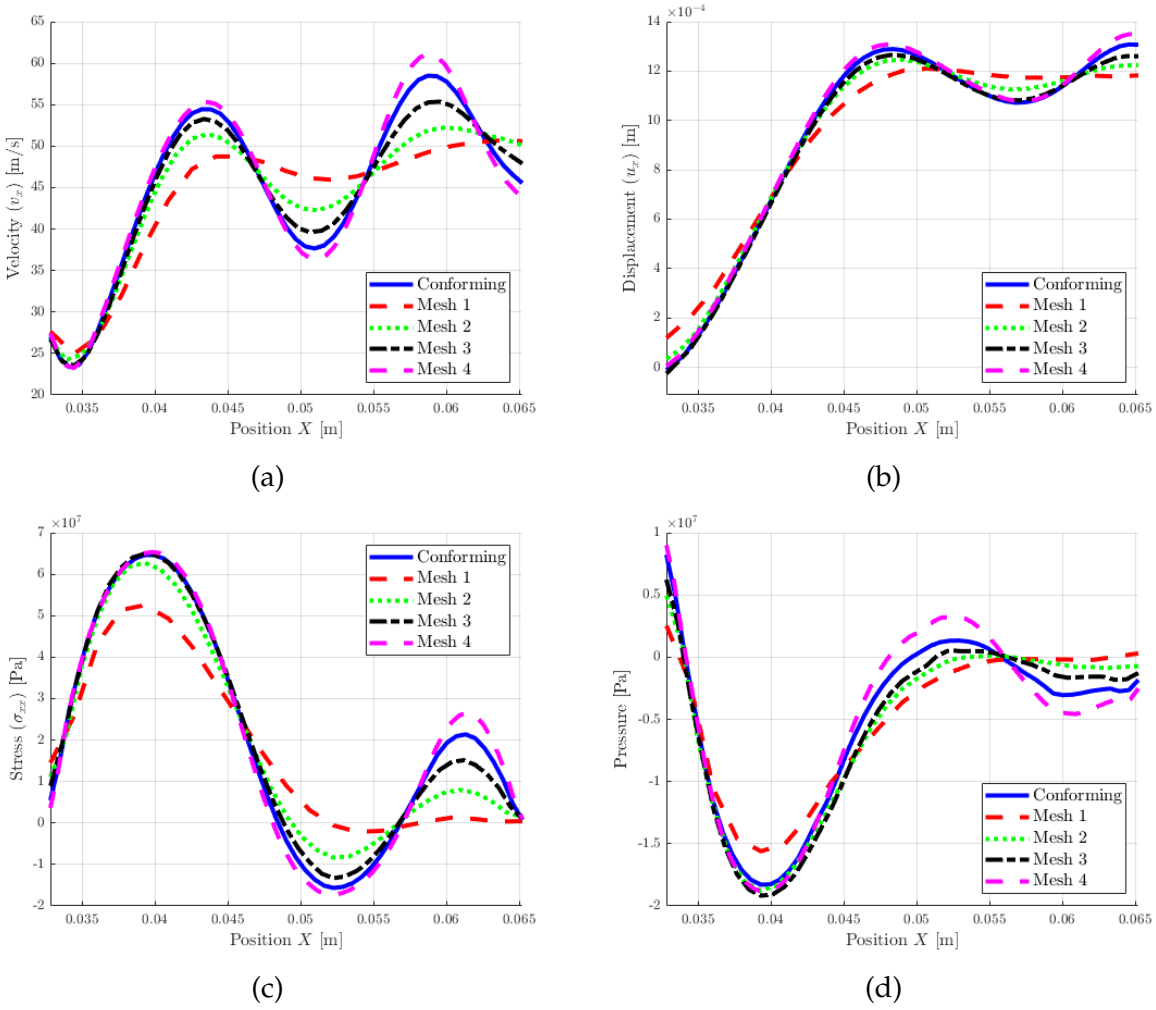


Figure 7.30: Non-Matching Nearly Incompressible Bars - Comparison of Meshes along centreline $\boldsymbol{X} = [32.8, 1.6, 1.6]^T to \boldsymbol{X} = [65.2, 1.6, 1.6]^T$ of Bar Two

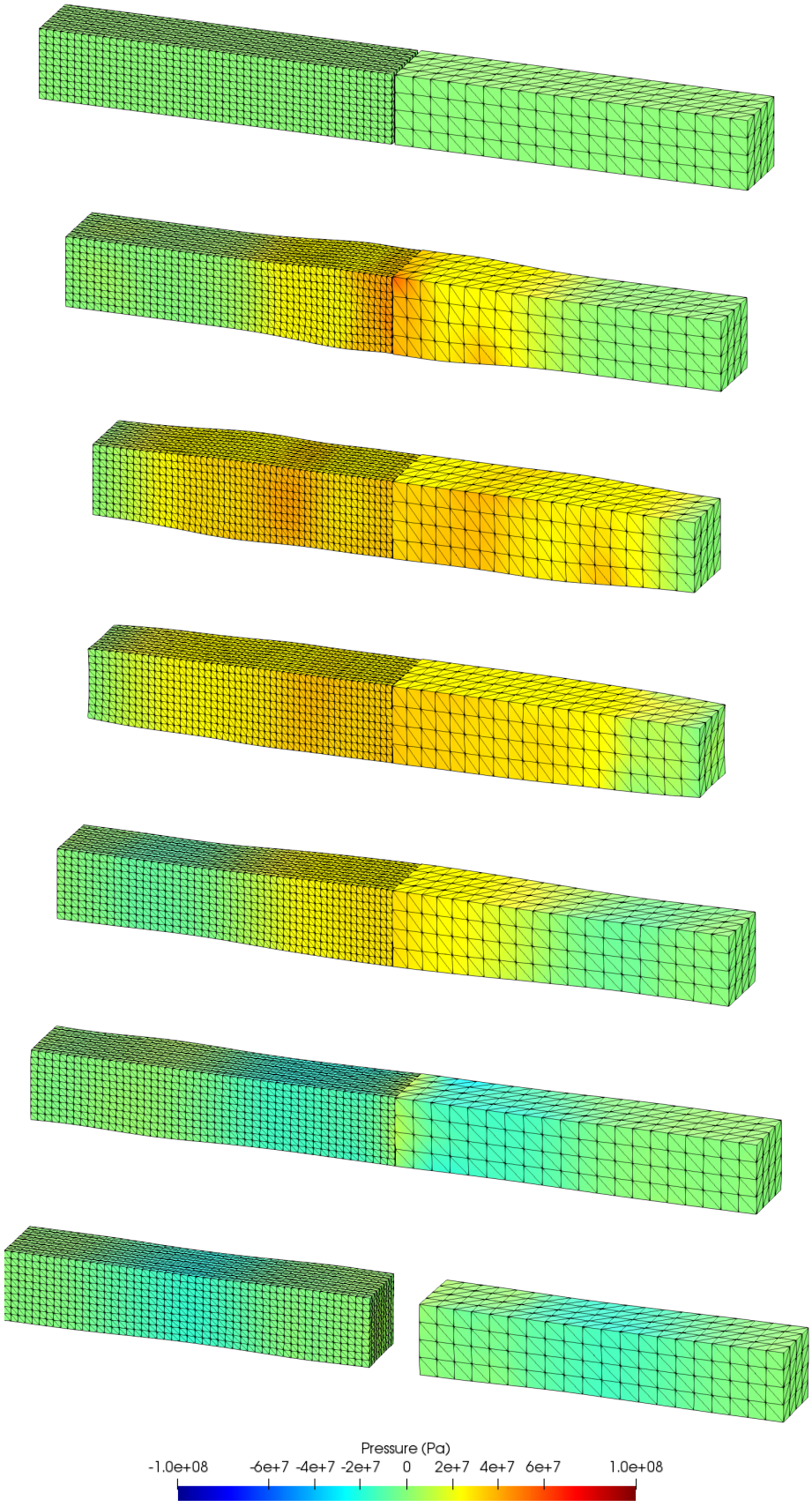


Figure 7.31: Non-Matching Nearly Incompressible Bars - Time Evolution of non-matching Mesh 1 at 0μs, 50μs, 75μs, 100μs, 150μs, 200μs, 250μs and 300μs

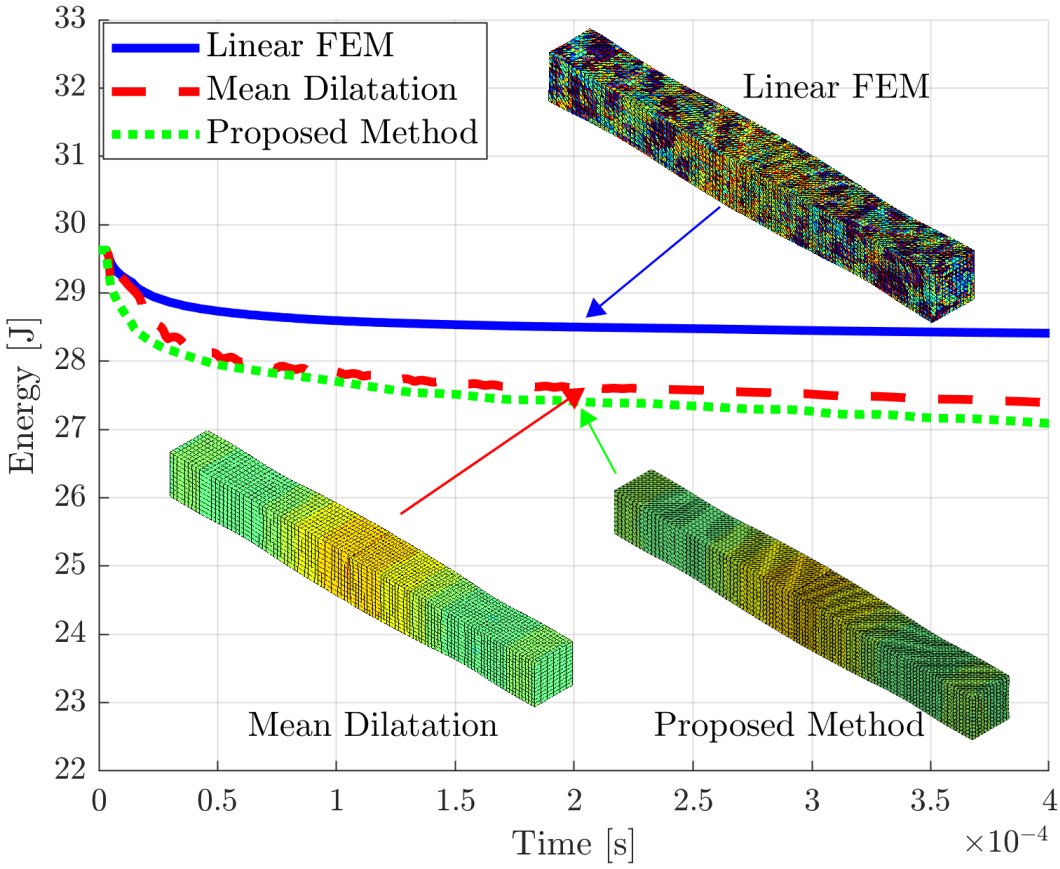


Figure 7.32: Non-Matching Nearly Incompressible Bars - Global total energy comparison of the proposed method, linear FEM and mean dilatation for non-matching Mesh 4

7.7 Car Wall Impact

7.7.1 Problem Description

This problem considers a simplified car travelling at an initial velocity of -8.94 m/s (20 MPH) towards a wall with an initial gap δ_n of 0.0894 m and is subjected to a gravitational body force of 9.81 N, where the car cross-section (XY-Plane) is depicted in Figure 7.33 with total length L_T of 2.6 m, total height H_T of 1 m, total width W_T of 1 m and the detailed dimensions in Table 7.9. In this problem the car model is simplified to be considered as a single solid homogeneous material with the material properties in Table 7.10 for two different material models, nearly incompressible neo-Hookean and von-Mises plasticity with isotropic hardening. The purpose of this problem was to investigate the robustness of the proposed method and compare with classical FEM for more complex geometries in relation to a practical real world application.

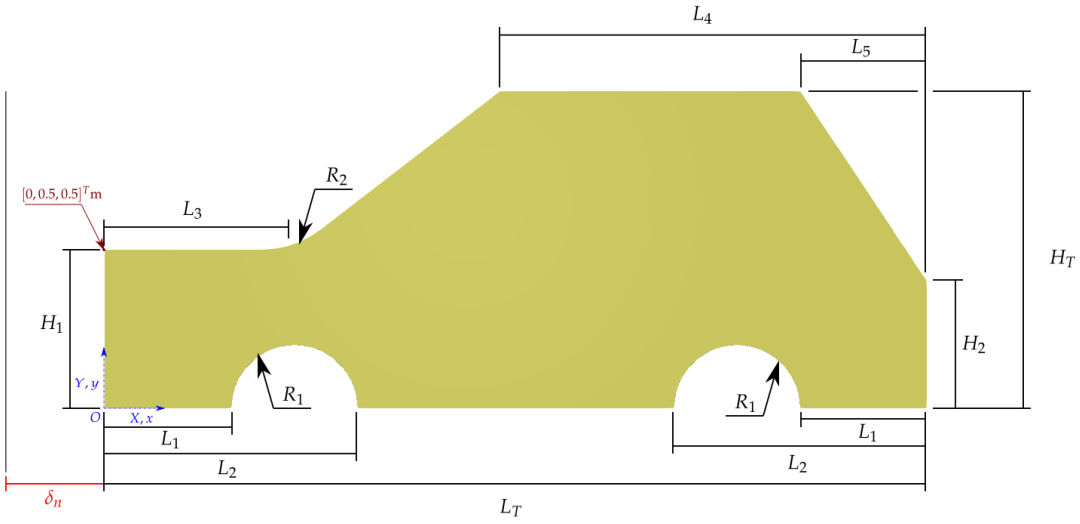


Figure 7.33: Car Wall Impact - Problem description

Table 7.9: Car Wall Impact - Dimensions

Length	[m]	Height	m	Radius	m
L_1	0.4	H_1	0.5	R_1	0.2
L_2	0.8	H_2	0.4	R_2	0.3
L_3	0.6				
L_4	1.35				
L_5	0.4				

Table 7.10: Car Wall Impact - Material parameters

		Elastic car	Plastic car	
Constitutive Model		neo-Hookean	von-Mises Plasticity	
Young's Modulus	E_0	$195 \cdot 10^6$	$195 \cdot 10^6$	$[\mathrm{N/m^2}]$
Material density	ρ_0	8000	8000	$[\mathrm{kg/m^3}]$
Poisson Ratio	ν	0.3	0.3	
Lamé parameters	μ	75.38	75.38	$[\mathrm{MN/m^2}]$
	λ	113.077	113.077	$[\mathrm{MN/m^2}]$
Yield stress	$\sigma_{y,0}$	-	$2 \cdot 10^6$	$[\mathrm{N/m^2}]$
Hardening parameter	H	-	$20 \cdot 10^6$	$[\mathrm{N/m^2}]$
Friction coefficient	k	0	0	

7.7.2 Numerical Results

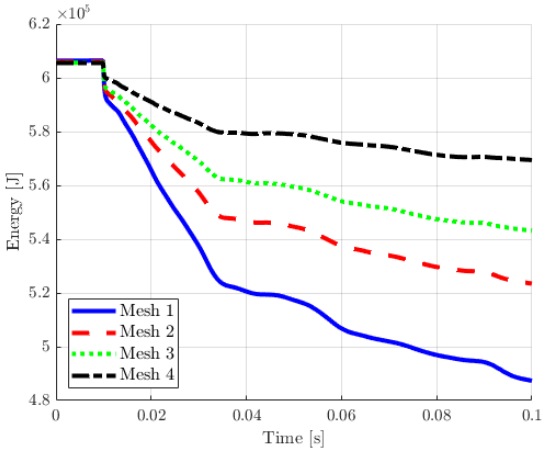
This problem was simulated for the proposed method using a CFL of 0.3 and for four different tetrahedral meshes with the following mesh statistics: Mesh 1 had 2391 nodes and 9772 elements, Mesh 2 had 8582 nodes and 44911 elements, Mesh 3 had 17496 nodes and 92479 elements, Mesh 4 had 96359 nodes and 554190 elements. The resulting total system energy history, numerical error history and energy history for each mesh and material model are shown in Figure 7.34 to demonstrate consistency.

