Swansea University
Prifysgol Abertawe

UPC

An upwind cell centred Finite Volume Method for large strain explicit solid dynamics in OpenFOAM

by

Jibran Haider

Submitted to the College of Engineering
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

at

Swansea University

December 2017

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. When correction services have been used, the extent and nature of the correction is clearly marked in a footnote(s). Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (candidate)

Date ....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date ....

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Antonio J. Gil and Dr. Chun Hean Lee who have always motivated me and provided excellent supervision throughout the course of this work. I would also like to thank Prof. Javier Bonet for encouraging me to work with OpenFOAM and for all the useful interactions we had during his presence in Swansea. Special thanks to Prof. Antonio Huerta for all the support and guidance he provided as part of my secondment in UPC BarcelonaTech. I am truly honoured and privileged that I got this oportunity to work and interact with all of you.

I am grateful to the European Commission EACEA Agency for providing me the financial assistance under the Framework Partnership Agreement 2013-0043 Erasmus Mundus Action 1b as a part of the EM Joint Doctorate Simulation in Engineering and Entrepreneurship Development (SEED). Moreover, this programme has given me the exposure and opportunity to undertake research in two top of the line universities in the field of computational mechanics which has consequently helped me to grow professionally.

I would also like to acknowledge the free and open source software community, in particular to OpenFOAM®, Linux, ParaView, Gmsh, OpenMPI, gnuplot and LaTeX projects. I would like to thank the OpenFOAM User Forum present on the website CFD online for providing answers to many OpenFOAM related programming questions.

I take this opportunity to thank all my collegues in Zienkiewicz Centre for Computational Engineering (ZCCE) and in Laboratory of Computational Methods and Numerical Analysis (LaCàN). Last but certainly not the least, I would like to thank my family and friends who have always supported me throughout these years.

Abstract

In practical engineering applications involving extremely complex geometries, meshing typically constitutes a large portion of the overall design and analysis time. In the computational mechanics community, the ability to perform calculations on tetrahedral meshes has become increasingly important. For these reasons, automated tetrahedral mesh generation by means of Delaunay and advancing front techniques has recently received increasing attention in a number of applications, namely: crash simulations, cardiovascular modelling, blast and fracture modelling.

Unfortunately, modern industry codes in solid mechanics (e.g. LS-DYNA, ANSYS AUTODYN, ABAQUS/Explicit, Altair Hypercrash) typically rely on the use of traditional displacement based Finite Element formulations which possess several distinct disadvantages, namely: (1) reduced order of convergence for strains and stresses in comparison to displacements; (2) high frequency noise in the vicinity of shocks; and (3) numerical instabilities associated with shear locking, volumetric locking and pressure checker-boarding.

In order to address the above mentioned shortcomings, a new mixed-based set of equations for solid dynamics formulated in a system of first order hyperbolic conservation laws was introduced [1–19]. Importantly, the new set of conservation laws has a similar structure to that of the well known Euler equations in the context of Computational Fluid Dynamics (CFD). This enables us to borrow some of the available CFD technologies and to adapt the method in the context of solid dynamics.

This thesis builds on the work carried out in [6] by further developing the upwind cell centred finite volume framework for the numerical analysis of large strain explicit solid dynamics and its tailor-made implementation within the open source code OpenFOAM, extensively used in industrial and academic environments. The object oriented nature of OpenFOAM implementation provides a very efficient platform for future development. In this computational framework, the primary unknown variables are linear momentum and deformation gradient tensor of the system. Moreover, the formulation is further extended for an additional set of geometric strain measures comprising of the co-factor of deformation gradient tensor and the Jacobian of deformation, in order to simulate polyconvex constitutive models ensuring material stability.

The domain is spatially discretised using a standard Godunov-type cell centred framework where second order accuracy is achieved by employing a linear reconstruction procedure in conjunction with a slope limiter. This leads to discontinuities in variables at the cell interface which motivate the use of a Riemann solver by introducing an upwind bias into the evaluation of numerical contact fluxes. The acoustic Riemann solver presented is further developed by applying preconditioned dissipation to improve its performance in the near incompressibility regime and extending its range to contact applications. Moreover, two evolutionary frameworks are proposed in this study to satisfy the underlying involutions (or compatibility conditions) of the system. Additionally, the spatial discretisation is also represented through a node-based cell centred finite volume framework [20] for comparison purposes.

From a temporal discretisation point of view, a two stage Total Variation Diminishing Runge-Kutta time integrator is employed to ensure second order accuracy. Additionally, inclusion of a posteriori global angular momentum projection procedure enables preservation of angular momentum of the system.

Finally, benchmark numerical examples are simulated to demonstrate mesh convergence, momentum preservation and the locking-free nature of the formulation. Moreover, the robustness and accuracy of the computational framework has been thoroughly examined through a series of challenging numerical examples involving contact scenarios and complex computational domains.

Keywords: Cell centred scheme, Finite Volume Method, Hyperbolic conservation laws, Riemann solver, OpenFOAM

Research Output

Journal publications

- □ **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 of Numerical Methods in Engineering*, Vol. 109(3), 407–456, 2017. [DOI: 10.1002/nme.5293]
- □ **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 solid dynamic applications”, *Computer Methods in Applied Mechanics and Engineering*, (Under review).

Conference publications

- □ **J. Haider**, C. H. Lee, A. J. Gil, J. Bonet and A. Huerta. “An upwind cell centred Total Lagrangian finite volume algorithm for nearly incompressible explicit solid dynamic applications”. To appear in the proceedings of the *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, Glasgow, United Kingdom, 11-15 June, 2018.
- □ **J. Haider**, C. H. Lee, A. J. Gil, J. Bonet and A. Huerta. “Large strain solid dynamics in OpenFOAM”. In proceedings of the 4*th* *OpenFOAM User Conference*, Cologne, Germany, 11-13 October, 2016. ***Awarded best student paper at the conference by ESI group***.
- □ **J. Haider**, C. H. Lee, A. J. Gil, J. Bonet and A. Huerta. “A first order hyperbolic framework for large strain computational solid dynamics: An upwind Finite Volume Method”. In proceedings of the *12th World Congress on Computational Mechanics (WCCM)*, Seoul, Korea, 24-29 July, 2016. ***Awarded by the International Association for Computational Mechanics (IACM) to present in the conference***.
- □ C. H. Lee, A. J. Gil, **J. Haider**, O. I. Ismail and J. Bonet. “A first order hyperbolic framework for large strain computational solid dynamics: An upwind Finite Volume Method”. In proceedings of the *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, Crete, Greece, 5-10 June, 2016.

□ **J. Haider**, A. J. Gil, C. H. Lee and J. Bonet. “A first order conservation law formulation for Lagrangian fast solid dynamics in OpenFOAM”. In Proceedings of the *13th US National Congress on Computational Mechanics (USNCCM)*, San Diego, United States, 27-30 July, 2015.

□ **J. Haider**, A. J. Gil, J. Bonet, C. H. Lee and A. Huerta. “A first order conservation law formulation for fast solid dynamics in OpenFOAM”. In Proceedings of the *23rd UK National Conference of the Association for Computational Mechanics in Engineering (ACME)*, Swansea, United Kingdom, 8-10 April, 2015.

Invited talks

□ **J. Haider**, C. H. Lee, A. J. Gil, J. Bonet and A. Huerta.“Large strain solid dynamics in OpenFOAM” at 4th *OpenFOAM User Conference* conducted by ESI group, Cologne, Germany, 11-13 October, 2016.

□ **J. Haider**. “OpenFOAM workshop for beginners: Hands-on training” at *Laboratori de Càlcul Numèric*, UPC BarcelonaTech, Spain, 31 May & 3 June, 2016.