Figure 7.35 and 7.36 show the local time history of axial velocity, axial stress, axial displacement and pressure at initial position $\boldsymbol{X} = [0, 0.5, 0]^T$, for the elastic car impact (left) and plastic car impact (right), comparing the proposed method using Mesh 4, linear FEM using a tetrahedral mesh consisting of 84921 nodes and 458757 elements and mean dilatation FEM using a hexahedral mesh consisting of 157563 nodes and 148040 elements. Through comparing the local results, each methodology matches closely in the elastic car impact while for the plastic car impact example the axial velocity and axial displacement match closely but for the axial stress and pressure the linear FEM has oscillations in the solution while the proposed method matches closely with the mean dilatation FEM.

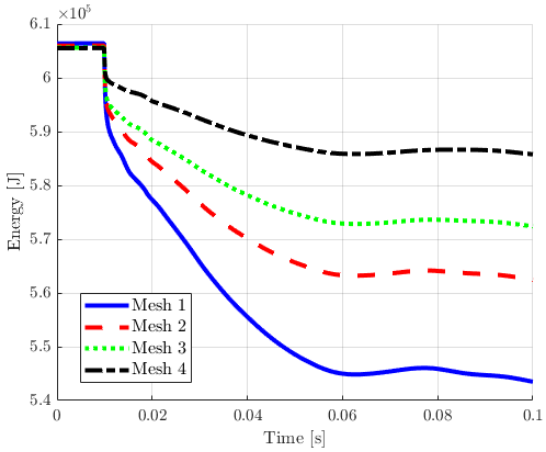
Figure 7.37 shows the pressure contour snapshots for each methodology of the elastic car impact at various time intervals. Each methodology demonstrates a smooth pressure contour with good agreement across each methodology with slight differences in resolution. For completeness the entire time evolution for the proposed method is shown in Figure 7.38.

Figure 7.39 shows the pressure contour snapshots for each methodology of the plastic

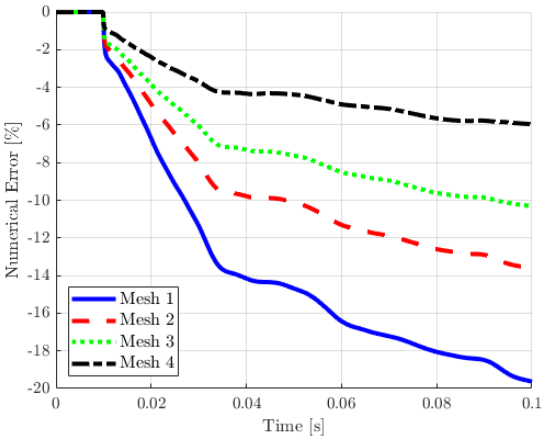
car impact at various time intervals. The pressure contour using the linear FEM with a tetrahedral mesh results in pressure checker boarding in the vicinity of high stress concentrations which are plasticly deforming. While the proposed method and mean dilatation FEM produce a smooth pressure contour which match closely with a slight difference in resolution. For completeness the entire time evolution for the proposed method is shown in Figure 7.40.



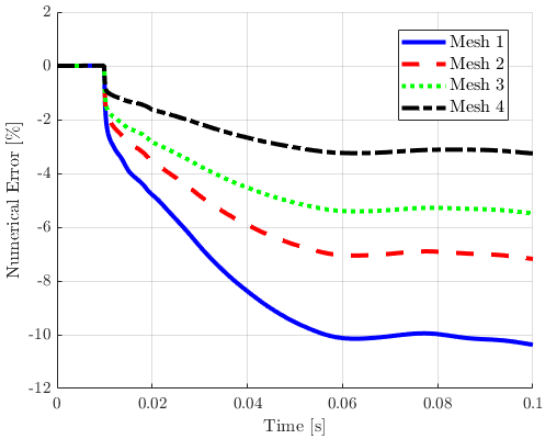
(a) Total system energy - Mesh refinement



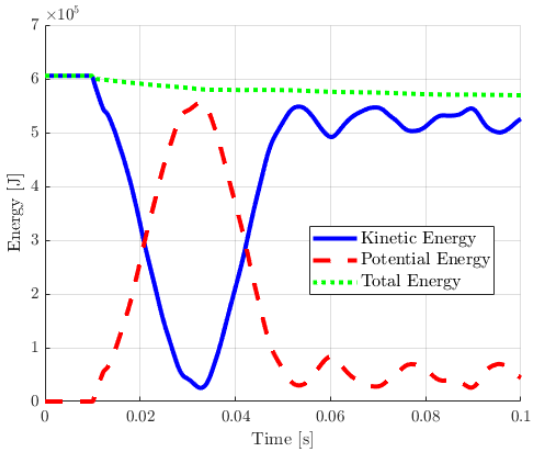
(b) Total system energy - Mesh refinement



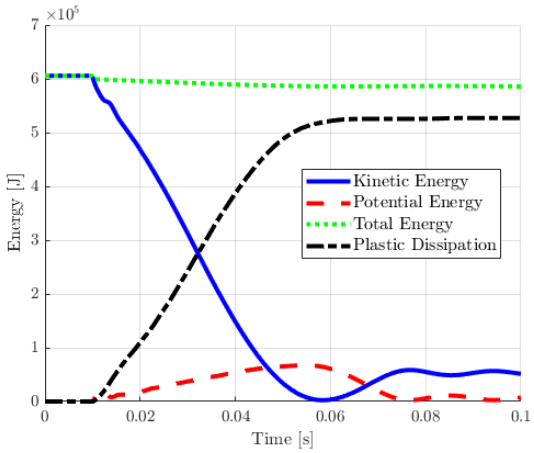
(c) Numerical error - Mesh refinement



(d) Numerical error - Mesh refinement

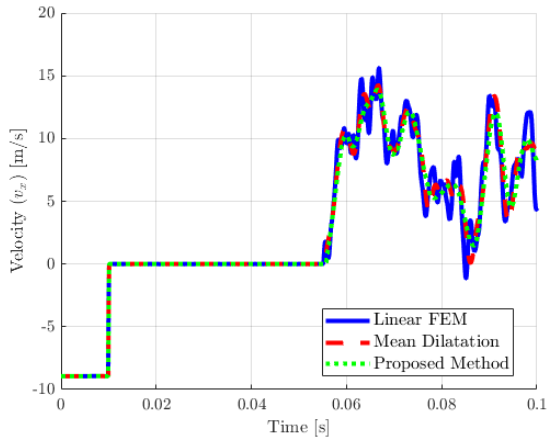


(e) Energy history (Mesh 4)

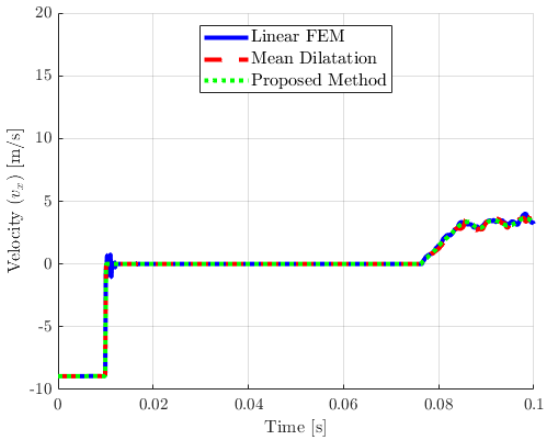


(f) Energy history (Mesh 4)

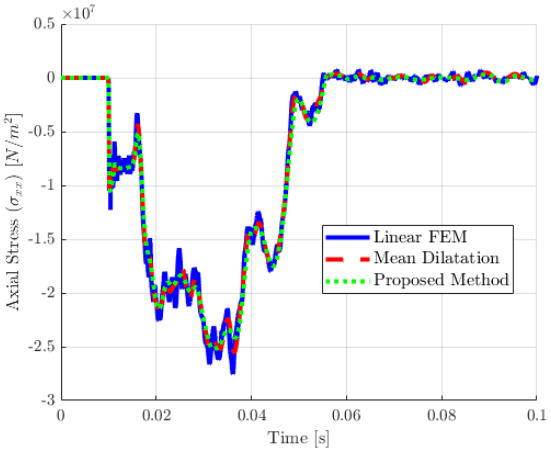
Figure 7.34: Car Wall Impact - Energy time histories for the elastic car impact (left) and plastic car impact (right)



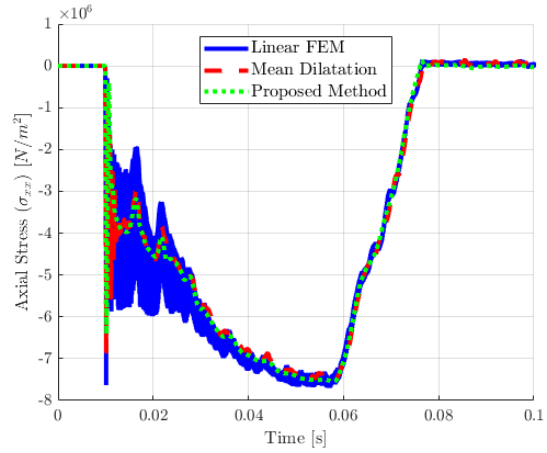
(a) Velocity v_x



(b) Velocity v_x

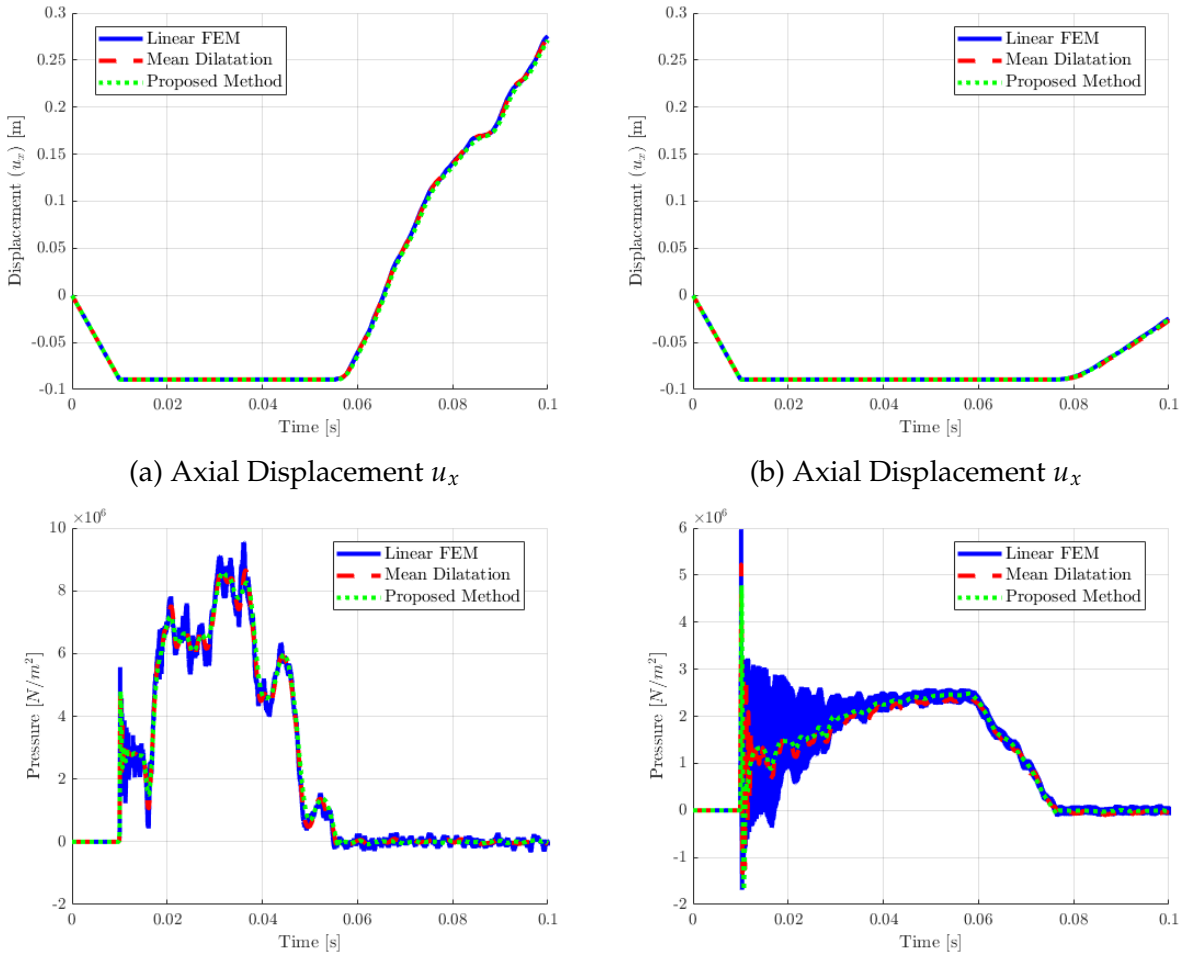


(c) Axial Stress σ_{xx}



(d) Axial Stress σ_{xx}

Figure 7.35: Car Wall Impact - Comparison of local axial velocity and axial stress for the elastic car (left) and plastic car (right) using linear tetrahedral FEM, mean dilatation hexahedral FEM and tetrahedral proposed method



(a) Axial Displacement u_x

(b) Axial Displacement u_x

(c) Pressure

(d) Pressure

Figure 7.36: Car Wall Impact - Comparison of local axial displacement and pressure for the elastic car (left) and plastic car (right) using linear tetrahedral FEM, mean dilatation hexahedral FEM and tetrahedral proposed method

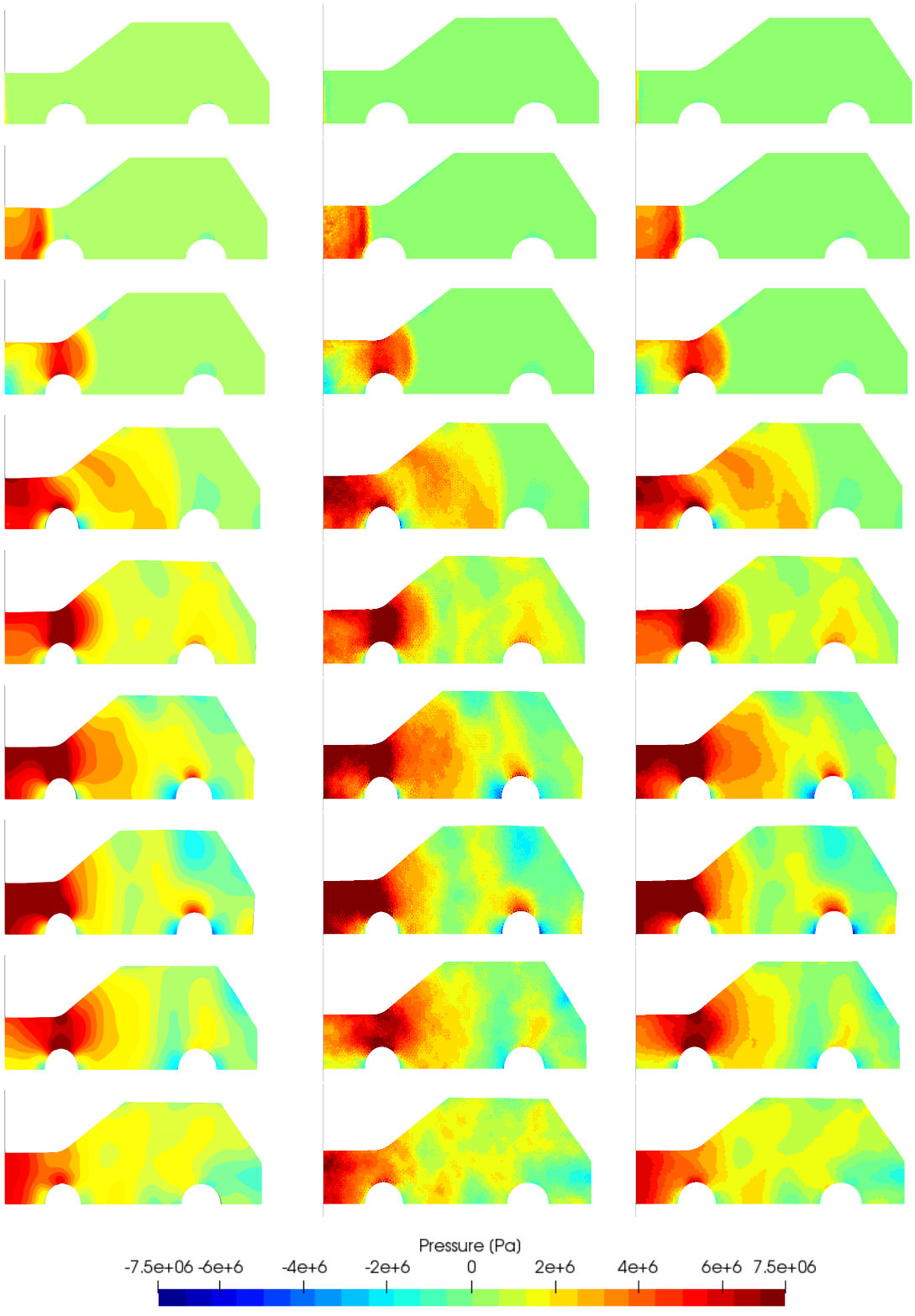


Figure 7.37: Elastic Car Impact - Pressure Time Evolution of proposed method using a tetrahedral mesh (left), linear FEM using a tetrahedral mesh and mean dilatation FEM using a hexahedral mesh 100ms, 125ms,150ms, 200ms, 250ms, 300ms, 350ms, 400ms, 450ms

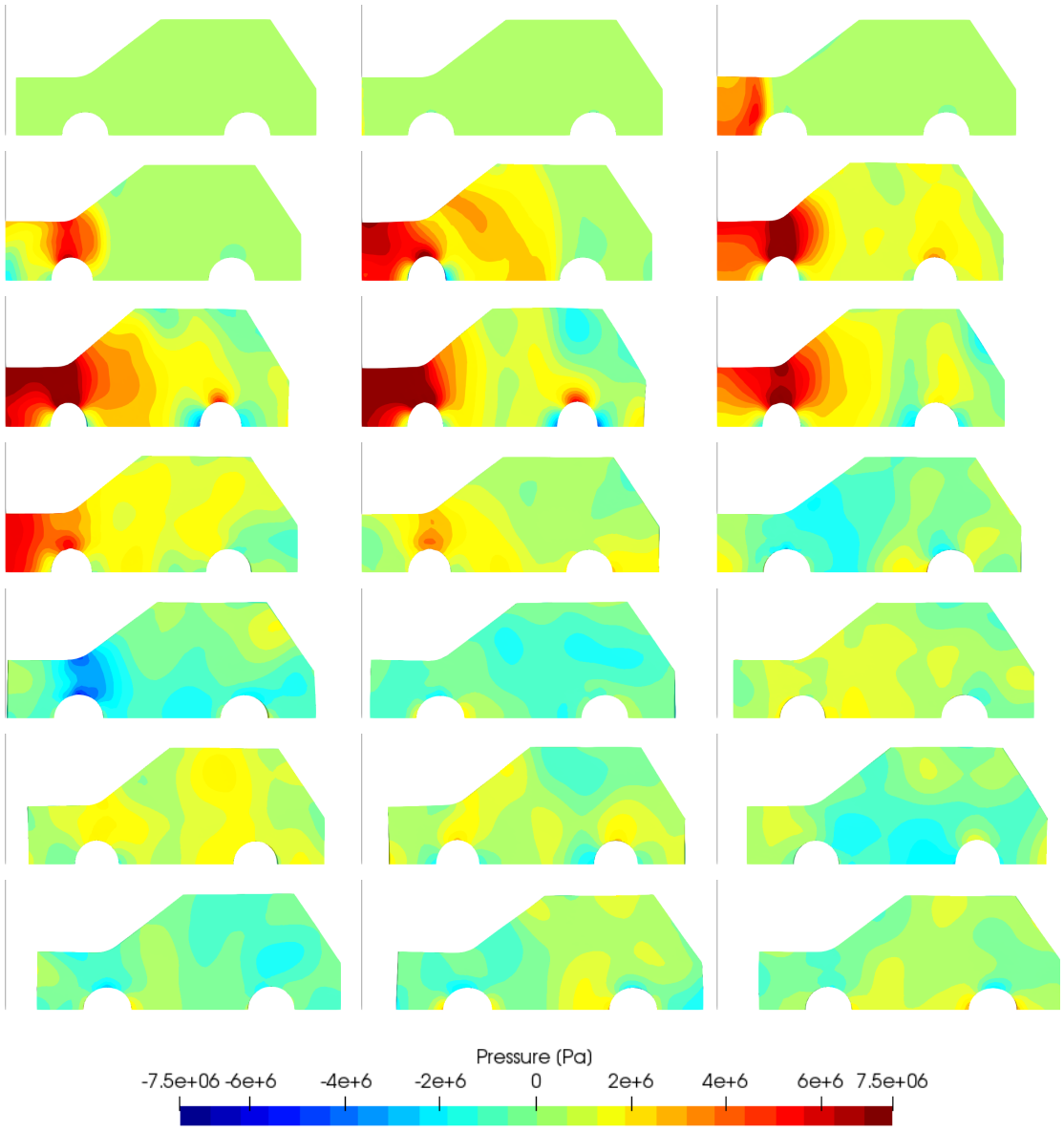


Figure 7.38: Elastic Car Impact - Pressure Time Evolution of proposed method using a tetrahedral mesh (from left to right and top to bottom) at 0ms, 100ms, 125ms, 150ms, 200ms, 250ms ... 900ms, 950ms, 1000ms

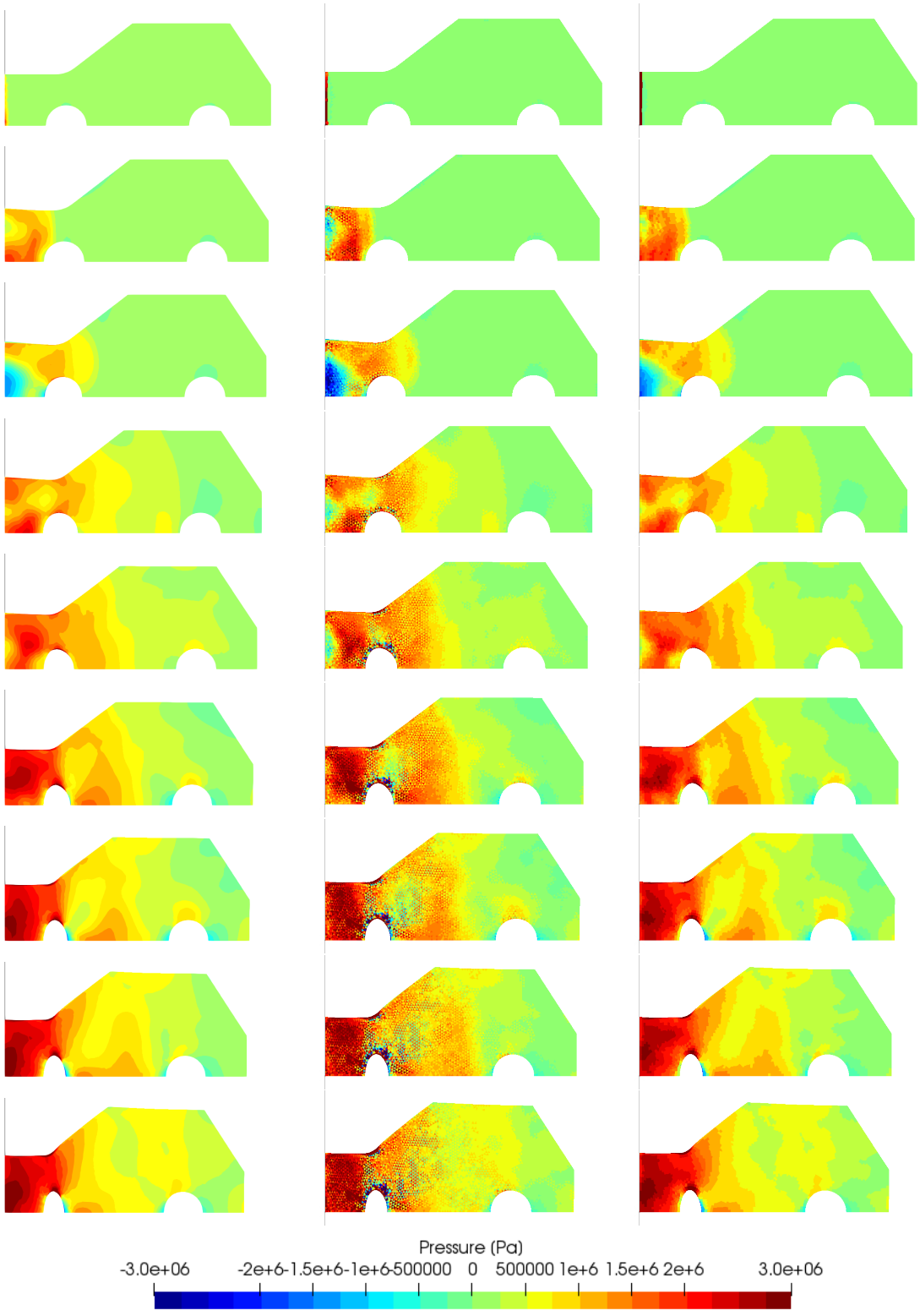


Figure 7.39: Plastic Car Impact - Pressure Time Evolution of proposed method using a tetrahedral mesh (left), linear FEM using a tetrahedral mesh and mean dilatation FEM using a hexahedral mesh 100ms, 125ms,150ms, 200ms, 250ms, 300ms, 350ms, 400ms, 450ms

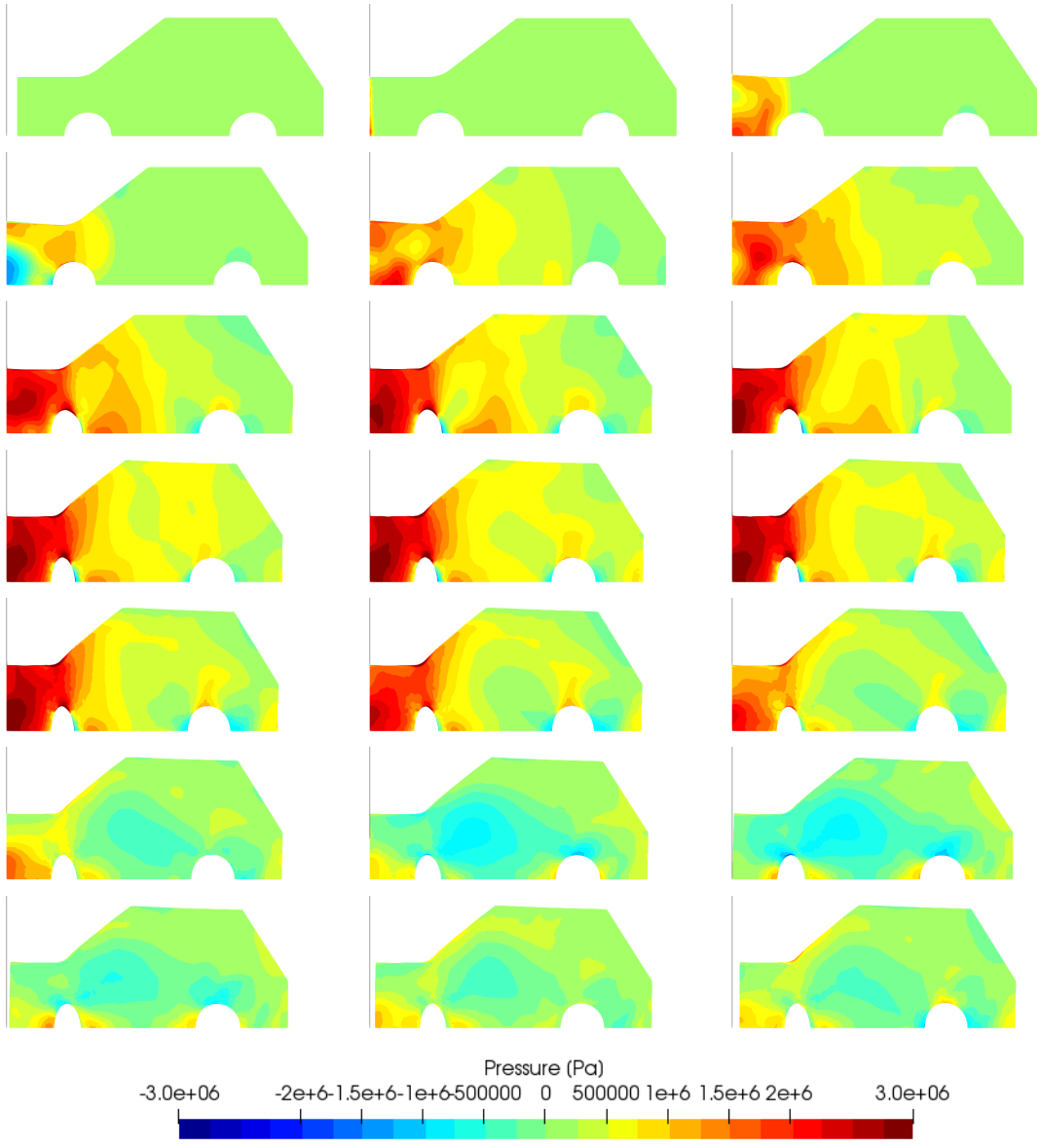


Figure 7.40: Plastic Car Impact - Pressure Time Evolution of proposed method using a tetrahedral mesh (from left to right and top to bottom) at 0ms, 100ms, 125ms,150ms, 200ms, 250ms ... 900ms, 950ms, 1000ms

Part V

Concluding Remarks

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis has presented a novel vertex centred finite volume method for the dynamic solution of non-smooth contact problems, where a mixed system of first order conservation equations, in combination with the associated jump conditions was used. By utilising the jump condition for the conservation of linear momentum, several dynamic contact models were derived ensuring the preservation of the hyperbolic characteristic structure across the contact interface, this included two types of solver based on the acoustic material wave speed and Hugoniot shock wave speed. This novel approach was able to enforce both kinematic and kinetic contact constraints at a continuum level and was demonstrated at a discrete level, first through a one-dimensional implementation in MATLAB followed by multi-dimensional implementations in two-dimensions through MATLAB and three-dimensions through open source software "OpenFOAM". The numerical examples presented in this thesis, prove the proposed algorithm performs extremely well in dynamic contact-impact problems without requiring any ad-hoc algorithmic regularisation. The proposed method, by construction, also overcomes several numerical drawbacks commonly found in traditional methodologies. Specifically, no spurious oscillations or hour-glassing was observed and a smooth pressure solution was obtained unlike the traditional linear tetrahedral finite element method. The proposed algorithm, crucially ensures long-term stability through monitoring the global entropy production via the Hamiltonian energy of the system.