□ **J. Haider**, C. H. Lee, A. J. Gil, J. Bonet and A. Huerta. “A first order hyperbolic framework for large strain computational solid dynamics” at *Laboratori de Càlcul Numèric*, UPC BarcelonaTech, Spain, 20 May, 2016.

Posters

□ **J. Haider**, C. H. Lee, A. J. Gil, J. Bonet and A. Huerta.“Large strain solid dynamics in OpenFOAM” at *The Annual Zienkiewicz Centre for Computational Engineering (ZCCE) Postgraduate Workshop*, Swansea University, United Kingdom, 1-2 February, 2017. ***Awarded best poster at the workshop***.

□ **J. Haider**, A. J. Gil, C. H. Lee, J. Bonet and A. Huerta.“Development of a monolithic conservation law formulation for Fluid-Structure Interaction” at *The Annual Zienkiewicz Centre for Computational Engineering (ZCCE) Postgraduate Workshop*, Swansea University, United Kingdom, 27-28 January, 2015.

“The task is, not so much to see what no one has yet seen; but to think what nobody has yet thought, about that which everybody sees”.

Erwin Schrödinger

CONTENTS

LIST OF FIGURES

Title **Page No.**

LIST OF TABLES

Title **Page No.**

LIST OF ALGORITHMS

Title **Page No.**

LISTINGS

ABBREVIATIONS

ALE	**A**rbitrary **L**agrangian **E**ulerian
AMPA	**A**ngular **M**omentum **P**reserving **A**lgorithm
CAD	**C**omputer **A**ided **D**esign
CCFVM	**C**ell **C**entred **FVM**
CFD	**C**omputational **F**luid **D**ynamics
CFL	**C**ourant-**F**riedrichs-**L**ewy
CSM	**C**omputational **S**olid **M**echanics
C-TOUCH	**C**onstrained-**TOUCH**
FDM	**F**inite **D**ifference **M**ethod
FEM	**F**inite **E**lement **M**ethod
FVM	**F**inite **V**olume **M**ethod
GCL	**G**eometric **C**onservation **L**aw
GLACE	**G**odunov-type **LA**grangian scheme **C**onservative for total **E**nergy
GUI	**G**raphical **U**ser **I**nterface
JST	**J**ameson-**S**chmidt-**T**urkel
LBB	**L**adyzhenskaya-**B**abuška-**B**rezzi
MHD	**M**agneto**h**ydro**d**ynamics
P-TOUCH	**P**enalised-**TOUCH**
OpenFOAM	**Open** Source **F**ield **O**peration **A**nd **M**anipulation
PG-FEM	**P**etrov **G**alerkin **FEM**
RK	**R**unge **K**utta
RS	**R**iemann **S**olver

SPH	**S**moothed **P**article **H**ydrodynamics
TG-FEM	**T**aylor **G**alerkin **FEM**
TL	**T**otal **L**agrangian
TOUCH	**TO**tal Lagrangian **U**pwind **C**ell centred Finite Volume Method for **H**yperbolic conservation laws
TVD	**T**otal **V**ariation **D**iminishing
UL	**U**pdated **L**agrangian
VOF	**V**olume **O**f **F**luid
VMS	**V**ariational **M**ulti-**S**cale
VCFVM	**V**ertex **C**entred **FVM**
X-GLACE	E**X**tended **GLACE**

NOMENCLATURE

Scalars

c	Wave speed	$[m\, s^{-1}]$
c_p	Longitudinal or volumetric wave speed	$[m\, s^{-1}]$
c_s	Transverse or shear wave speed	$[m\, s^{-1}]$
dv	Differential spatial volume	$[m^3]$
dV	Differential material volume	$[m^3]$
E_T	Total energy	$[kg\, m^{-1}\, s^{-2}]$
E	Young's modulus	$[kg\, m^{-1}\, s^{-2}]$
J	Jacobian of deformation	$[-]$
$J_{\boldsymbol{F}}$	Jacobian of deformation based on $\boldsymbol{F}$	$[-]$
N	Finite element shape function	$[-]$
p	Pressure	$[kg\, m^{-1}\, s^{-2}]$
t	Time	$[s]$
x	Distance	$[m]$
α_{CFL}	Courant-Fredrichs-Lewy number	$[-]$
ϵ	Strain	$[-]$
ϵ_p	Plastic strain	$[-]$
κ	Bulk modulus	$[kg\, m^{-1}\, s^{-2}]$
$\tilde{\kappa}$	User defined material constant	$[kg\, m^{-1}\, s^{-2}]$
λ	Lame constant	$[kg\, m^{-1}\, s^{-2}]$
μ	Lame constant	$[kg\, m^{-1}\, s^{-2}]$
ν	Poisson's ratio	$[-]$
$\varOmega$	Volume in material configuration	$[m^3]$
$\varOmega_0$	Volume in spatial configuration	$[m^3]$
ϕ	Slope limiter	$[-]$

ψ	Strain energy	$[]$
ρ	Spatial density	$\left[kg\,m^{-3}\right]$
ρ_0	Material density	$\left[kg\,m^{-3}\right]$
ξ	Penalisation factor	$[-]$

Vectors

$\boldsymbol{a}$	Spatial area vector	$\left[m^2\right]$
$\boldsymbol{A}$	Material area vector	$\left[m^2\right]$
$\boldsymbol{d}$	Distance	$[m]$
$\boldsymbol{\mathcal{F}}$	Flux vector	
$\boldsymbol{\mathcal{F}}^C$	Contact flux	
$\boldsymbol{\mathcal{L}}$	Left eigenvector of $\boldsymbol{\mathcal{A}}$	
$\boldsymbol{n}$	Spatial outward unit normal	$[m]$
$\boldsymbol{N}$	Material outward unit normal	$[m]$
$\boldsymbol{p}$	Linear momentum	$\left[kg\,m\,s^{-1}\right]$
$\boldsymbol{Q}$	Heat flux vector	$[]$
$\boldsymbol{\mathcal{R}}$	Right eigenvector of $\boldsymbol{\mathcal{A}}$	
$\boldsymbol{\mathcal{S}}$	Source vector	
$\boldsymbol{t}$	Material or First Piola Kirchoff vector	$\left[kg\,m^{-1}s^{-2}\right]$
$\boldsymbol{u}$	Displacement field, $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$	$[m]$
$\boldsymbol{\mathcal{U}}$	Vector of conserved variables	
$\boldsymbol{v}$	Spatial velocity	$\left[m\,s^{-1}\right]$
$\boldsymbol{V}$	Material velocity	$\left[m\,s^{-1}\right]$
$\boldsymbol{x}$	Spatial or Eulerian coordinates	$[m]$
$\boldsymbol{X}$	Material or Lagrangian coordinates	$[m]$
$\boldsymbol{\sigma}$	Cauchy or spatial traction	$\left[kg\,m\;s^{-2}\right]$
$\boldsymbol{\lambda}$	Lagrange multiplier	$[-]$

Tensors

$\boldsymbol{\mathcal{A}}$	Flux Jacobian matrix	
$\boldsymbol{C}$	Right Cauchy-Green strain	$[-]$
$\boldsymbol{F}$	Deformation gradient	$[-]$
$\boldsymbol{F}_{\boldsymbol{x}}$	Deformation gradient based on $\boldsymbol{x}$	$[-]$
$\boldsymbol{G}$	Gradient of a scalar	
$\boldsymbol{H}$	Cofactor of deformation	$[-]$
$\boldsymbol{H}_{\boldsymbol{x}}$	Cofactor of deformation based on $\boldsymbol{x}$	$[-]$
$\boldsymbol{I}$	Second order identity tensor	$[-]$
$\boldsymbol{P}$	First Piola Kirchoff stress	$\left[kg\, m^{-1}\, s^{-2}\right]$
$\boldsymbol{\delta}$	Kronecker delta	$[-]$
$\boldsymbol{\epsilon}$	Alternating or permutation tensor	$[-]$

Subscripts [1]

$\mathcal{U}_0$	Property in material configuration
$\mathcal{U}_a$	Nodal value
$\mathcal{U}_\alpha$	Neighbour value
$\mathcal{U}_e$	Cell center value
$\mathcal{U}_{ea}$	Property at the node associated to cell e
$\mathcal{U}_f$	Face center value
$\mathcal{U}_{ef}$	Property at the face associated to cell e
$\mathcal{U}_g$	Property at Gauss point
$\boldsymbol{A}_{\text{dev}}$	Deviatoric component
$\boldsymbol{A}_{\text{vol}}$	Volumetric component

Superscripts [1]

$\mathcal{U}^+$	State left of the interface

[1] $\mathcal{U}$ represents either a scalar or vector quantity whereas $\boldsymbol{A}$ represents a tensorial quantity.