This thesis outlined the governing equations for the kinematics of a single continuum body in relation to solid dynamics. Followed by the kinematic description along with the definitions for the Total Lagrangian description of the first order conservation framework first developed by Lee *et al.* [100]. In terms of the conservation linear

momentum $\boldsymbol{p}$ and triplet of kinematic conservation equations $\{\boldsymbol{F}, \boldsymbol{H}, J\}$. The solid dynamic system was closed by defining a polyconvex constitutive model and a classic elastoplasticity model. To consider discontinuous problems as part of the first order framework, such as contact-impact scenarios, the associated jump conditions were derived and discussed. To evaluate these conditions, appropriate acoustic material wave speeds and Hugoniot material wave speeds were derived.

By examining the linear momentum jump condition, expressions for the contact interface conditions were derived for the acoustic material wave speed and Hugoniot shock wave speeds, in the form of a Riemann solver, which directly enforce the contact conditions. To demonstrate the application of these conditions, the exact solution to a one-dimensional contact scenario was presented. The second law of thermodynamics was introduced in the context of the Hamiltonian energy when considering the first order conservation framework. It was demonstrated that at a continuum level the proposed formulation satisfies the Coleman-Noll for entropy production.

In terms of numerical scheme, the first order conservation framework was discretised in space using vertex centred finite volume method. In order to achieve a second order accurate solution, the fluxes were reconstructed using a linear reconstruction procedure with a Barth-Jespersen limiter. The entropy production was examined using the Coleman-Noll procedure and demonstrated that the proposed method fulfils the second law of thermodynamics at a semi-discrete level. The proposed method was discretised in time using a time-variation diminishing two stage Runge-Kutta time integrator. With this numerical scheme the proposed method was implemented as a purpose built explicit multi-body contact solver in open-source software OpenFOAM with a custom VCFVM library.

One-dimensional numerical examples were presented to demonstrate the accuracy and consistency of the proposed algorithm, while monitoring the Hamiltonian energy to ensure fulfilment of the second law of thermodynamics. These examples showed near perfect agreement with the exact solutions and demonstrated the satisfaction of the second law of thermodynamics. The influence of the Hugoniot wave speed using a non-linear Riemann solver was also investigated showing good agreement with traditional FEMs. A parametric study was conducted that demonstrated the importance of using the Hugoniot shock model for large velocity impacts, while the importance of utilising a non-linear Riemann solver requires further investigation.

More advanced multi-dimensional problems were also presented and compared with standard finite element methodologies, linear triangular and mean dilatation quadrilaterals elements in two dimensions then linear tetrahedral and mean dilatation hexa-

hedral elements in three dimensions. The proposed method again demonstrated good agreement with the mean dilatation method across both dimensions, free from spurious oscillations and pressure checker boarding which were observed in traditional linear methods. The Hamiltonian energy was monitored and examined for each example and demonstrated the satisfaction of the second law of thermodynamics for each case.

The main contributions and novelties of this thesis are as follows:

Continuum Level

- Application of the first order conservation framework to multiple body contact dynamics.
- Derivation of continuum contact conditions based on the jump conditions for two types of wave speed in the form of an acoustic and non-linear Riemann solvers.
- This combination of methods directly enforces the contact conditions (impenetrability and traction conservation) at a continuum level.
- The satisfaction of entropy production using a Coleman-Noll procedure at a continuum level for the proposed formulation.

Discrete Level

- The enforcement of both contact constraints at each node of the contact interface across multiple bodies.
- The discrete satisfaction of local entropy production using a Coleman-Noll procedure for the proposed method.

Computational Implementation

- Purpose built "OpenFOAM" implementation of the proposed method including a vertex centred finite volume library and multiple body contact solver for predefined matching and non-matching contact interfaces.

8.2 Future Work

Following this research there are several avenues for future work as follows:

- **Non-linear Riemann Solver**
  The advantage of the non-linear Riemann solver, introduced in this thesis, is

not yet clear requiring further investigation through multi-dimensional scenarios such as fictional or angled contact.

- **Constitutive models**
  This work was focused on the contact-impact of a small selection of large strain constitutive models, further more complex constitutive models could be investigated such as viscoelasticity [129, 130, 156] or viscoplasticity [157] as well as the contact between bodies of different constitutive models.

- **Thermo-mechanical contact**
  In this work only isothermal contact was considered, this can be extended to include thermal effects in contact scenarios such as friction or heat transfer requiring additional contact constraints [158]. The addition of these contact constraints can be investigated by the proposed method through an additional conservation equation for the first law of thermodynamics as demonstrated by Bonet *et al.* in [117] with the required theory presented in Appendix A. Modelling Thermo-mechanical contact would require more complex constitutive models such as thermo-elasticity [159], thermo-plasticity [160] or thermo-viscoelasticity [161].

- **Non-matching/Sliding Contact Interfaces**
  The investigation of methodologies in order to address non-matching/sliding contact interfaces was outside the scope of this work as such further improvements can be made to the proposed algorithm to address these scenarios. To fully consider these types of scenarios an efficient search algorithm, as discussed in Section 1.2.2, would be required to identify contact regions.

- **OpenFOAM Integration and Extension to "preCICE"**
  The implementation of the proposed method in "OpenFOAM" for this thesis was entirely purpose-built as such various improvements can be made to the implementation to improve the integration with OpenFOAM, and ultimately efficiency, such as creating a complete template library for the vertex centred finite volume algorithm following the coding guidelines. By doing so the proposed algorithm could easily be scaled and adapted for different computational algorithms within OpenFOAM and extended to integration with third party modules such as "preCICE" [162] for the coupling of Fluid-Structure Interaction (FSI) problems.

- **Computational Efficiency Comparison**
  The proposed method has a series of advantages over traditional displacement based methods as discussed throughout this thesis, however to fully assess this framework, a computational efficiency study can be undertaken in order to assess whether the additional geometric conservation equations have an impact on

overall efficiency in comparison to other methodologies.

Part VI

Appendix

Appendix A

Governing Equations for Thermo-Elasticity

A.1 Preliminaries

This thesis focused on isothermal processes, in this appendix, the relevant governing equations to extend the proposed method to thermo-elastic scenarios is presented following the work of Ghavamian *et al.* [116] and Bonet *et al.* [117]. This appendix begins with the introduction of the first law of thermodynamics in Section A.2 expressed in terms of total energy and in terms of entropy. Two appropriate constitutive relations are introduced in Section A.3 namely, Modified Entropic Elasticity (MEE) and Mie-Grüneisen (MG) in order to close the system of equations. Lastly, the extension of the proposed method to thermo-mechanical contact is briefly discussed in Section A.4.

A.2 First Law of Thermodynamics

A.2.1 Expressed in Terms of Total Energy

The first law of thermodynamics is expressed in terms of the total energy in global form as [21]:

$$\frac{d}{dt}\int_{\Omega_0} E_T dV = \int_{\partial\Omega_0} \boldsymbol{t}\cdot\boldsymbol{v} dA - \int_{\partial\Omega_0} Q_B dA + \int_{\Omega_0} (\rho_0 \boldsymbol{b}_0)\cdot\boldsymbol{v} dV + \int_{\Omega_0} r_0 dV, \tag{A.1}$$

where $E_T(\boldsymbol{X}, t)$ is the total energy density per unit undeformed volume, $\boldsymbol{t} = \boldsymbol{P}\boldsymbol{N}$ is the boundary traction vector, $Q_B = \boldsymbol{Q}\cdot\boldsymbol{N}$ is the heat flow normal to boundary $\partial\Omega_0$, r_0 is the thermal heat source per unit reference volume. The local form of this conservation

equation is expressed as:

$$\frac{\partial E_T}{\partial t} + \text{DIV}\left(\boldsymbol{Q} - \boldsymbol{P}^T\boldsymbol{v}\right) = (\rho_0 \boldsymbol{b}_0)\cdot\boldsymbol{v} + r_0, \tag{A.2}$$

with jump condition defined as [100]:

$$U[\![E_T]\!] = [\![\boldsymbol{Q}]\!]\cdot\boldsymbol{N} - [\![\boldsymbol{P}^T\boldsymbol{v}]\!]\cdot\boldsymbol{N}. \tag{A.3}$$

where $[\![\cdot]\!] = [\![\cdot]\!]^R - [\![\cdot]\!]^L$ is the jump across a discontinuous surface by unit normal $\boldsymbol{N}$ propagating with wave speed U.

A.2.2 Expressed in Terms of Internal Energy and the Entropy

The total energy density in Equation (A.2) includes kinetic and internal energy contributions. Therefore to obtain an expression in terms of the internal energy only, the linear momentum balance principle in Equation (2.7) is multiplied by velocity $\boldsymbol{v}$ and subtracted from Equation (A.2), after some algebra yields:

$$\frac{\partial \mathscr{E}}{\partial t} + \text{DIV}\,\boldsymbol{Q} = \boldsymbol{P} : \boldsymbol{\nabla}_0\boldsymbol{v} + r_0, \tag{A.4}$$

where $\mathscr{E}(\boldsymbol{X},t) = E_T - \frac{1}{2\rho_0}(\boldsymbol{p}\cdot\boldsymbol{p})$ represents the internal energy per unit undeformed volume. In strict thermo-elasticity, the internal energy $\mathscr{E}(\boldsymbol{X},t)$ will depend on the triplet of deformation measures $\boldsymbol{\mathcal{X}} = \{\boldsymbol{F},\boldsymbol{H},J\}$ and entropy density (per unit of undeformed volume) η, defined as:

$$\mathscr{E}(\boldsymbol{X},t) = \mathcal{E}\left(\boldsymbol{\mathcal{X}}_\eta\right); \quad \boldsymbol{\mathcal{X}}_\eta = \{\boldsymbol{\mathcal{X}},\eta\} = \{\boldsymbol{F},\boldsymbol{H},J,\eta\}, \tag{A.5}$$

where $\mathcal{E}$ denotes the same internal energy per unit undeformed volume as $\mathscr{E}$ but with different functional dependency. The entropy density field $\eta(\boldsymbol{X},t)$ is defined as the energy dual conjugate variable to the temperature $\theta(\boldsymbol{X},t)$ as:

$$\theta(\boldsymbol{X},t) = \frac{\partial\mathcal{E}\left(\boldsymbol{\mathcal{X}}_\eta\right)}{\partial\eta} = \Theta\left(\boldsymbol{\mathcal{X}}_\eta\right). \tag{A.6}$$

Similarly, the notation θ and Θ are used to denote the same temperature with different functional dependency. Similarly, energy conjugate fields can be defined to the three deformation measures of the triplet $\boldsymbol{\mathcal{X}}$ as:

$$\boldsymbol{\Sigma}_{\boldsymbol{F}} = \frac{\partial\mathcal{E}\left(\boldsymbol{\mathcal{X}}_\eta\right)}{\partial\boldsymbol{F}}; \quad \boldsymbol{\Sigma}_{\boldsymbol{H}} = \frac{\partial\mathcal{E}\left(\boldsymbol{\mathcal{X}}_\eta\right)}{\partial\boldsymbol{H}}; \quad \Sigma_J = \frac{\partial\mathcal{E}\left(\boldsymbol{\mathcal{X}}_\eta\right)}{\partial J}. \tag{A.7}$$

By comparing the time rate of the internal strain energy $\mathscr{E}(\boldsymbol{X},t)$ to that of its equivalent re-expression $\mathcal{E}(\boldsymbol{\mathcal{X}}_\eta)$ and, using the properties of the tensor cross product [106], it is possible to relate the conjugate stresses defined previously (Equation(A.7)) to the standard first Piola-Kirchhoff stress tensor (refer to Section 2.5) as:

$$\frac{\partial \mathscr{E}(\boldsymbol{X},t)}{\partial t} = \frac{\partial \mathcal{E}(\boldsymbol{\mathcal{X}}_\eta)}{\partial \boldsymbol{F}} : \frac{\partial \boldsymbol{F}}{\partial t} + \frac{\partial \mathcal{E}(\boldsymbol{\mathcal{X}}_\eta)}{\partial \boldsymbol{H}} : \frac{\partial \boldsymbol{H}}{\partial t} + \frac{\partial \mathcal{E}(\boldsymbol{\mathcal{X}}_\eta)}{\partial J}\frac{\partial J}{\partial t} + \frac{\partial \mathcal{E}(\boldsymbol{\mathcal{X}}_\eta)}{\partial \eta}\frac{\partial \eta}{\partial t}; \tag{A.8}$$

$$= \boldsymbol{\Sigma_F} : \boldsymbol{\nabla}_0 \boldsymbol{v} + \boldsymbol{\Sigma_H} : (\boldsymbol{F} \times \boldsymbol{\nabla}_0 \boldsymbol{v}) + \Sigma_J (\boldsymbol{H} : \boldsymbol{\nabla}_0 \boldsymbol{v}) + \theta \frac{\partial \eta}{\partial t}; \tag{A.9}$$

$$= [\boldsymbol{\Sigma_F} + \boldsymbol{\Sigma_H} \times \boldsymbol{F} + \Sigma_J \boldsymbol{H}] : \boldsymbol{\nabla}_0 \boldsymbol{v} + \theta \frac{\partial \eta}{\partial t}, \tag{A.10}$$

leading to:

$$\boldsymbol{P}(\boldsymbol{\mathcal{X}}_\eta) = \boldsymbol{\Sigma_F} + \boldsymbol{\Sigma_H} \times \boldsymbol{F} + \Sigma_J \boldsymbol{H}. \tag{A.11}$$

Now it is possible to re-write the first law of thermodynamics in terms of the entropy η by combining Equations (A.4) and (A.8) to give a thermal expression in which the mechanical terms have been eliminated to give [163]:

$$\theta \frac{\partial \eta}{\partial t} + \mathrm{DIV}\, \boldsymbol{Q} = r_0. \tag{A.12}$$

Alternatively, noting $\frac{1}{\theta}\mathrm{DIV}\,\boldsymbol{Q} = \mathrm{DIV}\left(\frac{\boldsymbol{Q}}{\theta}\right) + \frac{1}{\theta^2}\boldsymbol{Q}\cdot\boldsymbol{\nabla}_0\theta$, a conservation type of law for the entropy emerges as:

$$\frac{\partial \eta}{\partial t} + \mathrm{DIV}\left(\frac{\boldsymbol{Q}}{\theta}\right) = \frac{r_0}{\theta} - \frac{1}{\theta^2}\boldsymbol{Q}\cdot\boldsymbol{\nabla}_0\theta, \tag{A.13}$$

where $\frac{\boldsymbol{Q}}{\theta}$ represents the flux of entropy and the right hand side term is the entropy source per unit of undeformed volume. The expression above for entropy, however, assumes a smooth solution but is still an expression for the first law. The global form can be obtained by integration over an arbitrary volume Ω_0 with the use of the divergence theorem to give:

$$\frac{\partial}{\partial t}\int_{\Omega_0} \eta dV + \int_{\partial\Omega}\left(\frac{Q_B}{\theta}\right) dA = \int_{\Omega_0}\left(\frac{r_0}{\theta}\right) dV - \int_{\Omega_0}\frac{1}{\theta^2}\boldsymbol{Q}\cdot\boldsymbol{\nabla}_0\theta dV. \tag{A.14}$$

In relation to heat flux vector $\boldsymbol{Q}$, the typical Fourier law is considered to hold and which can be defined in Total Lagrangian fashion as:

$$\boldsymbol{Q} = -\boldsymbol{K}\boldsymbol{\nabla}_0\theta; \quad \boldsymbol{K} = J^{-1}\boldsymbol{H}^T\boldsymbol{k}\boldsymbol{H}. \tag{A.15}$$

where $\boldsymbol{k}$ represents the positive semi-discrete second-order thermal conductivity tensor

in the deformed configuration.