$\mathcal{U}^-$	State right of the interface
$\mathcal{U}^*$	First Runge-Kutta stage
$\mathcal{U}^{**}$	Second Runge-Kutta stage
$\mathcal{U}^C$	Contact Riemann values
$\boldsymbol{A}^T$	Transpose of a tensor/matrix
$\boldsymbol{A}^{-1}$	Inverse of a tensor/matrix
$\hat{\boldsymbol{A}}$	Unit vector
$\dot{\mathcal{U}}$	Time derivative

Operators

$[\![\ ]\!]$	Jump operator
$\det[\ ]$	Determinant
$\nabla \cdot [\ ], \quad \mathrm{div}\,[\ ]$	Spatial divergence
$\nabla_0 \cdot [\ ], \quad \mathrm{DIV}\,[\ ]$	Material divergence
$\nabla\, [\ ], \quad \mathrm{grad}\,[\ ]$	Spatial gradient
$\nabla_0\, [\ ], \quad \mathrm{GRAD}\,[\ ]$	Material gradient
$\nabla \times [\ ], \quad \mathrm{curl}\,[\ ]$	Spatial curl
$\nabla_0 \times [\ ], \quad \mathrm{CURL}\,[\ ]$	Material curl
$[\ ] : [\ ]$	Double dot product
$[\ ] \times [\ ]$	Cross product
$[\ ] \otimes [\ ]$	Dyadic product
$[\ ] \boldsymbol{\times} [\ ]$	Tensor cross product
$\mathrm{tr}\,[\ ]$	Trace of tensor/matrix

Chapter 1

INTRODUCTION

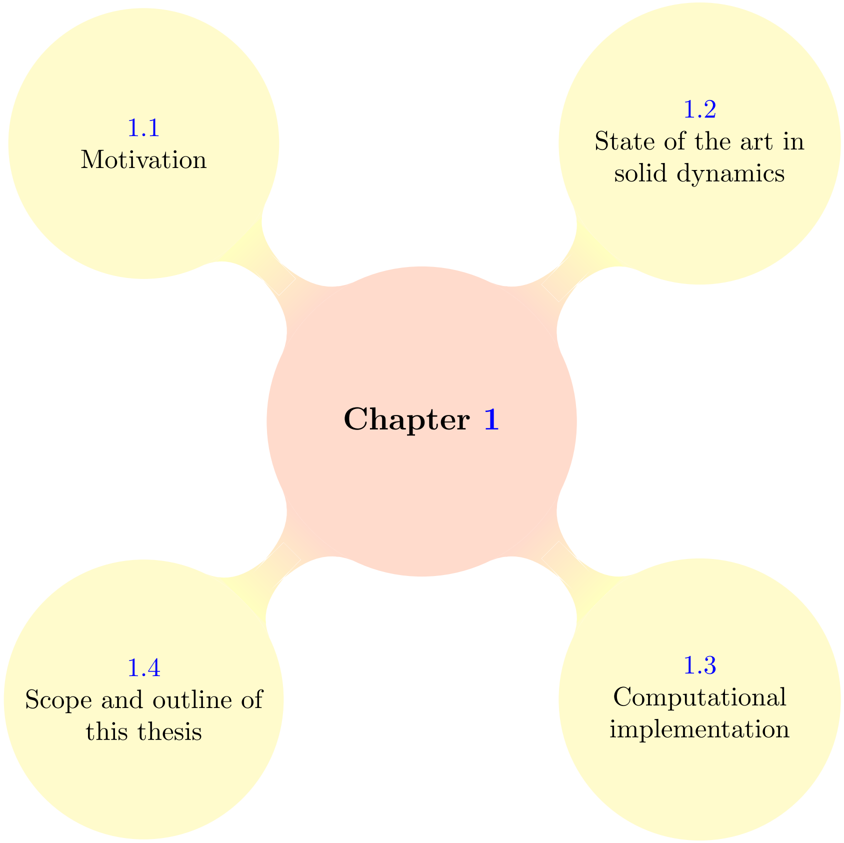

1.1
Motivation
1.2
State of the art in solid dynamics
Chapter 1
1.4
Scope and outline of this thesis
1.3
Computational implementation

FIGURE 1.1: Structure of Chapter 1

1.1 Motivation

Numerous engineering applications in automotive, aerospace, defence and biomedical industries (see Fig. 1.2) involve large material deformation occurring in very small periods of time. Traditionally, the structural analysis relies on experimental procedures through construction of expensive prototypes. Typically, several experiments are performed, often involving destruction of the prototype, to determine an efficient and optimised product design. Moreover, in certain applications construction of a prototype is even not possible, such as soft tissue modelling in biomedical engineering. Therefore, in recent years, reliance on computational mechanics to predict solid behaviour of structures is gaining wide prominence, primarily due to the fact that it is comparatively a very cheap alternative. The disadvantage of this approach is the fact that the computational model must be validated through some experimental data. Once validated, the model can be used to simulate various scenarios and predict solid behaviour to a high degree of reliability. In Computational Solid Mechanics (CSM), the quantities of interest are mechanical/thermal stresses, deformations, vibration of the solid parts including fatigue analysis and life prediction [21]. Today, several commercial CSM software are available for simulation of complex structural problems.

Fig. 1.3 shows a schematic of the detailed procedure for undertaking a computational approach to provide a solution to an engineering problem [21, 22]. For a given engineering problem, the first task at hand is to define an appropriate mathematical model able to correctly capture the physics of the problem. This step usually borrows help from experimental procedures for material characterisation. Defining an adequate mathematical model is one of the most crucial parts of this computational workflow, since an incorrect model will lead to invaluable results. A well defined mathematical model leads to a continuous problem usually defined in terms of partial differential equations. The next step in this workflow is the discretisation process. This involves discretisation of the spatial domain into a finite number of subdomains, a process known as grid generation, where the unknown quantities are computed [2]. Once a computational grid is available, the mathematical model described as a continuous system is replaced with a discrete representation. Traditionally, the discretisation approaches widely utilised are the Finite Element Method (FEM), Finite Volume Method (FVM), Finite Difference Method (FDM) and meshless methods. Now it is possible to define appropriate numerical schemes for spatial and temporal discretisation. The next phase involves obtaining a solution for the algebraic system of equations to get numerical results. These numerical results cannot be trusted blindly and must be validated through published literature either through available experimental data or solution from well-known commercial software packages. Moreover, it should be ensured that the numerical quantities of interest are mesh independent by carrying out a grid independence study, a vital ingredient of numerical analysis. The computational results should also be checked in terms of stability, accuracy and convergence of the algorithm.

[2] The subdomains are usually referred to as ‘elements’ in structural mechanics and ‘cells’ or ‘control volumes’ in CFD jargon.

(a) Automotive

(b) Aerospace

(c) Defence

(d) Biomedical

FIGURE 1.2: Applications of computational solid dynamics in (a) automotive; (b) aerospace; (c) defence; and (d) biomedical industries.

Once all of this is taken into account, a numerical solution to our engineering problem is obtained.

Current commercial codes (e.g. PAM-CRASH, ANSYS AUTODYN, LS-DYNA, ABAQUS, Altair HyperCrash) used in industry for the simulation of large-scale solid mechanics problems (e.g. crash, contact, impact, fracture) are typically based on the use of classical low order Finite Element displacement based formulations. Linear tetrahedral elements tend to be preferred due to the availability of robust unstructured mesh generators [23]. However, it is well-known that these formulations present a number of shortcomings [24, 25], namely (1) reduced order of convergence for strains and stresses in comparison with displacements [26, 27]; (2) high frequency noise in the vicinity of shocks or sharp spatial gradients [28–31] and (3) numerical instabilities associated with shear locking [32], volumetric locking [33, 34] and pressure checker-boarding [35]. These drawbacks are more pronounced when simulating materials in the near incompressibility regime such as rubber or rubber-like materials. Incompressible or nearly incompressible materials are characterised by a large ratio of bulk modulus to shear modulus ($\kappa/\mu \geq 100$) [36–38]. Problems arise in the presence of incompressibility which lead to the so-called volumetric locking phenomenon [39–41]. This leads to an overestimation of the stiffness related to the volumetric part, which results in overly stiff behaviour [24, 32, 42].

To alleviate some of the shortcomings mentioned above, a variety of methodologies have been developed and implemented over time into commercial codes. These enhancements are appealing to Industry as, without much extra cost (in terms of number of degrees of freedom), the necessary modifications into a standard displacement formulation tend to be very minor. This is in clear contrast to the case of high order interpolation schemes [43–45], an equally valid alternative. However, the increase in number of integration points in high order schemes, leads to a reduction in computational efficiency as opposed to its low order alternative.

This thesis offers an alternative low order computational framework by solving a new set of conservation laws that overcome the drawbacks posed by the conventional displacement based formulations. The representation of layout of this chapter is shown in Fig. 1.1.

1.2 State of the art in solid dynamics

1.2.1 Traditional displacement-based formulations

Traditional displacement-based formulations have long been used within the context of Finite Element Method. These formulations show a tendency to lock for nearly incompressible materials, specially in bending dominated scenarios, if low order tetrahedral elements are utilised [32]. Unfortunately, many real-life engineering applications of interest involve complex domains which can often only be meshed using tetrahedras [46]. Moreover, in the context of tetrahedral elements, robust and efficient tetrahedral mesh generators are widely available, primarily relying on Delaunay tetrahedralisation and advancing front techniques [47, 48] [3]. One of the methods developed to avoid locking is the reduced integration technique [24, 49, 50] where reduced Gauss integration points are utilised to under-integrate the overall stiffness component [4]. This clearly shows that a compromise is achieved by sacrificing accuracy of results against computational time. An alternative methodology, known as the B-Bar method or Mean Dilatation Technique [52, 53] was proposed to prevent locking by decomposing the stiffness contribution into its volumetric and deviatoric components. The deviatoric part of the stiffness is properly integrated whereas the volumetric component of the stress is under-integrated (equivalent to selective reduced integration). This method successfully overcomes volumetric locking, however it still exihibits shear locking in bending dominated situations. Both of these approaches are widely employed in commercial solid dynamic codes, despite not circumventing the inf-sup Ladyzhenskaya-Babuška-Brezzi (LBB) condition [54] [5].

In the case of linear tetrahedral elements, a very popular choice is that of nodally integrated elements, introduced by Bonet *et al.* [40], where the pressure field is under-integrated at nodes.

[3] Development of robust unstructured hexahedral mesh generators is still an open field for research.

[4] Note that reduced integration technique is not applicable to linear tetrahedral elements, since the integration order cannot be reduced further due to the presence of a single fully integrated point [51].

[5] Ladyzhenskaya (1969), Babuška (1971) and Brezzi (1974) put forward the LBB (or inf-sup) compatibility condition, that provides the basic mathematical criterion to guarantee a stable and convergent mixed finite element methodology [32, 54].

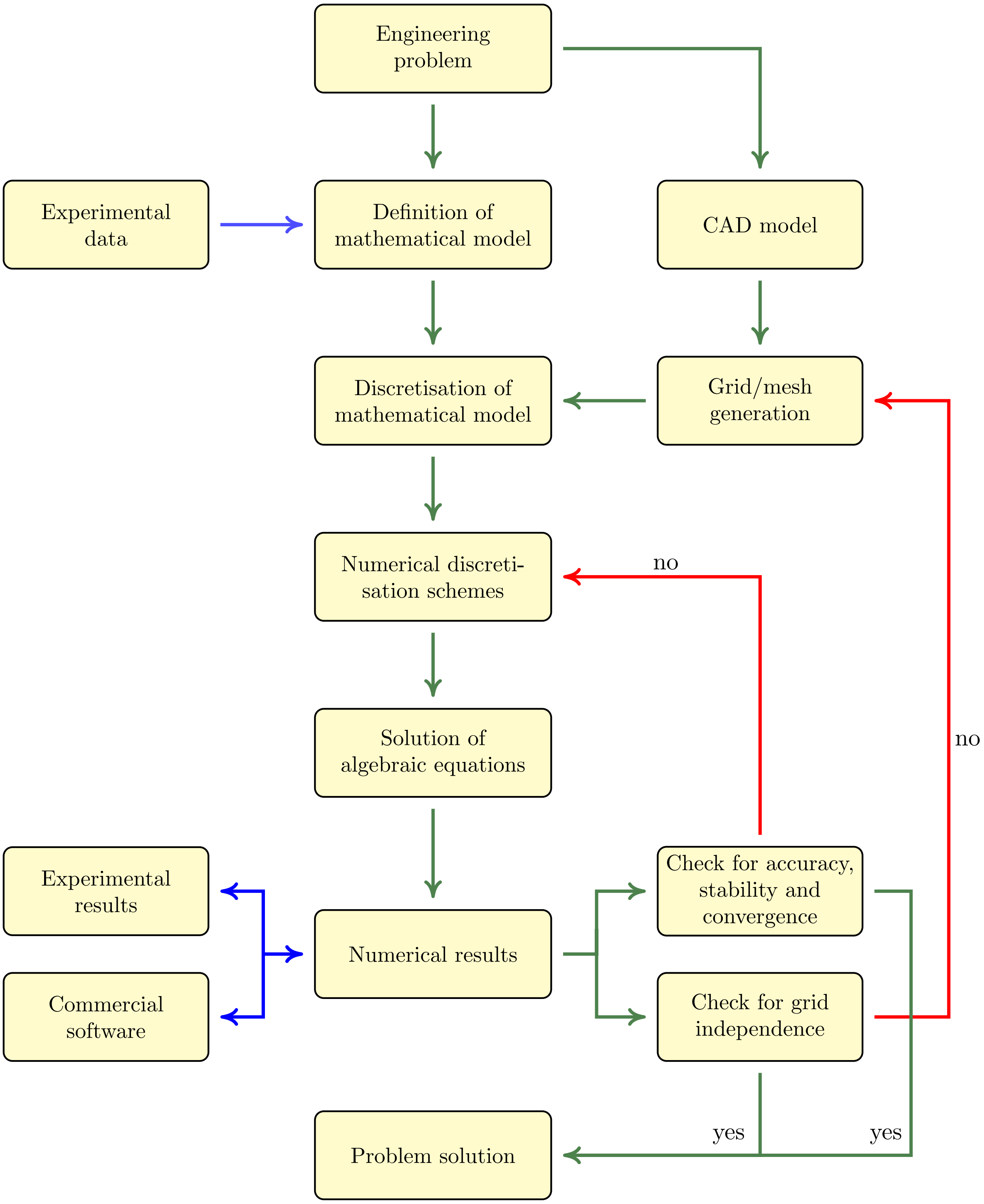

Engineering problem
Experimental data
Definition of mathematical model
CAD model
Discretisation of mathematical model
Grid/mesh generation
Numerical discretisation schemes
no
no
Solution of algebraic equations
Experimental results
Numerical results
Check for accuracy, stability and convergence
Commercial software
Check for grid independence
yes
yes
Problem solution

FIGURE 1.3: Flowchart for numerical simulation of engineering problems.