A.3 Thermal Constitutive Relations

In general, the Calorimetry relationships between internal energy $\mathcal{E}$, temperature θ and entropy η can be derived from the definition of the specific heat at constant volume c_v [117]. This requires the re-definition of the entropy $\eta\left(\boldsymbol{X},t\right)$ and the internal energy density $\mathscr{E}\left(\boldsymbol{X},t\right)$ in terms of the triplet of deformation measures $\boldsymbol{\mathcal{X}}$ and the temperature θ namely, $\boldsymbol{\mathcal{X}}_\theta = \{\boldsymbol{\mathcal{X}},\theta\} = \{\boldsymbol{F},\boldsymbol{H},J,\theta\}$. Defined as:

$$c_v \overset{\text{def}}{=} \left.\frac{d\mathcal{E}}{d\theta}\right|_{\boldsymbol{\mathcal{X}}=const};\quad \tilde{\mathcal{E}}\left(\boldsymbol{\mathcal{X}}_\theta\right) = \mathcal{E}\left(\boldsymbol{\mathcal{X}},\tilde{\eta}\left(\boldsymbol{\mathcal{X}}_\theta\right)\right);\quad \eta\left(\boldsymbol{X},t\right) = \tilde{\eta}\left(\boldsymbol{\mathcal{X}}_\theta\right), \tag{A.16}$$

with $c_v = \rho_0 C_v > 0$, where ρ_0 is the density at reference temperature θ_0 and C_v the specific heat per unit mass. As the internal energy $\mathscr{E}\left(\boldsymbol{X},t\right)$ can be expressed as a function of the set of arguments $\boldsymbol{\mathcal{X}}_\eta$, Equation (A.5) and observing that from Equation (A.6) that, $\partial\mathcal{E}\left(\boldsymbol{\mathcal{X}}_\eta\right)/\partial\eta = \theta$, a constitutive relationship between the temperature θ and the entropy η at constant deformation can be established [117] by re-expressing Equation (A.16) using the chain rule as:

$$\frac{\partial\tilde{\eta}\left(\boldsymbol{\mathcal{X}}_\theta\right)}{\partial\theta} = \frac{c_v}{\theta}. \tag{A.17}$$

Restricting the derivation to the simple constant heat coefficient case, allows Equation (A.17) to be integrated analytically with respect to the entropy or temperature changes as:

$$\int_{\tilde{\eta}_0(\boldsymbol{\mathcal{X}})}^{\tilde{\eta}(\boldsymbol{\mathcal{X}}_\theta)} d\eta = c_v \int_{\theta_0}^{\theta} \frac{1}{\theta} d\theta, \tag{A.18}$$

which lead to a relationship between entropy and temperature as:

$$\tilde{\eta}\left(\boldsymbol{\mathcal{X}}_\theta\right) = \tilde{\eta}_0\left(\boldsymbol{\mathcal{X}}\right) + c_v \ln\frac{\theta}{\theta_0};\quad \tilde{\eta}_0\left(\boldsymbol{\mathcal{X}}\right) = \tilde{\eta}\left(\boldsymbol{\mathcal{X}},\theta=\theta_0\right), \tag{A.19}$$

where $\tilde{\eta}_0\left(\boldsymbol{\mathcal{X}}\right)$ is the entropy at constant temperature θ_0 and expressed as a function of the deformation $\boldsymbol{\mathcal{X}}$ (In other words, after deformation, the temperature is allowed to return to the reference value θ_0).

Using Equation (A.19), and noting the alternative functional representation from Equation (A.16), the reverse relationship yielding the temperature as a function of the deformation and entropy is given as:

$$\Theta\left(\boldsymbol{\mathcal{X}}_\eta\right) = \theta_0 e^{(\eta-\tilde{\eta}_0(\boldsymbol{\mathcal{X}}))/c_v}. \tag{A.20}$$

Again, with the assumption of a constant specific heat coefficient c_v, it is possible to

write an explicit relationship for the internal energy $\mathscr{E}(\boldsymbol{X}, t)$ as functions of deformations $\boldsymbol{\mathcal{X}}$ and temperature θ, that is $\tilde{\mathcal{E}}(\boldsymbol{\mathcal{X}}_\theta)$, instead of $\mathcal{E}(\boldsymbol{\mathcal{X}}_\eta)$. This is achieved by directly integrating Equation (A.16) with respect to temperature between the limits θ_0 and a given value θ to give:

$$\int_{\tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}})}^{\tilde{\mathcal{E}}(\boldsymbol{\mathcal{X}}_\theta)} d\mathcal{E} = c_v \int_{\theta_0}^{\theta} d\theta, \tag{A.21}$$

which then yields:

$$\tilde{\mathcal{E}}(\boldsymbol{\mathcal{X}}_\theta) = \tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}}) + c_v\vartheta; \quad \tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}}) = \tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}}, \theta = \theta_0); \quad \vartheta = \theta - \theta_0; \tag{A.22}$$

Here, ϑ represents the temperature change (with respect the reference temperature θ_0) and the term $\tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}})$ represents the internal energy per unit reference volume caused by the deformation after the temperature has been allowed to return back to the reference value θ_0. Since the terms $\mathscr{E}(\boldsymbol{X}, t)$ and $\tilde{\mathcal{E}}(\boldsymbol{\mathcal{X}}_\theta)$ are used to denote the same energy (with different functional dependency), and recalling $\mathscr{E}(\boldsymbol{X}, t) = E_T - \frac{1}{2\rho_0}(\boldsymbol{p}\cdot\boldsymbol{p})$ rearrangement of Equation (A.22) gives the temperature update as:

$$\hat{\theta}(\boldsymbol{p}, \boldsymbol{\mathcal{X}}_E) = \theta_0 + \frac{1}{c_v}\left(\tilde{\mathcal{E}}(\boldsymbol{\mathcal{X}}_\theta) - \tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}})\right); \tag{A.23}$$

$$= \theta_0 + \frac{1}{c_v}\left(E_T - \frac{1}{2\rho_0}\boldsymbol{p}\cdot\boldsymbol{p} - \tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}})\right), \tag{A.24}$$

where $\boldsymbol{\mathcal{X}}_E = \{\boldsymbol{\mathcal{X}}, E_T\}$. Equation (A.23) provides an expression of the temperature in terms of the linear momentum, the triplet of deformation measures and the total energy density, namely, $\theta(\boldsymbol{X}, t) = \hat{\theta}(\boldsymbol{p}, \boldsymbol{\mathcal{X}}_E)$. Moreover, noticing Equation (A.19), it is also possible to obtain a similar relationship between the entropy density η and the set $\boldsymbol{\mathcal{X}}_E$, namely, $\eta(\boldsymbol{X}, t) = \hat{\eta}(\boldsymbol{p}, \boldsymbol{\mathcal{X}}_E)$, that is:

$$\hat{\eta}(\boldsymbol{p}, \boldsymbol{\mathcal{X}}_E) = \tilde{\eta}_0(\boldsymbol{\mathcal{X}}) + c_v \ln\left(\frac{\hat{\theta}(\boldsymbol{p}, \boldsymbol{\mathcal{X}}_E)}{\theta_0}\right). \tag{A.25}$$

A final useful relationship necessary to compute the conjugate stresses $\{\boldsymbol{\Sigma}_{\boldsymbol{F}}, \boldsymbol{\Sigma}_{\boldsymbol{H}}, \Sigma_J\}$ in Equation (A.7) is that of the internal energy density, namely:

$$\hat{\mathcal{E}}(\boldsymbol{p}, \boldsymbol{\mathcal{X}}_E) = E_T - \frac{1}{2\rho_0}\boldsymbol{p}\cdot\boldsymbol{p}, \tag{A.26}$$

where $\hat{\mathcal{E}}(\boldsymbol{p}, \boldsymbol{\mathcal{X}}_E)$ represents an alternative functional representation of the internal energy $\mathcal{E}(\boldsymbol{\mathcal{X}}_\eta)$. Finally, in order to complete the definition of the thermo-elastic constitutive model (refer to Equations (A.23), (A.25) and (A.26)), it is necessary to provide suitable functional expressions for $\tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}})$ and $\tilde{\eta}_0(\boldsymbol{\mathcal{X}})$, which will depend on the spe-

cific material under consideration. Following [117], in order to guarantee the existence of real wave speeds (hyperbolicity) in the material for the entire thermo-elastic deformation process, that is regardless of the amount of deformation $\boldsymbol{\mathcal{X}}$ and thermal state η (or θ), a sufficient condition is that of selecting $\tilde{\mathcal{E}}_0(\boldsymbol{\mathcal{X}})$ and $\tilde{\eta}_0(\boldsymbol{\mathcal{X}})$ to be convex in $\boldsymbol{\mathcal{X}}$, that is both functions shall be polyconvex. For further details, refer to [117]. Two well-established thermo-elastic models derived from universally polyconvex strain energy functions will be introduced, namely modified entropic elasticity and Mie-Grüneisen. [1]

A.3.1 Modified Entropic Elasticity (MEE) Model

The internal energy density at reference temperature θ_0 for this material is typically given by a purely volumetric function as [117] :

$$\tilde{\mathcal{E}}_0^{MEE}(J) = c_v \Gamma_0 \theta_0 (J-1), \tag{A.27}$$

here Γ_0 is a positive material constant. In addition, the entropy function at reference temperature θ_0 is given by:

$$\tilde{\eta}_0^{MEE}(\boldsymbol{\mathcal{X}}) = \frac{1}{\theta_0}\left(\tilde{\mathcal{E}}_0^{MEE}(J) - \psi_0^{MEE}(\boldsymbol{\mathcal{X}})\right), \tag{A.28}$$

where $\psi_0^{MEE}(\boldsymbol{\mathcal{X}})$ is the Helmholtz's free energy function at reference temperature θ_0, where a possible deviatoric-volumetric expression of the Mooney-Rivlin type is:

$$\psi_0^{MEE}(\boldsymbol{\mathcal{X}}) = \zeta_0\left(J^{-2/3}(\boldsymbol{F}:\boldsymbol{F}) - 3\right) + \xi_0\left(J^{-2}(\boldsymbol{H}:\boldsymbol{H})^{3/2} - 3\sqrt{3}\right) + \frac{\kappa_0}{2}(J-1)^2, \tag{A.29}$$

where $\{\zeta_0, \xi_0, \kappa_0\}$ are material parameters. Notice that if $\xi_0 = 0$, the material degenerates to a neo-Hookean type of model (refer to Section 2.5). Typically, these material parameters and Γ_0 introduced above, are calibrated against those of linear elasticity, namely, shear modulus μ, bulk modulus κ and thermal expansion coefficient α as [108]:

$$\mu = 2\zeta_0 + 3\sqrt{3}\xi_0; \quad \kappa = \kappa_0; \quad \alpha = \frac{c_v \Gamma_0}{3\kappa}. \tag{A.30}$$

[1] Refer to Bonet *et al.*[117] where hyperbolicity is demonstrated for the two thermo-elastic constitutive models presented here where the closed-form computation of bounds for the propagating wave speeds is also shown.

A.3.2 Mie-Grüneisen (MG) Model

A plausible deviatoric-volumetric expression for the internal energy density of the Mooney-Rivlin type at reference temperature θ_0 for this material can be given as:

$$\tilde{\mathcal{E}}_0^{MG} = \zeta_0 \left(J^{-2/3} \left(\boldsymbol{F} : \boldsymbol{F} \right) - 3 \right) + \gamma_0 \left(J^{-2} \left(\boldsymbol{H} : \boldsymbol{H} \right)^{3/2} - 3\sqrt{3} \right) + \frac{\chi_0}{2} \left(J - 1 \right)^2 + c_v \theta_0 \Gamma_0 \left(J - 1 \right), \tag{A.31}$$

where $\{\zeta_0, \gamma_0, \chi_0\}$ are material parameters (if $\gamma_0 = 0$, the material degenerates to a neo-Hookean type of model) and the entropy function at reference temperature θ_0 is given by:

$$\tilde{\eta}_0^{MG} \left(\boldsymbol{\mathcal{X}} \right) = c_v \Gamma_0 \left(\frac{J^q - 1}{q} \right), \tag{A.32}$$

where q is a dimensionless coefficient that varies from zero (for a perfect gas) to one (for solid materials). The material parameters can be calibrated against those of linear elasticity, namely, shear modulus μ, bulk modulus κ and thermal expansion coefficient α as:

$$\mu = 2\zeta_0 + 3\sqrt{3}\gamma_0; \quad \kappa = \chi_0 + c_v \theta_0 \Gamma_0 \left(1 - q \right); \quad \alpha = \frac{c_v \Gamma_0}{3\kappa}. \tag{A.33}$$

A.4 Extension to Thermo-Mechanical Contact

With the theory presented in this Appendix, the proposed method can be extended to thermo-elasticity by introducing the conservation of energy from Equation (A.2) into the first order framework from Section 2.4 as an additional conservation equation and the system is then closed by using an appropriate thermo-mechanical constitutive model such as those presented in Section A.3 as presented by Ghavamian *et al.* [116] and Bonet *et al.* [117]. To extend this to thermo-mechanical contact, a new set of derived contact conditions are required to determine the temperature interaction between multiple bodies and would be expanded expressions of those from Section 3.3 to include temperature effects for the contact fluxes. Furthermore, the second law of thermodynamics presented in Section 3.5, used to assess the entropy production of the numerical scheme, would now be expressed in terms of the so called ballistic free energy. This ballistic energy is simply the total energy presented in Equation (3.46) plus an additional thermal conduction term [21]. This addition would therefore require a reassessment of the entropy production of the semi-discrete system of equations to ensure discrete satisfaction of the second law of thermodynamics.

Appendix B

Mesh Generation Example

B.1 Preliminaries

In Chapter 5, the computational implementation of the proposed method in open-source software OpenFOAM was discussed. This included an example problem setup for the purpose built OpenFOAM solver for explicit multi-body contact dynamics. This appendix contains the source code required to generate the mesh for this example problem setup, for full details refer to Chapter 5. Listing B.1 defines the geometry and mesh for body 1 while Listing B.2 defines the geometry and mesh for body 2 which can be opened in open-source mesh generator gmsh [145] in order to create the relevant mesh (`*.msh`) files. Once created, the mesh files can be converted to the OpenFOAM format by using the executable utility in Listing B.3, which requires the two `createPatchDict` files in Listings B.4 and B.5 to be saved in the relevant system sub-directories.

B.2 Source Code

```
// Define geometry
SetFactory("OpenCASCADE");
Box(1) = {0, -0.5, -0.5, 10, 1, 1};

// Define structured lines
Transfinite Line {3, 4, 1, 2} = 9 Using Progression 1;
Transfinite Line {6, 8, 5, 7} = 9 Using Progression 1;
Transfinite Line {11, 12, 10, 9} = 81 Using Progression 1;

// Define structured surfaces
Transfinite Surface {1} = {4, 3, 2, 1};
Transfinite Surface {2} = {7, 5, 6, 8};
Transfinite Surface {6} = {3, 1, 5, 7};
Transfinite Surface {4} = {8, 4, 3, 7};
Transfinite Surface {3} = {5, 6, 2, 1};
Transfinite Surface {5} = {4, 8, 6, 2};
```

```
// Define structured volume
Transfinite Volume{1} = {2, 4, 3, 1, 6, 8, 7, 5};

// Assign volume name
Physical Volume("body1") = {1};
```

Listing B.1: gmsh script - Body 1

```
// Define geometry
SetFactory("OpenCASCADE");
Box(1) = {10.01, -0.5, -0.5, 10, 1, 1};

// Define structured lines
Transfinite Line {3, 4, 1, 2} = 9 Using Progression 1;
Transfinite Line {6, 8, 5, 7} = 9 Using Progression 1;
Transfinite Line {11, 12, 10, 9} = 81 Using Progression 1;

// Define structured surfaces
Transfinite Surface {1} = {4, 3, 2, 1};
Transfinite Surface {2} = {7, 5, 6, 8};
Transfinite Surface {6} = {3, 1, 5, 7};
Transfinite Surface {4} = {8, 4, 3, 7};
Transfinite Surface {3} = {5, 6, 2, 1};
Transfinite Surface {5} = {4, 8, 6, 2};

// Define structured volume
Transfinite Volume{1} = {2, 4, 3, 1, 6, 8, 7, 5};

// Assign volume name
Physical Volume("body2") = {1};
```

Listing B.2: gmsh script - Body 2

```
#!/bin/bash
cd ${0%/*} || exit 1 # Run from this directory

# Remove old Meshes
echo -e "\n
Remove old meshes
------------------------------------------------------------------------------
"
rm -r constant/polyMesh
rm -r 0/polyMesh
rm -r constant/body1/polyMesh
rm -r constant/body2/polyMesh

# Generate mesh
# Body1
echo -e "\n
Generate body1 mesh
------------------------------------------------------------------------------
"

gmshToFoam body1.msh > log.mesh
autoPatch 90 -overwrite >> log.mesh
mv constant/polyMesh constant/body1/polyMesh
createPatch -overwrite -region body1 >> log.mesh
```

```
26 # Body2
   echo -e "\n
28 Generate body2 mesh
   ------------------------------------------------------------------------------
30 "

32 gmshToFoam body2.msh >> log.mesh
   autoPatch 90 -overwrite >> log.mesh
34 mv constant/polyMesh constant/body2/polyMesh
   createPatch -overwrite -region body2 >> log.mesh
36
   echo -e "\n
38 Meshing complete
   ------------------------------------------------------------------------------
```

Listing B.3: createRegionMeshes

```
   *---------------------------------*- C++ -*----------------------------------*\
2    =========                 |
     \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
4     \\    /   O peration     | Website:  https://openfoam.org
       \\  /    A nd           | Version:  6
6       \\/     M anipulation  |
   \*---------------------------------------------------------------------------*/
8  FoamFile
   {
10     version     2.0;
       format      ascii;
12     class       dictionary;
       object      createPatchDict;
14 }

16 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //


18
   // Patches to create.
20 patches
   (
22     {
           name free;
24         patchInfo
           {
26             type patch;
           }
28         constructFrom patches;
           patches (auto0 auto1 auto2 auto4 auto5);
30     }
       {
32         name contact;
           patchInfo
34         {
               type patch;
36         }
           constructFrom patches;
38         patches (auto3);
       }
40 );
```

```
42 // ************************************************************************* //
```

Listing B.4: createPatchDict - Body 1

```
*---------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      createPatchDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// Patches to create.
patches
(
    {
        name free;
        patchInfo
        {
            type patch;
        }
        constructFrom patches;
        patches (auto0 auto1 auto3 auto4 auto5);
    }
    {
        name contact;
        patchInfo
        {
            type patch;
        }
        constructFrom patches;
        patches (auto2);
    }
);

// ************************************************************************* //
```

Listing B.5: createPatchDict - Body 2

Appendix C

OpenFOAM Input Files Example

C.1 Preliminaries

In Chapter 5, the computational implementation of the proposed method in open-source software OpenFOAM was discussed. This included an example problem setup for the purpose built OpenFOAM solver for explicit multi-body contact dynamics. This appendix contains the relevant files in order to define the problem. Listing C.1 defines the initial linear momentum for body 1, while Listing C.2 defines the initial linear momentum for body 2. The required `regionProperties` file, which defines the region names, is provided in Listing C.3. The `materialProperties` file for both bodies is provided in Listing C.4 and must be stored in both region sub-directories. The source files for the system directory are provided in Listing C.5 for the control dictionary, Listing C.6 for the finite volume solution dictionary and Listing C.7 for the finite volume schemes. Lastly, the `run` executable file is provided in Listing C.8 which runs the purpose built OpenFOAM solver.

C.2 Source Code

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       pointVectorField;
    location    "0";
```

```
    object      lm;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -2 -1 0 0 0 0];

internalField   uniform (0.001 0 0);

boundaryField
{
    free
    {
    type  zeroGradient;
    }

    contact
    {
      type  calculated;
    }
}

// ************************************************************************* //
```

Listing C.1: lm.orig - Body 1

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       pointVectorField;
    location    "0";
    object      lm;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -2 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    free
    {
    type  zeroGradient;
    }

    contact
    {
      type  calculated;
    }
}
```

```
34
// ************************************************************************* //
```

Listing C.2: lm.orig - Body 2

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      regionProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

regions
(
    bodies      (body1 body2)
);

// ************************************************************************* //
```

Listing C.3: regionProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      mechanicalProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
materialModel neoHookean;

rho    rho  [1 -3 0 0 0 0 0]  0.01;
E      E    [1 -1 -2 0 0 0 0] 100;
nu     nu   [0 0 0 0 0 0 0]   0.0;
s      s    [0 0 0 0 0 0 0]   0.0;
frictionCoef    frictionCoef  [0 0 0 0 0 0 0]   0.0;

// ************************************************************************* //
```

Listing C.4: materialProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application solidVertexMBFoam;

startFrom latestTime;

startTime 0;

stopAt endTime;

endTime 0.7;

timeStepping variable;

RKstages 2;

cfl 0.3;

deltaT 0;

writeControl runTime;

writeInterval 1e-3;

purgeWrite 0;

writeFormat ascii;

writePrecision 8;

writeCompression off;

timeFormat general;

timePrecision 6;

graphFormat raw;

runTimeModifiable false;

// ************************************************************************* //
```

Listing C.5: controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

JconservationLaw                on;
HconservationLaw                off;
angularMomentumConservation     off;
limiter                         on;
reconstructContactBoundary    on;
quasistatic             off;



// ************************************************************************* //
```

Listing C.6: fvSolution

```

/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  6
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

gradSchemes
{
    default         leastSquares;
}

divSchemes
{
      default         none;
}

```

```
laplacianSchemes
{
    default         none;
}

interpolationSchemes
{
    default         linear;
}

// ************************************************************************* //
```

Listing C.7: fvSchemes

```
#!/bin/bash
cd ${0%/*} || exit 1 # Run from this directory

# OpenFOAM functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
solver=`getApplication`

# Run solver
$solver > log.$solver &
tail -f log.$solver
```

Listing C.8: run

Bibliography

[1] Y. P. Liu, "Simulation of Vehicles Frontal Crash with Dummy Inside," *Advanced Materials Research*, vol. 760-762, pp. 1244–1249, 2013. DOI: 10.4028/www.scientific.net/AMR.760-762.1244.

[2] W. Gao, X. He, J. Wang, Y. Feng, and C. Wang, "Numerical investigation of oblique impact behavior of a helmeted headform on a windshield considering failure," *Thin-Walled Structures*, vol. 171, p. 108 722, 2022. DOI: https://doi.org/10.1016/j.tws.2021.108722.

[3] S. Heimbs, "Computational methods for bird strike simulations: A review," *Comput. Struct.*, vol. 89, no. 23-24, pp. 2093–2112, 2011. DOI: 10.1016/j.compstruc.2011.08.007.

[4] J. Berkley, G. Turkiyyah, D. Berg, M. Ganter, and S. Weghorst, "Real-time finite element modeling for surgery simulation: An application to virtual suturing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 3, pp. 314–325, 2004. DOI: 10.1109/TVCG.2004.1272730.

[5] S. F. Johnsen *et al.*, "NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 10, no. 7, pp. 1077–1095, 2015. DOI: 10.1007/s11548-014-1118-5.

[6] G. Maliaris, C. Gakias, M. Malikoutsakis, and G. Savaidis, "A fem-based 2d model for simulation and qualitative assessment of shot-peening processes," *Materials (Basel).*, vol. 14, no. 11, 2021. DOI: 10.3390/ma14112784.

[7] H. Wang, S. Ding, T. Taylor, and J. Yanagimoto, "Cold Rolling Texture Prediction Using Finite Element Simulation with Zooming Analysis," *Materials (Basel).*, vol. 14, no. 22, p. 6909, 2021. DOI: 10.3390/ma14226909.

[8] R. F. Shepherd *et al.*, "Multigait soft robot," *Proc. Natl. Acad. Sci.*, vol. 108, no. 51, pp. 20 400–20 403, 2011. DOI: 10.1073/pnas.1116564108.

[9] Insurance Institute For Highway Safety, *Frontal small-overlap crash test of a 2012 honda odyssey.* Licensed under Creative Commons Attribution-Share Alike 4.0 International. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/., 2012. [Online]. Available: http://www.iihs.org/.

[10] European Space Agency, *Hypervelocity impact*. Licensed under Creative Commons Attribution-Share Alike 3.0 International. To view a copy of this license, visit https://creativecommons.org/licenses/by-sa/3.0/igo/deed.en., 2013. [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2013/04/Hypervelocity_Impact.

[11] Sasin Tipchai, Licensed under Creative Commons 1.0 Universal (CC0 1.0) Public Domain Dedication. To view a copy of this license, visit https://creativecommons.org/publicdomain/zero/1.0/., 2016. [Online]. Available: https://pixabay.com/photos/surgery-hospital-doctor-care-1822458/.

[12] T. Belytschko, W. K. Liu, B. Moran, and K. I. Elkhodary, *Nonlinear Finite Elements for Continua and Structures*, 2nd. Chichester: John Wiley & Sons, 2014, p. 834.

[13] J. Barber, *Contact Mechanics* (Solid Mechanics and Its Applications). Springer International Publishing, 2018, vol. 250, p. 585. DOI: 10.1007/978-3-319-70939-0.

[14] C. Hirsch, *Numerical Computation of Internal and External Flows*, 2nd. Elsevier, 2007, vol. 1, p. 696. DOI: 10.1016/B978-0-7506-6594-0.X5037-1.

[15] P. Wriggers, *Computational Contact Mechanics*, 2nd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, p. 518. DOI: 10.1007/978-3-540-32609-0.

[16] *ABAQUS User's Manual*, English. United States: Dassault Systèmes Simulia Corp, 2018.

[17] J. O. Hallquist, *LS-DYNA® theory manual*. 2006.

[18] K.-J. Bathe, *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[19] "The finite element method: Its basis and fundamentals," in *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, O. Zienkiewicz, R. Taylor, and J. Zhu, Eds., Seventh Edition, Oxford: Butterworth-Heinemann, 2013, p. i. DOI: https://doi.org/10.1016/B978-1-85617-633-0.00019-8.

[20] N. M. Newmark, "A method of computation for structural dynamics," *Journal of the Engineering Mechanics Division*, vol. 85, no. 3, pp. 67–94, 1959. DOI: 10.1061/JMCEA3.0000098.

[21] J. Bonet, A. J. Gil, and R. D. Wood, *Nonlinear Solid Mechanics for Finite Element Analysis: Dynamics*. Cambridge: Cambridge University Press, 2021.

[22] M. A. Dokainish and K. Subbaraj, "A survey of direct time-integration methods in computational structural dynamics—I. Explicit methods," *Computers and Structures*, vol. 32, no. 6, pp. 1371–1386, 1989. DOI: 10.1016/0045-7949(89)90314-3.

[23] N. P. Weatherill and O. Hassan, "Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints," *Inter-*

national Journal for Numerical Methods in Engineering, vol. 37, no. 12, pp. 2005–2039, 1994. DOI: 10.1002/nme.1620371203.

[24] D. J. Payen and K. J. Bathe, "Improved stresses for the 4-node tetrahedral element," *Computers and Structures*, vol. 89, no. 13-14, 2011. DOI: 10.1016/j.compstruc.2011.02.009.

[25] D. J. Payen and K. J. Bathe, "A stress improvement procedure," *Computers and Structures*, vol. 112-113, 2012. DOI: 10.1016/j.compstruc.2012.07.006.

[26] H. M. Hilber, T. J. Hughes, and R. L. Taylor, "Improved numerical dissipation for time integration algorithms in structural dynamics," *Earthquake Engineering & Structural Dynamics*, 1977.

[27] W. L. Wood, M. Bossak, and O. C. Zienkiewicz, "An alpha modification of Newmark's method," *International Journal for Numerical Methods in Engineering*, 1980.

[28] C. R. Dohrmann, M. W. Heinstein, J. Jung, S. W. Key, and W. R. Witkowski, "Node-based uniform strain elements for three-node triangular and four-node tetrahedral meshes," *International Journal for Numerical Methods in Engineering*, vol. 47, no. 9, 2000. DOI: 10.1002/(SICI)1097-0207(20000330)47:9<1549::AID-NME842>3.0.CO;2-K.

[29] J. Bonet, H. Marriott, and O. Hassan, "An averaged nodal deformation gradient linear tetrahedral element for large strain explicit dynamic applications," *Communications in Numerical Methods in Engineering*, 2001.

[30] F. M. Pires, E. A. de Souza Neto, and J. L. de la Cuesta Padilla, "An assessment of the average nodal volume formulation for the analysis of nearly incompressible solids under finite strains," *Communications in Numerical Methods in Engineering*, 2004.

[31] E. A. de Souza Neto, D. Peri, and D. R. J. Owen, *Computational Methods for Plasticity*. Chichester, UK: John Wiley & Sons, Ltd, 2008, p. 814. DOI: 10.1002/9780470694626.

[32] Y. Onishi and K. Amaya, "A locking-free selective smoothed finite element method using tetrahedral and triangular elements with adaptive mesh rezoning for large deformation problems," *International Journal for Numerical Methods in Engineering*, 2014.

[33] J. Bonet and R. D. Wood, *Nonlinear continuum mechanics for finite element analysis, 2nd edition*, 2nd. Cambridge University Press, 2008.

[34] J. Bonet and A. J. Burton, "A simple average nodal pressure tetrahedral element for incompressible and nearly incompressible dynamic explicit applications," *Communications in Numerical Methods in Engineering*, 1998. DOI: 10.1002/(SICI)1099-0887(199805)14:5<437::AID-CNM162>3.0.CO;2-W.

[35] J. R. Williams and R. O'connor, "A linear complexity intersection algorithm for discrete element simulation of arbitrary geometries," *Engineering Computations*, 1995.

[36] J. R. Williams and R. O'Connor, "Discrete Element Simulation and the Contact Problem," *Archives of Computational Methods in Engineering*, vol. 6, no. 4, 1999. DOI: 10.1007/BF02818917.

[37] C. F. Li, Y. T. Feng, and D. R. Owen, "SMB: Collision detection based on temporal coherence," *Comput. Methods Appl. Mech. Eng.*, vol. 195, no. 19-22, pp. 2252–2269, 2006. DOI: 10.1016/j.cma.2005.05.003.