Although this methodology was found to perform extremely well in nearly incompressible impact problems, it behaved poorly in bending dominated scenarios. Extensive efforts have since been made in order to prevent the appearance of hourglassing-like modes [39, 55, 56], a shortcoming of this finite element. Several variants of the original nodal-pressure approach have since followed, including the averaged nodal deformation gradient [41], the F-bar method [57] and the Smoothed Finite Element Method [33]. Despite exhibiting locking-free deformation behaviour, all methods described above still suffer from spurious hydrostatic pressure fluctuations when simulating large strain nearly incompressible materials [34].

On the other hand, Finite Volume based Methods (FVM), despite their maturity in the field of Computational Fluid Dynamics, have not attracted that much attention in the Solid Mechanics community. Some attempts have been reported at trying to solve solid mechanics problems using these methods [58–62], with most of them restricted to the use of displacement based formulations for linear elastic small strain problems, with very limited effort directed towards dealing with nearly incompressible materials [63]. Recently, some work has been published using the Finite Volume Method in contact mechanics applications [64], moderate strains [65] and metal forming application [66] by using the open source platform OpenFOAM.

In the context of meshless methods, a Lagrangian Smooth Particle Hydrodynamics (SPH) formalism has been presented for solid mechanics [67–71]. One of the most attractive features of SPH is its mesh free nature. The absence of mesh and the calculation of interactions among particles based exclusively on their separation, allow ease of computation for large deformation problems[6]. Due to its low computational cost, reasonable accuracy and stability, as well as its ability to handle extremely large distortions [69], the SPH methodology has been shown to be very competitive [33]. It is well established that the standard displacement-based SPH methodology typically suffers from numerical errors near boundaries due to lack of consistency [72, 73], presence of tensile instability which can result in the non-physical clumping of particles [74] and zero-energy modes due to the rank-deficiency inherent to the use of Galerkin particle integration [75].

1.2.2 Mixed methodologies

One of the earliest attempts at employing a mixed system of first order conservation laws in large strain solid dynamics can be traced back to the work of Trangenstein and Colella [1–4], where the primary unknown variables were linear momentum $\boldsymbol{p}$ and deformation gradient tensor $\boldsymbol{F}$. In particular, a second order Total Lagrangian finite volume framework together with the use of an upwinding stabilisation was presented in two-dimensional linear elasticity [4]. Although the consideration of involutions is outlined as part of the study, its numerical implementation is not fully described. Moreover, the example presented is restricted to linear elasticity within the small strain regime.

[6] For mesh dependent frameworks, large deformation could lead to extremely distorted elements thereby leading to failure of the numerical scheme.

An interesting FVM approach was proposed by Després and Mazeran [20] for gas dynamics applications. This approach, further explored in [76], is based on the discretisation of conservation laws through the cell centred Lagrangian scheme termed GLACE. The GLACE scheme employs a node based solver with numerical interface tractions and velocities evaluated at the mesh nodes. This nodal scheme does not require ad-hoc procedures [25] to obtain the nodal velocities and thus it is consistent with the so-called Geometric Conservation Law. Subsequently, Maire and collaborators [77–79] proposed an alternative EUCCLHYD nodal scheme to overcome the numerical instabilities associated with the GLACE scheme. The extension of nodal scheme to (two dimensional) hyperelastic solids was first reported by Kluth and Després in [5], employing a $\{\boldsymbol{p}, \boldsymbol{F}\}$ Updated Lagrangian first order cell centred FVM. Later, EUCCLHYD scheme [7, 9] and a Cell-Centred Hydrodynamics (CCH) scheme [80] were also presented for hyper-elastic solids in the nodal framework.

Moreover, the research group at Massachusetts Institute of Technology discretised the $\{\boldsymbol{p}, \boldsymbol{F}\}$ system using a Hybridizable Discontinuous Galerkin (HDG) method [12]. On the other hand, Scovazzi and co-authors [46, 81] have proposed an effective alternative tetrahedral velocity/-pressure Updated Lagrangian mixed methodology. They introduced stabilising mechanisms through the use of the Variational Multi-Scale (VMS) method, widely used in the context of CFD [82–89].

Over the past few years, the research group at Swansea University have pursued the same $\{\boldsymbol{p}, \boldsymbol{F}\}$ based mixed methodology, in the form of a system of first order hyperbolic conservation laws. In this case, velocities, deviatoric stresses and volumetric stresses display the same rate of convergence, which proves ideal in the case of low order discretisations. This approach has been studied using a wide variety of second order spatial discretisation techniques, well known in the CFD community. Specifically, two dimensional cell centred upwind FVM (2D upwind-CCFVM) [6], two-step Taylor-Galerkin FEM (2D TG-FEM) [13], vertex centred Jameson-Schmidt-Turkel FVM (3D JST-VCFVM) [10], stabilised Petrov-Galerkin FEM (3D PG-FEM) [15], vertex centred upwind FVM (3D upwind-VCFVM) [11], three dimensional cell centred upwind FVM (3D upwind-CCFVM) [8], Jameson-Schmidt-Turkel SPH (3D JST-SPH) [18] and Streamline Upwind Smooth PetrovGalerkin SPH (3D SUPG-SPH) [19].

Subsequently, the two-field $\{\boldsymbol{p}, \boldsymbol{F}\}$ formulation was then augmented by considering a new conservation law for the Jacobian J of the deformation to effectively solve nearly incompressible and truly incompressible materials [11, 14]. Further enhancement of this $\{\boldsymbol{p}, \boldsymbol{F}, J\}$ framework has recently been reported in [16, 17], when considering compressible, nearly incompressible and truly incompressible materials governed by a polyconvex constitutive law, where the co-factor $\boldsymbol{H}$ of the deformation plays a dominant role, leading to a $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ system of conservation laws. The extended set of unknowns $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ yields an elegant system of conservation laws, where the existence of a generalised convex entropy function enables the derivation of an alternative system of conservation laws in terms of entropy conjugates [16, 17]. These CFD-based methodologies have proven to be very efficient circumventing the drawbacks of the traditional displacement based techniques. A brief comparison of the proposed mixed formulation and the standard low order displacement-based formulations is provided in Table 1.1.

		Displacement-based formulations	**Proposed mixed formulation**
1.	Locking in nearly incompressible scenarios	✓	✓
2.	Bending difficulty	✗	✓
3.	Robust shock capturing capability	✗	✓
4.	Equal order of convergence for stresses/strains	✗	✓
5.	No pressure instabilities	✗	✓

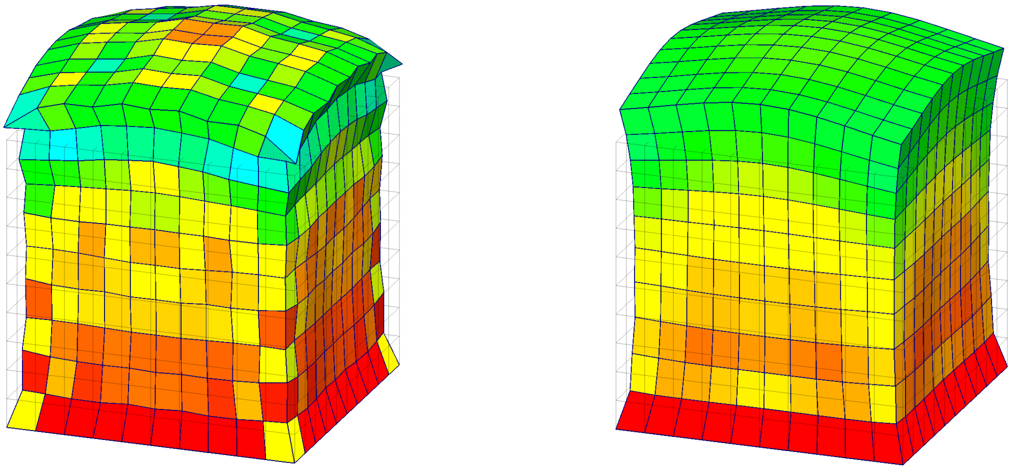
Table 1.1: A comparison of conventional low order displacement-based and proposed mixed formulations