[38] K Han, Y. T. Feng, and D. R. J. Owen, "Performance comparisons of tree-based and cell-based contact detection algorithms," *Engineering Computations*, vol. 24, no. 2, pp. 165–181, 2007. DOI: 10.1108/02644400710729554.

[39] D. J. Benson and J. O. Hallquist, "A single surface contact algorithm for the post-buckling analysis of shell structures," *Computer Methods in Applied Mechanics and Engineering*, 1990.

[40] C. Kane, E. A. Repetto, M. Ortiz, and J. E. Marsden, "Finite element analysis of nonsmooth contact," *Computer Methods in Applied Mechanics and Engineering*, 1999.

[41] Z. Zheng, M. Zang, S. Chen, and C. Zhao, "An improved 3D DEM-FEM contact detection algorithm for the interaction simulations between particles and structures," *Powder Technol.*, vol. 305, pp. 308–322, 2017, ISSN: 0032-5910. DOI: 10.1016/J.POWTEC.2016.09.076.

[42] Y. T. Feng, "An energy-conserving contact theory for discrete element modelling of arbitrarily shaped particles: Contact volume based model and computational issues," *Comput. Methods Appl. Mech. Eng.*, vol. 373, p. 113 493, 2021, ISSN: 0045-7825. DOI: 10.1016/J.CMA.2020.113493.

[43] Y. T. Feng, "An effective energy-conserving contact modelling strategy for spherical harmonic particles represented by surface triangular meshes with automatic simplification," *Comput. Methods Appl. Mech. Eng.*, vol. 379, p. 113 750, 2021, ISSN: 0045-7825. DOI: 10.1016/J.CMA.2021.113750.

[44] A. Francavilla and O. C. Zienkiewicz, "A note on numerical computation of elastic contact problems," *Int. J. Numer. Methods Eng.*, vol. 9, no. 4, pp. 913–924, 1975. DOI: 10.1002/nme.1620090410.

[45] T. J. Hughes, R. L. Taylor, J. L. Sackman, A. Curnier, and W. Kanoknukulchai, "A finite element method for a class of contact-impact problems," *Comput. Methods Appl. Mech. Eng.*, vol. 8, no. 3, pp. 249–276, 1976. DOI: 10.1016/0045-7825(76)90018-9.

[46] J. O. Hallquist, G. L. Goudreau, and D. J. Benson, "Sliding interfaces with contact-impact in large-scale Lagrangian computations," *Comput. Methods Appl. Mech. Eng.*, vol. 51, no. 1-3, pp. 107–137, 1985. DOI: 10.1016/0045-7825(85)90030-1.

[47] P. Papadopoulos, R. E. Jones, and J. M. Solberg, "A Novel Finite Element Formulation For Frictionless Contact Problems," Tech. Rep., 1995, pp. 2603–2617.

[48] G. Zavarise and L. De Lorenzis, "The node-to-segment algorithm for 2D frictionless contact: Classical formulation and special cases," *Comput. Methods Appl. Mech. Eng.*, vol. 198, no. 41-44, pp. 3428–3451, 2009. DOI: 10.1016/j.cma.2009.06.022.

[49] C. Hesch and P. Betsch, "A comparison of computational methods for large deformation contact problems of flexible bodies," *Angew. Math. Mech*, vol. 86, no. 10, pp. 818–827, 2006. DOI: 10.1002/zamm.200610289.

[50] G. Zavarise and P. Wriggers, "A segment-to-segment contact strategy," *Math. Comput. Model.*, vol. 28, no. 4-8, pp. 497–515, 1998. DOI: 10.1016/S0895-7177(98)00138-1.

[51] F. Cirak and M. West, "Decomposition contact response (DCR) for explicit finite element dynamics," *Int. J. Numer. Methods Eng.*, vol. 64, no. 8, pp. 1078–1110, 2005. DOI: 10.1002/nme.1400.

[52] T. Belytschko and M. O. Neal, "Contact-impact by the pinball algorithm with penalty and Lagrangian methods," *Int. J. Numer. Methods Eng.*, vol. 31, no. 3, pp. 547–572, 1991. DOI: 10.1002/nme.1620310309.

[53] P Wriggers and J. Simo, "A note on tangent stiffness for fully nonlinear contact problems," Tech. Rep., 1985, pp. 199–203.

[54] R. Abedi and R. B. Haber, "Riemann solutions and spacetime discontinuous Galerkin method for linear elastodynamic contact," *Computer Methods in Applied Mechanics and Engineering*, 2014.

[55] L. Taylor and D. Flanagan, "PRONTO 3D: A three-dimensional transient solid dynamics program," Tech. Rep. March, 1989.

[56] D. J. Benson, "Computational methods in Lagrangian and Eulerian hydrocodes," *Comput. Methods Appl. Mech. Eng.*, vol. 99, no. 2-3, pp. 235–394, 1992. DOI: 10.1016/0045-7825(92)90042-I.

[57] M. W. Heinstein, F. J. Mello, S. W. Attaway, and T. A. Laursen, "Contact - Impact modeling in explicit transient dynamics," *Computer Methods in Applied Mechanics and Engineering*, 2000.

[58] J. VonNeumann and R. D. Richtmyer, "A method for the numerical calculation of hydrodynamic shocks," *Journal of Applied Physics*, 1950.

[59] P. Wriggers, W. T. Rust, and B. D. Reddy, "A virtual element method for contact," *Comput. Mech.*, vol. 58, no. 6, pp. 1039–1050, 2016. DOI: 10.1007/s00466-016-1331-x.

[60] B. K. Zimmerman and G. A. Ateshian, "A Surface-to-Surface Finite Element Algorithm for Large Deformation Frictional Contact in febio," *J. Biomech. Eng.*, vol. 140, no. 8, pp. 1–15, 2018. DOI: 10.1115/1.4040497.

[61] W. Xing, J. Zhang, C. Song, and F. Tin-Loi, "A node-to-node scheme for three-dimensional contact problems using the scaled boundary finite element method," *Comput. Methods Appl. Mech. Eng.*, vol. 347, pp. 928–956, 2019. DOI: 10.1016/j.cma.2019.01.015.

[62] F. B. Belgacem, P. Hild, and L. Patrick, "Approximation du problème de contact unilatéral par la méthode des éléments finis avec joints," *Comptes Rendus l'Académie des Sci. - Ser. I - Math.*, vol. 324, no. 1, pp. 123–127, 1997. DOI: 10.1016/s0764-4442(97)80115-2.

[63] P. Hild, "Numerical implementation of two nonconforming finite element methods for unilateral contact," *Comput. Methods Appl. Mech. Eng.*, vol. 184, no. 1, pp. 99–123, 2000. DOI: 10.1016/S0045-7825(99)00096-1.

[64] R. H. Krause and B. I. Wohlmuth, "A Dirichlet-Neumann type algorithm for contact problems with friction," *Comput. Vis. Sci.*, vol. 5, no. 3, pp. 139–148, 2002. DOI: 10.1007/s00791-002-0096-2.

[65] M. A. Puso, "A 3D mortar method for solid mechanics," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 3, pp. 315–336, 2004.

[66] R. Mlika, Y. Renard, and F. Chouly, "An unbiased Nitsche's formulation of large deformation frictional contact and self-contact," *Comput. Methods Appl. Mech. Eng.*, vol. 325, pp. 265–288, 2017. DOI: 10.1016/J.CMA.2017.07.015.

[67] S. Hüeber and B. I. Wohlmuth, "A primal–dual active set strategy for non-linear multibody contact problems," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 27-29, pp. 3147–3166, 2005. DOI: 10.1016/J.CMA.2004.08.006.

[68] K. A. Fischer and P. Wriggers, "Mortar based frictional contact formulation for higher order interpolations using the moving friction cone," *Comput. Methods Appl. Mech. Eng.*, vol. 195, no. 37-40, pp. 5020–5036, 2006. DOI: 10.1016/j.cma.2005.09.025.

[69] T. Cichosz and M. Bischoff, "Consistent treatment of boundaries with mortar contact formulations using dual Lagrange multipliers," *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 9-12, pp. 1317–1332, 2011. DOI: 10.1016/J.CMA.2010.11.004.

[70] L De Lorenzis, P Wriggers, and G. Zavarise, "A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the aug-

mented Lagrangian method," *Comput. Mech.*, vol. 49, no. 1, pp. 1–20, 2012. DOI: 10.1007/s00466-011-0623-4.

[71] T. A. Laursen, M. A. Puso, and J. Sanders, "Mortar contact formulations for deformable–deformable contact: Past contributions and new extensions for enriched and embedded interface formulations," *Comput. Methods Appl. Mech. Eng.*, vol. 205-208, no. 1, pp. 3–15, 2012. DOI: 10.1016/J.CMA.2010.09.006.

[72] A. Popp, B. I. Wohlmuth, M. W. Gee, and W. A. Wall, "Dual quadratic mortar finite element methods for 3D finite deformation contact," *SIAM J. Sci. Comput.*, vol. 34, no. 4, 2012. DOI: 10.1137/110848190.

[73] M. Tur, E. Giner, F. J. Fuenmayor, and P. Wriggers, "2D contact smooth formulation based on the mortar method," *Comput. Methods Appl. Mech. Eng.*, vol. 247-248, pp. 1–14, 2012. DOI: 10.1016/J.CMA.2012.08.002.

[74] L. De Lorenzis, P. Wriggers, and T. J. Hughes, "Isogeometric contact: A review," *GAMM Mitteilungen*, vol. 37, no. 1, pp. 85–123, 2014. DOI: 10.1002/gamm.201410005.

[75] A. Popp and W. A. Wall, "Dual mortar methods for computational contact mechanics - Overview and recent developments," *GAMM Mitteilungen*, vol. 37, no. 1, pp. 66–84, 2014. DOI: 10.1002/gamm.201410004.

[76] S. Sitzmann, K. Willner, and B. I. Wohlmuth, "Variationally consistent quadratic finite element contact formulations for finite deformation contact problems on rough surfaces," *Finite Elem. Anal. Des.*, vol. 109, pp. 37–53, 2016. DOI: 10.1016/J.FINEL.2015.09.009.

[77] M. Hiermeier, W. A. Wall, and A. Popp, "A truly variationally consistent and symmetric mortar-based contact formulation for finite deformation solid mechanics," *Comput. Methods Appl. Mech. Eng.*, vol. 342, pp. 532–560, 2018. DOI: 10.1016/J.CMA.2018.07.020.

[78] M. Agrawal, A. Nandy, and C. S. Jog, "A hybrid finite element formulation for large-deformation contact mechanics," *Comput. Methods Appl. Mech. Eng.*, vol. 356, pp. 407–434, 2019. DOI: 10.1016/j.cma.2019.07.017.

[79] M. A. Puso and J. M. Solberg, "A dual pass mortar approach for unbiased constraints and self-contact," *Comput. Methods Appl. Mech. Eng.*, vol. 367, p. 113 092, 2020. DOI: 10.1016/J.CMA.2020.113092.

[80] S. Hartmann and E. Ramm, "A mortar based contact formulation for non-linear dynamics using dual Lagrange multipliers," *Finite Elem. Anal. Des.*, vol. 44, no. 5, pp. 245–258, 2008. DOI: 10.1016/J.FINEL.2007.11.018.

[81] L De Lorenzis, I Temizer, P Wriggers, and G Zavarise, "A large deformation frictional contact formulation using NURBS-based isogeometric analysis," *Int. J.*

Numer. METHODS Eng. Int. J. Numer. Meth. Engng, vol. 87, pp. 1278–1300, 2011. DOI: 10.1002/nme.3159.

[82] P. Farah, A. Popp, and W. A. Wall, "Segment-based vs. Element-based integration for mortar methods in computational contact mechanics," *Comput. Mech.*, vol. 55, no. 1, pp. 209–228, 2015. DOI: 10.1007/s00466-014-1093-2.

[83] T. Doca and F. M. Andrade Pires, "Finite element modeling of wear using the dissipated energy method coupled with a dual mortar contact formulation," *Comput. Struct.*, vol. 191, pp. 62–79, 2017. DOI: 10.1016/J.COMPSTRUC.2017.06.001.

[84] Y. Tong, M. Müller, and G. P. Ostermeyer, "Investigations on the dynamic influence of the contact angle on frictional sliding processes between rough surfaces using NURBS and mortar-based augmented Lagrangian method," *Tribol. Int.*, vol. 158, p. 106 889, 2021. DOI: 10.1016/J.TRIBOINT.2021.106889.

[85] P. Otto, L. De Lorenzis, and J. F. Unger, "Explicit dynamics in impact simulation using a NURBS contact interface," *Int. J. Numer. Methods Eng.*, vol. 121, no. 6, pp. 1248–1267, 2020. DOI: 10.1002/nme.6264.

[86] J Nitsche, "Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind," *Abhandlungen aus dem Math. Semin. der Univ. Hambg.*, vol. 36, no. 1, pp. 9–15, 1971. DOI: 10.1007/BF02995904.

[87] Becker, Roland, Hansbo, Peter, and Stenberg, Rolf, "A finite element method for domain decomposition with non-matching grids," *ESAIM: M2AN*, vol. 37, no. 2, pp. 209–225, 2003.

[88] P Wriggers and ·. G. Zavarise, "A formulation for frictionless contact problems using a weak form introduced by Nitsche," vol. 41, pp. 407–420, 2008. DOI: 10.1007/s00466-007-0196-4.

[89] F. Chouly and P. Hild, "A Nitsche-Based method for unilateral contact," vol. 51, no. 2, pp. 1295–1307, 2013.

[90] C. Annavarapu, M. Hautefeuille, and J. E. Dolbow, "A Nitsche stabilized finite element method for frictional sliding on embedded interfaces. Part I: Single interface," *Comput. Methods Appl. Mech. Eng.*, vol. 268, pp. 417–436, 2014. DOI: 10.1016/J.CMA.2013.09.002.

[91] F. Chouly, "An adaptation of Nitsche method to the Tresca friction problem," *J. Math. Anal. Appl.*, vol. 411, no. 1, pp. 329–339, 2014. DOI: 10.1016/J.JMAA.2013.09.019.

[92] F. Chouly *et al.*, "Symmetric and non-symmetric variants of Nitsche ' s method for contact problems in elasticity : theory and numerical experiments," 2015.

[93] A. Seitz, W. A. Wall, and ·. A. Popp, "Nitsche's method for finite deformation thermomechanical contact problems," *Comput. Mech.*, vol. 63, pp. 1091–1110, 2019. DOI: 10.1007/s00466-018-1638-x.

[94] F. Chouly, P. Hild, and Y. Renard, "A Nitsche finite element method for dynamic contact: 1. Space semi-discretization and time-marching schemes," *ESAIM Math. Model. Numer. Anal.*, vol. 49, no. 2, pp. 481–502, 2015. DOI: 10.1051/m2an/2014041.

[95] F. Chouly, P. Hild, and Y. Renard, "A Nitsche finite element method for dynamic contact: 2. Stability of the schemes and numerical experiments," *ESAIM Math. Model. Numer. Anal.*, vol. 49, no. 2, pp. 503–528, 2015. DOI: 10.1051/m2an/2014046.

[96] F. Chouly and Y. Renard, "Explicit Verlet time-integration for a Nitsche-based approximation of elastodynamic contact problems," *Model. Simul. Eng. Sci*, vol. 5, p. 31, 2018. DOI: 10.1186/s40323-018-0124-5.

[97] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002. DOI: 10.1017/CBO9780511791253.

[98] N. R. Morgan, M. A. Kenamond, D. E. Burton, T. C. Carney, and D. J. Ingraham, "An approach for treating contact surfaces in Lagrangian cell-centered hydrodynamics," *J. Comput. Phys.*, vol. 250, pp. 527–554, 2013. DOI: 10.1016/j.jcp.2013.05.015.

[99] R. Abedi and P. L. Clarke, "A computational approach to model dynamic contact and fracture mode transitions in rock," *Comput. Geotech.*, vol. 109, pp. 248–271, 2019. DOI: 10.1016/J.COMPGEO.2019.01.010.

[100] C. H. Lee, A. J. Gil, and J. Bonet, "Development of a cell centred upwind finite volume algorithm for a new conservation law formulation in structural dynamics," *Computers and Structures*, 2013.

[101] M. Aguirre, A. J. Gil, J. Bonet, and A. Arranz Carreño, "A vertex centred Finite Volume Jameson-Schmidt-Turkel (JST) algorithm for a mixed conservation formulation in solid dynamics," *Journal of Computational Physics*, 2014.

[102] C. H. Lee, A. J. Gil, and J. Bonet, "Development of a stabilised Petrov-Galerkin formulation for conservation laws in Lagrangian fast solid dynamics," *Computer Methods in Applied Mechanics and Engineering*, 2014.

[103] A. J. Gil, C. H. Lee, J. Bonet, and M. Aguirre, "A stabilised Petrov-Galerkin formulation for linear tetrahedral elements in compressible, nearly incompressible and truly incompressible fast dynamics," *Computer Methods in Applied Mechanics and Engineering*, 2014.

[104] I. A. Karim, C. H. Lee, A. J. Gil, and J. Bonet, "A two-step taylor-galerkin formulation for fast dynamics," *Engineering Computations (Swansea, Wales)*, 2014.

[105] M. Aguirre, A. J. Gil, J. Bonet, and C. H. Lee, "An upwind vertex centred Finite Volume solver for Lagrangian solid dynamics," *Journal of Computational Physics*, 2015.

[106] J. Bonet, A. J. Gil, C. H. Lee, M. Aguirre, and R. Ortigosa, "A first order hyperbolic framework for large strain computational solid dynamics. Part I: Total Lagrangian isothermal elasticity," *Computer Methods in Applied Mechanics and Engineering*, 2015.

[107] C. H. Lee, A. J. Gil, G. Greto, S. Kulasegaram, and J. Bonet, "A new Jameson–Schmidt–Turkel Smooth Particle Hydrodynamics algorithm for large strain explicit fast dynamics," *Computer Methods in Applied Mechanics and Engineering*, 2016.

[108] A. J. Gil, C. H. Lee, J. Bonet, and R. Ortigosa, "A first order hyperbolic framework for large strain computational solid dynamics. Part II: Total Lagrangian compressible, nearly incompressible and truly incompressible elasticity," *Computer Methods in Applied Mechanics and Engineering*, 2016.

[109] R. Ortigosa, A. J. Gil, and C. H. Lee, "A computational framework for large strain nearly and truly incompressible electromechanics based on convex multivariable strain energies," *Computer Methods in Applied Mechanics and Engineering*, 2016.

[110] J. Haider, C. H. Lee, A. J. Gil, and J. Bonet, "A first-order hyperbolic framework for large strain computational solid dynamics: An upwind cell centred Total Lagrangian scheme," *International Journal for Numerical Methods in Engineering*, 2017.

[111] C. H. Lee, A. J. Gil, O. I. Hassan, J. Bonet, and S. Kulasegaram, "A variationally consistent Streamline Upwind Petrov–Galerkin Smooth Particle Hydrodynamics algorithm for large strain solid dynamics," *Computer Methods in Applied Mechanics and Engineering*, 2017.

[112] J. Haider, C. H. Lee, A. J. Gil, A. Huerta, and J. Bonet, "An upwind cell centred Total Lagrangian finite volume algorithm for nearly incompressible explicit fast solid dynamic applications," *Computer Methods in Applied Mechanics and Engineering*, 2018.

[113] E. Garcia-Blanco, R. Ortigosa, A. J. Gil, C. H. Lee, and J. Bonet, "A new computational framework for electro-activation in cardiac mechanics," *Computer Methods in Applied Mechanics and Engineering*, 2019.

[114] O. I. Hassan, A. Ghavamian, C. H. Lee, A. J. Gil, J. Bonet, and F. Auricchio, "An upwind vertex centred finite volume algorithm for nearly and truly incompressible explicit fast solid dynamic applications: Total and Updated Lagrangian

formulations," *Journal of Computational Physics: X*, vol. 3, p. 100 025, 2019. DOI: 10.1016/j.jcpx.2019.100025.

[115] C. H. Lee, A. J. Gil, A. Ghavamian, and J. Bonet, "A Total Lagrangian upwind Smooth Particle Hydrodynamics algorithm for large strain explicit solid dynamics," *Computer Methods in Applied Mechanics and Engineering*, 2019.

[116] A. Ghavamian, C. H. Lee, A. J. Gil, J. Bonet, T. Heuzé, and L. Stainier, "An entropy-stable Smooth Particle Hydrodynamics algorithm for large strain thermo-elasticity," *Comput. Methods Appl. Mech. Eng.*, vol. 379, pp. 1–48, 2021. DOI: 10.1016/j.cma.2021.113736.

[117] J. Bonet, C. H. Lee, A. J. Gil, and A. Ghavamian, "A first order hyperbolic framework for large strain computational solid dynamics. Part III: Thermo-elasticity," *Comput. Methods Appl. Mech. Eng.*, vol. 373, 2021. DOI: 10.1016/j.cma.2020.113505.

[118] K. W. Low, C. H. Lee, A. J. Gil, J. Haider, and J. Bonet, "A parameter-free total Lagrangian smooth particle hydrodynamics algorithm applied to problems with free surfaces," *Comput. Part. Mech.*, vol. 8, no. 4, pp. 859–892, 2021. DOI: 10.1007/s40571-020-00374-x.

[119] P. Cardiff and I. Demirdžić, "Thirty Years of the Finite Volume Method for Solid Mechanics," *Arch. Comput. Methods Eng.*, vol. 28, no. 5, pp. 3721–3780, 2021. DOI: 10.1007/s11831-020-09523-0.

[120] OpenFOAM-Foundation, *OpenFOAM Documentation*, 2021.

[121] J. Bonet, A. J. Gil, and R. Ortigosa, "On a tensor cross product based formulation of large strain solid mechanics," *International Journal of Solids and Structures*, 2016.

[122] S. P. Timoshenko and J. Goodier, *Theory of Elasticity*, 2nd. London: McGraw-Hill, 1951.

[123] J. M. Ball, "Convexity conditions and existence theorems in nonlinear elasticity," *Arch. Ration. Mech. Anal.*, vol. 63, no. 4, pp. 337–403, 1976. DOI: 10.1007/BF00279992.

[124] J. Bonet, A. J. Gil, and R. Ortigosa, "A computational framework for polyconvex large strain elasticity," *Computer Methods in Applied Mechanics and Engineering*, vol. 283, pp. 1061–1094, 2015.

[125] M. A. Meyers, *Dynamic Behavior of Materials*. Wiley-InterSc, 1994, p. 474.

[126] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 2009.

[127] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. 2010.

[128] N. Abboud and G. Scovazzi, "Elastoplasticity with linear tetrahedral elements: A variational multiscale method," *Int. J. Numer. Methods Eng.*, vol. 115, no. 8, pp. 913–955, 2018. DOI: 10.1002/NME.5831.

[129] X. Zeng, G. Scovazzi, N. Abboud, O. Colomés, and S. Rossi, "A dynamic variational multiscale method for viscoelasticity using linear tetrahedral elements," *Int. J. Numer. Methods Eng.*, vol. 112, no. 13, pp. 1951–2003, 2017. DOI: 10.1002/NME.5591.

[130] N. Abboud and G. Scovazzi, "A variational multiscale method with linear tetrahedral elements for multiplicative viscoelasticity," *Mech. Res. Commun.*, vol. 112, p. 103 610, 2021. DOI: 10.1016/J.MECHRESCOM.2020.103610.

[131] J. Bonet, A. J. Gil, and R. D. Wood, *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*. Cambridge: Cambridge University Press, 2016. DOI: 10.1017/CBO9781316336144.

[132] P. R. de Campos, A. J. Gil, C. H. Lee, M. Giacomini, and J. Bonet, "A New Updated Reference Lagrangian Smooth Particle Hydrodynamics algorithm for isothermal elasticity and elasto-plasticity," *Comput. Methods Appl. Mech. Eng.*, vol. 392, p. 114 680, 2022. DOI: 10.1016/J.CMA.2022.114680.

[133] G. A. Holzapfel, *Nonlinear solid mechanics: A continuum approach for engineering*. Wiley and Sons, 2000.

[134] P. W. McDonald, "The computation of transonic flow through two-dimensional gas turbine cascades," American Society of Mechanical Engineers, Mar. 1971. DOI: 10.1115/71-GT-89.

[135] R. MACCORMACK and A. PAULLAY, "Computational efficiency achieved by time splitting of finite difference operators," in *10th Aerosp. Sci. Meet.*, Reston, Virigina: American Institute of Aeronautics and Astronautics, 1972. DOI: 10.2514/6.1972-154.

[136] A. W. RIZZI and M. INOUYE, "Time-Split Finite-Volume Method for Three-Dimensional Blunt-Body Flow," *AIAA J.*, vol. 11, no. 11, pp. 1478–1485, 1973. DOI: 10.2514/3.50614.

[137] J. Blasek, *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2005. DOI: 10.1016/B978-0-08-044506-9.X5000-0.

[138] S. Mazumder, "The Finite Volume Method (FVM)," in *Numer. Methods Partial Differ. Equ.* Elsevier, 2016, pp. 277–338. DOI: 10.1016/B978-0-12-849894-1.00006-8.

[139] D. J. Mavriplis, "Unstructured grid techniques," *Annu. Rev. Fluid Mech.*, vol. 29, pp. 473–514, 1997. DOI: 10.1146/annurev.fluid.29.1.473.

[140] K. A. Sørensen, O. Hassan, K. Morgan, and N. P. Weatherill, "A multigrid accelerated hybrid unstructured mesh method for 3D compressible turbulent flow," *Comput. Mech.*, vol. 31, no. 1-2 SPEC. Pp. 101–114, 2003. DOI: 10.1007/s00466-002-0397-9.

[141] H. Luo, J. Baum, and R. Lohner, "An improved finite volume scheme for compressible flows on unstructured grids," in *33rd Aerospace Sciences Meeting and Exhibit*, ser. Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, 1995.

[142] T. J. Barth and D. C. Jesperson, "The design and application of upwind schemes on unstructured meshes," *AIAA*, 1989.

[143] P. K. Sweby, "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws," *SIAM J. Numer. Anal.*, vol. 21, no. 5, pp. 995–1011, 1984. DOI: 10.1137/0721062.

[144] B. Stroustrup, *The C ++ Programming*, 4th. Pearson Education Limited, 2013, p. 1346.

[145] C. Geuzaine and J. F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *Int. J. Numer. Methods Eng.*, vol. 79, no. 11, pp. 1309–1331, 2009. DOI: 10.1002/nme.2579.

[146] M. Beaudoin and H. Jasak, "Development of a Generalized Grid Interface for Turbomachinery simulations with OpenFOAM," *Talk*, no. July, pp. 10–11, 2008.

[147] P. E. Farrell and J. R. Maddison, "Conservative interpolation between volume meshes by local Galerkin projection," *Computer Methods in Applied Mechanics and Engineering*, 2011. DOI: 10.1016/j.cma.2010.07.015.

[148] C. Wen-Hwa and T. Pwu, "Finite element analysis of elastodynamic sliding contact problems with friction," *Comput. Struct.*, vol. 22, no. 6, pp. 925–938, 1986. DOI: 10.1016/0045-7949(86)90153-7.

[149] F. F. Mahmoud, M. M. Hassan, and N. J. Salamon, "Dynamic contact of deformable bodies," *Comput. Struct.*, vol. 36, no. 1, pp. 169–181, 1990. DOI: 10.1016/0045-7949(90)90186-6.

[150] D. Sha, K. K. Tamma, and M. Li, "Robust explicit computational developments and solution strategies for impact problems involving friction," *Int. J. Numer. Methods Eng.*, vol. 39, no. 5, pp. 721–739, 1996. DOI: 10.1002/(SICI)1097-0207(19960315)39:5<721::AID-NME865>3.0.CO;2-J.

[151] T. A. Laursen and V. Chawla, "Design of energy conserving algorithms for frictionless dynamic contact problems," *International Journal for Numerical Methods in Engineering*, 1997. DOI: 10.1002/(SICI)1097-0207(19970315)40:5<863::AID-NME92>3.0.CO;2-V.

[152] J. M. Solberg and P. Papadopoulos, "A finite element method for contact/impact," *Finite Elem. Anal. Des.*, vol. 30, no. 4, pp. 297–311, 1998. DOI: 10.1016/S0168-874X(98)00041-9.

[153] A. Czekanski and S. A. Meguid, "Analysis of dynamic frictional contact problems using variational inequalities," *Finite Elem. Anal. Des.*, vol. 37, no. 11, pp. 861–879, 2001. DOI: 10.1016/S0168-874X(01)00072-5.

[154] A. Czekanski and S. A. Meguid, "Solution of dynamic frictional contact problems using nondifferentiable optimization," *Int. J. Mech. Sci.*, vol. 43, no. 6, pp. 1369–1386, 2001. DOI: 10.1016/S0020-7403(00)00095-3.

[155] G. Taylor, "The use of flat-ended projectiles for determining dynamic yield stress," *Proceedings of the Royal Society of London A*, 1948.

[156] A. E. Assie, M. Eltaher, and F. F. Mahmoud, "Modeling of viscoelastic contact-impact problems," *Appl. Math. Model.*, vol. 34, no. 9, pp. 2336–2352, 2010. DOI: 10.1016/j.apm.2009.11.001.

[157] X. Wang and X. Cheng, "Numerical analysis of a dynamic viscoplastic contact problem," *International Journal of Computer Mathematics*, vol. 99, no. 6, pp. 1178–1200, 2022. DOI: 10.1080/00207160.2021.1955107.

[158] P. Wriggers and C. Miehe, "Contact constraints within coupled thermomechanical analysis—A finite element model," *Comput. Methods Appl. Mech. Eng.*, vol. 113, no. 3-4, pp. 301–319, 1994. DOI: 10.1016/0045-7825(94)90051-5.

[159] M. Aouadi and M. I. Copetti, "Analytical and numerical results for a dynamic contact problem with two stops in thermoelastic diffusion theory," *ZAMM Zeitschrift fur Angew. Math. und Mech.*, vol. 96, no. 3, pp. 361–384, 2016. DOI: 10.1002/zamm.201400285.

[160] A. Seitz, W. A. Wall, and A. Popp, "A computational approach for thermo-elasto-plastic frictional contact based on a monolithic formulation using non-smooth nonlinear complementarity functions," *Adv. Model. Simul. Eng. Sci.*, vol. 5, no. 1, p. 5, 2018. DOI: 10.1186/s40323-018-0098-3.

[161] I. Demirdžić, E. Džaferović, and A. Ivanković, "Finite-volume approach to thermoviscoelasticity," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 47, no. 3, pp. 213–237, 2005. DOI: 10.1080/10407790590901675.

[162] G. Chourdakis, "A general openfoam adapter for the coupling library precice," 2017.

[163] M. E. Gurtin, E. Fried, and L. Anand, "The Mechanics and Thermodynamics of Continua," *Mech. Thermodyn. Contin.*, 2010. DOI: 10.1017/CBO9780511762956.