1.3 Computational implementation

Computational mechanics provides us with the fundamental tools for numerical simulation of numerous physical processes including crash simulations, contact/impact mechanics and fracture/damage mechanics. With the availability of highly optimised commercial and open source software packages, it is now possible to revolutionise how computational mechanics impacts design and optimisation areas including the simulation of large-scale real-life complex problems. Although commercially available codes (e.g. PAM-CRASH, ANSYS AUTODYN, LS-DYNA, ABAQUS, Altair HyperCrash) present various advantages in terms of usability of existing technology, they pose a serious serious drawback when it comes to development and extension of current capabilities of software due to the closed nature of implementation. To overcome this, preferential use of open source codes has gained popularity in both industrial and academic environments.

One such open source code used to solve continuum mechanics problems is OpenFOAM which is based on the cell centred Finite Volume Method[7]. OpenFOAM was primarily developed for solving CFD problems and includes various fluid solvers with multi-physics capability. These solvers are generally very robust and several studies have been carried out to prove that OpenFOAM results are comparable with other commercial software. Apart from fluids, OpenFOAM also includes solvers for electromagnetics, molecular dynamics and solid mechanics applications. However, the solid solvers possess a major drawback since they are only capable of simulating linear elasticity within small strain deformation regime. A comparison of the deformation and pressure distribution obtained using the existing solid solver and the implemented proposed

[7] OpenFOAM is licensed under the open source General Public License (GPL) which gives user the flexibility to freely download, install, use and modify this high-end code.



(a) Standard formulation

(b) Proposed formulation

Figure 1.4: Comparison of (a) standard displacement based; and (b) proposed mixed formulations in OpenFOAM.

		OpenFOAM	**Commercial software**
1.	Cost effectiveness	✓	✗
2.	Parallel computing	✓	✓
3.	Source code	✓	✗
4.	Redistribution of code	✓	✗
5.	Collaborative development	✓	✗
6.	Documentation	✗	✓
7.	GUI / user friendliness	✗	✓

Table 1.2: Comparison of OpenFOAM against commercial software.

formulation is shown in Fig. 1.4. Therefore, these solid solvers have plenty of room for development before they can be used to simulate real-life problems. Keeping this in mind, it was decided to implement the proposed mixed formulation from scratch in OpenFOAM code. A comprehensive comparison of OpenFOAM against alternative commercial software is shown in Table 1.2 highlighting the various advantages offered by the software.

1.4 Scope and outline of this thesis

The objective of this thesis is to present a new computational framework for the numerical analysis of large strain explicit solid dynamic problems. The algorithm is entitled **TOUCH**, which is an abbreviation for **TO**tal Lagrangian **U**pwind **C**ell Centred Finite Volume Method for **H**yperbolic conservation laws. The proposed algorithm is formulated based on a new set of first order conservation laws in solid dynamics, where the unknown variables are linear momentum and the extended set of geometric strain measures. The new equations must then be supplemented by appropriate constitutive laws to close the system of equations. This work focuses only on isothermal hyper-elastic and elasto-plastic constitutive models. Moreover, the new system is spatially discretised using the second order cell centred Finite Volume Method. Due to the nature of the scheme, discontiuity in solution at the cell interface motivates the use of an upwind Riemann solver to evaluate numerical fluxes. Furthermore, the semi-discrete equations are evolved in time using a family of Total Variation Diminishing (TVD) Runge-Kutta (RK) time integration schemes [90]. The scope of this thesis is summarised in Fig. 1.5.

For the ease of understanding and clarity, the outline of this thesis is shown below

□ **Chapter 2**: *Conservation laws in solid dynamics.*
This chapter introduces the conservation laws used in solid dynamics presented in a Total Lagrangian framework. For closure of the problem, appropriate (polyconvex) constitute models including plasticity are also discussed. Finally, an eigenstructure analysis of the system of conservation laws is also presented to demonstrate hyperbolicity of the problem.

□ **Chapter 3**: *Finite volume spatial discretisation.*
This chapter focuses on the spatial discretisation using the cell centred finite volume methodology in the context of both; (a) Godunov-type CCFVM; and (b) nodal CCFVM. Second order spatial accuracy is ensured by employing a linear reconstruction procedure together with a slope limiter to ensure satisfaction of monotonicity.

□ **Chapter 4**: *Riemann solver.*
For evaluation of numerical fluxes, an acoustic Riemann solver is presented in this chapter. The Riemann solver is discussed in the context of both face-based and node-based scenarios. Moreover, the derivation of various boundary conditions is also presented.

□ **Chapter 5**: *Involutions.*
In this chapter, two Godunov-type finite volume methodologies namely; (a) constrained-TOUCH; and (b) penalised-TOUCH are introduced in order to control the spurious curl mechanisms that accumulate over a long term response and lead to the breakdown of the numerical scheme.

□ **Chapter 6**: *Temporal discretisation.*
Time discretisation of the governing equations by utilising a one-step two-stage Runge-Kutta time integrator is presented in this chapter. Moreover, for completeness, an angular

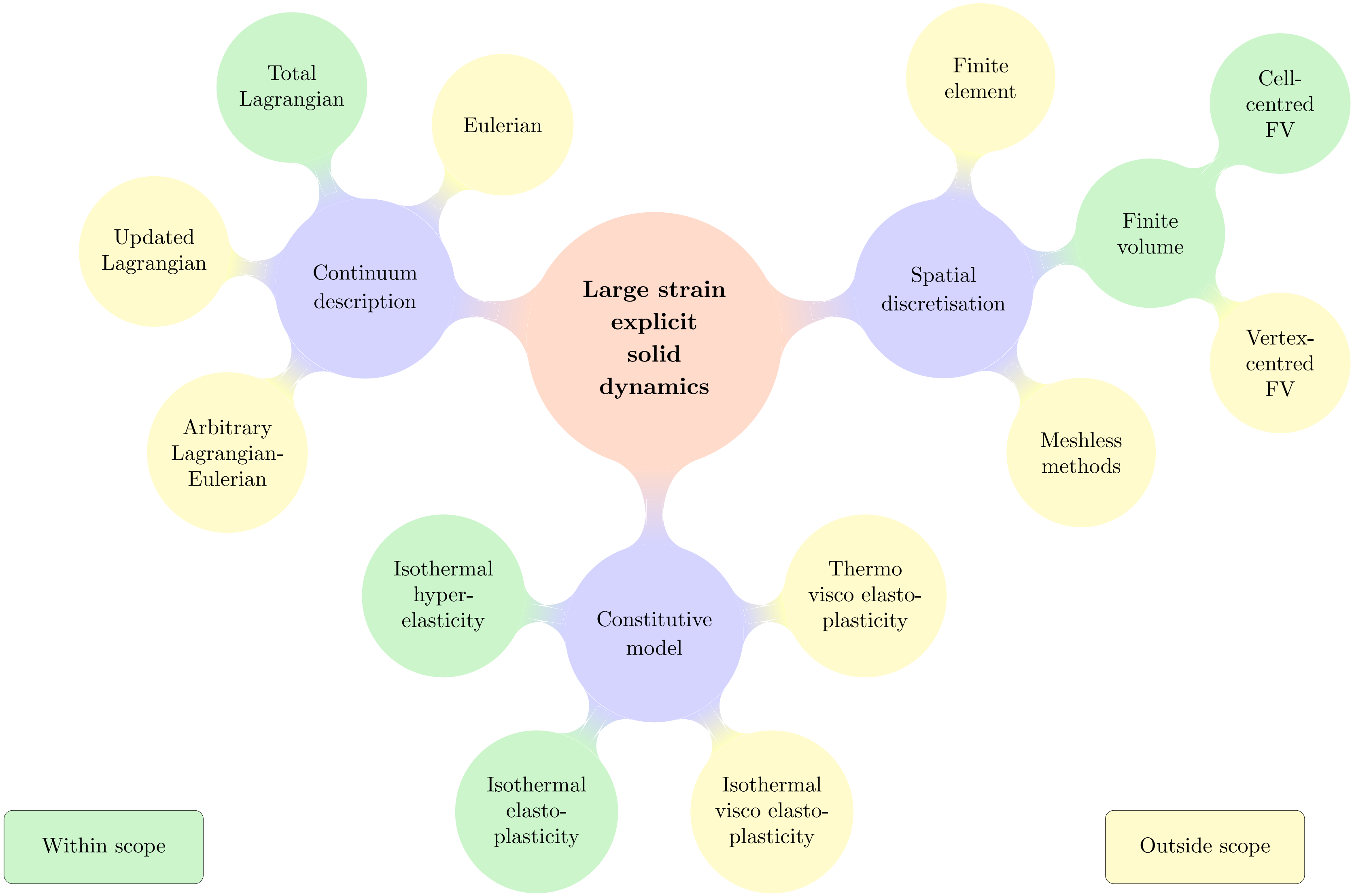


Figure 1.5: Scope of this thesis

momentum preserving algorithm is also discussed. Finally, at the end of this chapter, a complete numerical algorithm is presented to summarise the flow of the proposed formulation.

□ **Chapter 7**: *Implementation in OpenFOAM.*
The proposed mixed formulation has been implemented from scratch in the open source finite volume code OpenFOAM. The numerical implementation of the solid mechanics solver is discussed in detail in this chapter.

□ **Chapter 8**: *Benchmark tests.*
In this chapter, a series of benchmark numerical examples are simulated to demonstrate convergence, grid independence and conservation properties of the proposed mixed formulations. For benchmarking purposes, results are compared against other in-house numerical techniques, including finite element and meshless methods.

□ **Chapter 9**: *Complex problems.*
To further examine the robustness of the algorithm, more complex numerical examples are presented. This involves physical contact problems with complex geometries.

□ **Chapter 10**: *Concluding remarks.*
Finally, a summary of the work done as part of this thesis is presented along with concluding remarks. Future potential directions of research are also discussed.

1.5 General remarks

Some general remarks applicable to this thesis are mentioned below:

□ All quantities mentioned in this work are expressed in *SI units* unless otherwise stated.

□ *Vectorial* and *tensorial* quantities are represented with **bold** characters.

□ Important *equations* are presented within a **red** framed box, whereas useful *remarks* are shown in a **blue** colored box.

□ Linux terminal *commands* are always preceded by “<<” symbol.

□ OpenFOAM *files* and *directories* are mentioned in an *`italic teletype font`*.

□ Implementation of the proposed solid mechanics solver was carried out in *OpenFOAM version 2.3.0.* Since the solver has been implemented from scratch, only minor modifications are required to make it compatible with newer versions of OpenFOAM.

□ Post-processing of results has been done using *ParaView version 4.1.0* and plots have been generated with *MATLAB* software.

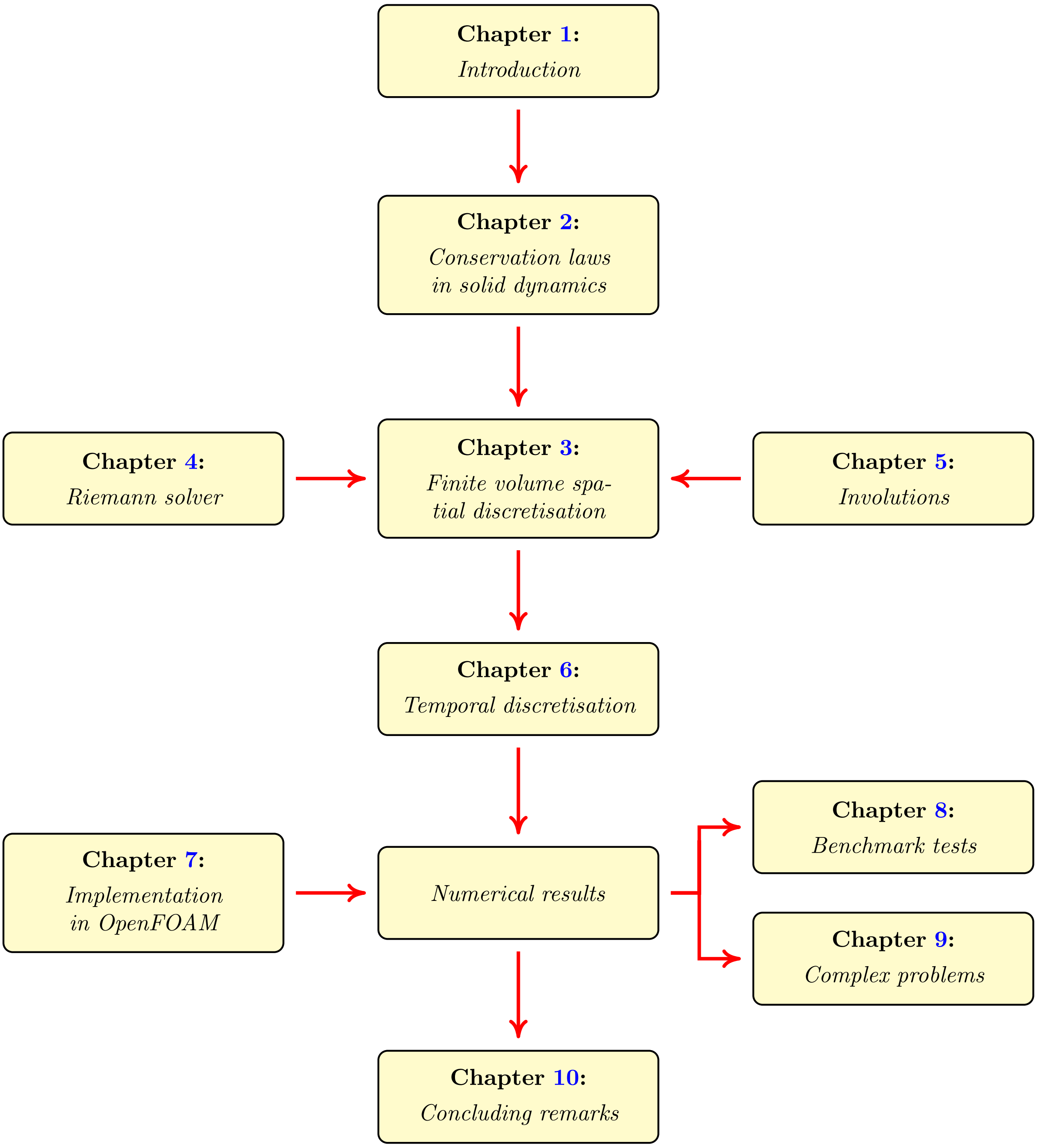


FIGURE 1.6: Outline of this thesis.

Chapter 2

CONSERVATION LAWS IN SOLID DYNAMICS

2.1 Preliminaries

This chapter presents the basics of nonlinear continuum mechanics. Firstly, various methodologies for defining the kinematic description of a continuum body are presented in Section 2.2. This is followed by a kinematic description of motion presented in Section 2.3. The governing equations to be used for the numerical analysis of solid dynamic problems are presented (Section 2.4) and formulated as a system of hyperbolic conservation laws (Section 2.5). In order to close the system of equations, an appropriate constitutive model needs to be defined (Section 2.6). Finally an eigenvalue analysis of the hyperbolic system is carried out in Section 2.7 to prove well-posedness of the problem. The structural layout of this chapter is shown in Fig. 2.1.

2.2 Continuum description

An appropriate kinematic description of continuum body motion is essential to perform numerical analysis. This choice determines the relationship between deforming continuum and the computational mesh. In continuum mechanics, the two classical approaches (see Fig. 2.2) generally employed are: (a) Lagrangian description generally preferred for solids and (b) Eulerian description widely used in the context of fluid dynamics [50, 54].

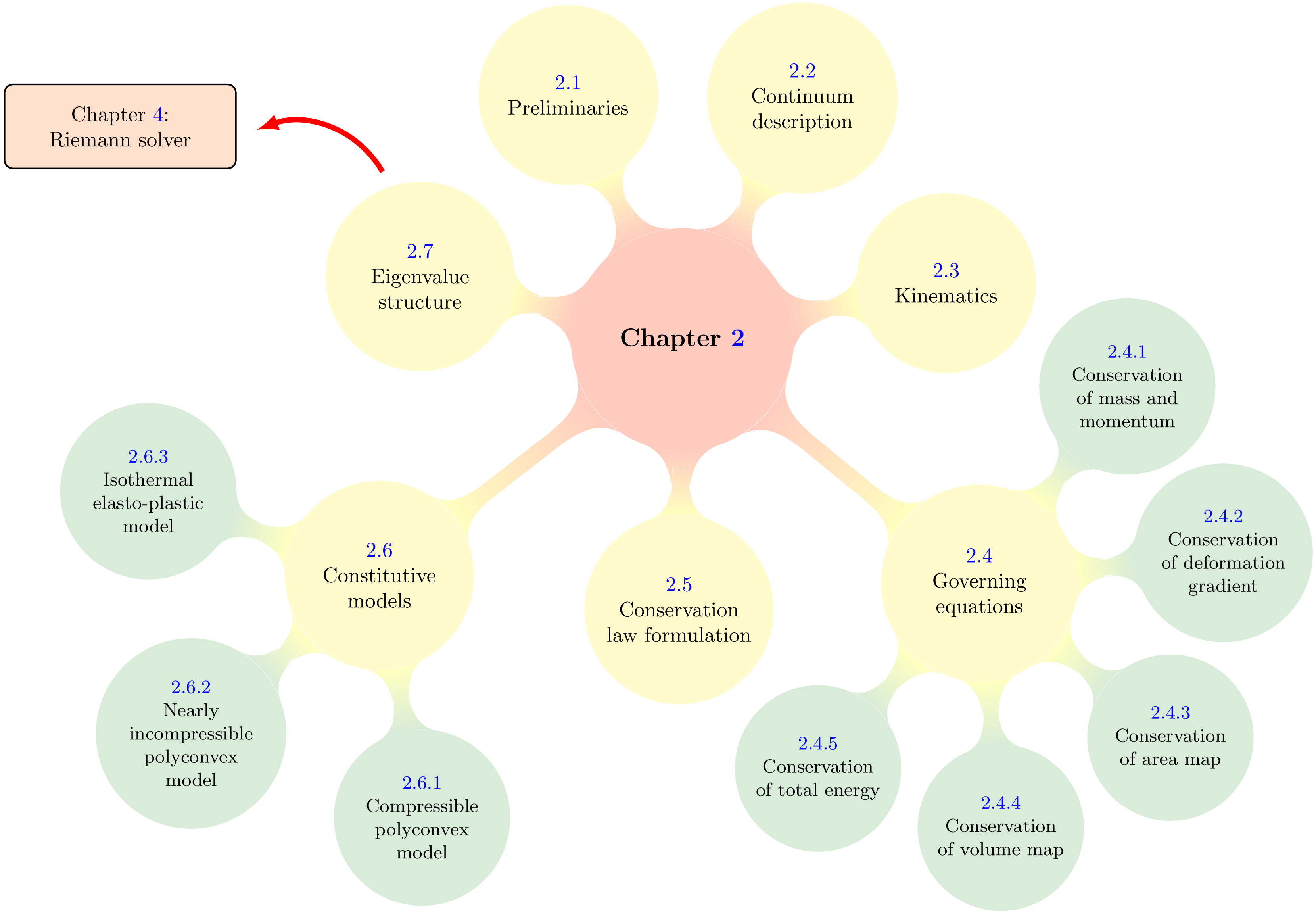

Chapter 4:
Riemann solver
2.1
Preliminaries
2.2
Continuum description
2.7
Eigenvalue structure
Chapter 2
2.3
Kinematics
2.4.1
Conservation of mass and momentum
2.6.3
Isothermal elasto-plastic model
2.6
Constitutive models
2.5
Conservation law formulation
2.4
Governing equations
2.4.2
Conservation of deformation gradient
2.6.2
Nearly incompressible polyconvex model
2.6.1
Compressible polyconvex model
2.4.5
Conservation of total energy
2.4.4
Conservation of volume map
2.4.3
Conservation of area map

FIGURE 2.1: Structure of Chapter 2

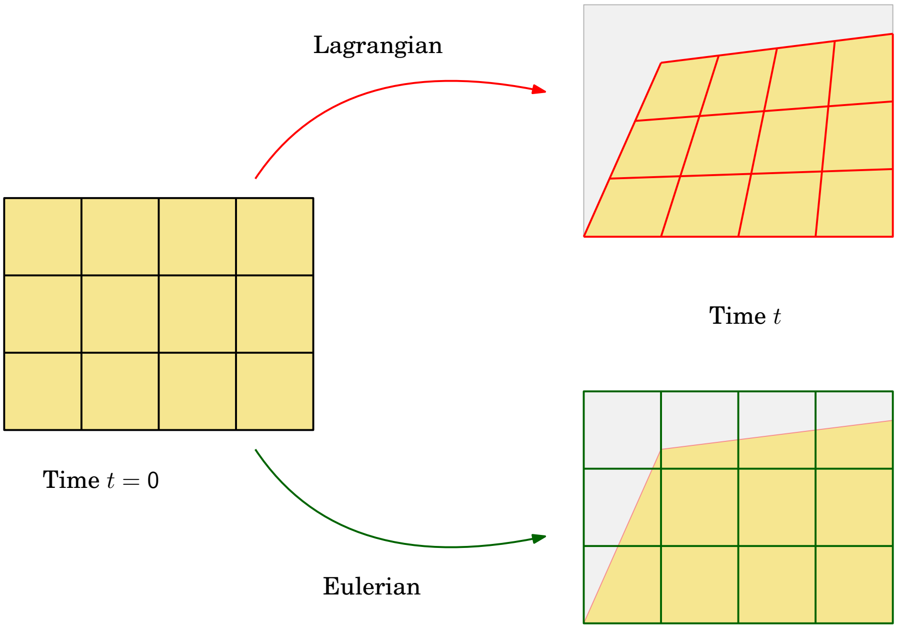


FIGURE 2.2: Lagrangian and Eulerian frameworks

In the case of Lagrangian algorithms, the computational mesh follows the material points during the deformation. As a direct consequence, mass is always conserved within an element of a Lagrangian mesh since no mass transfer occurs across the element interface. Obvious advantage lies in the easiness with which complicated boundary conditions can be applied since the material boundary remains coincident with the mesh boundary throughout the evolution of the problem. Similarly, for multi-material problems, a mesh point on the interface in reference configuration remains at the interface throughout the deformation process. Moreover, Lagrangian algorithms facilitate the treatment of constitutive models that are history dependent. However, Lagrangian algorithms are often prone to mesh entanglement specially for complex problems where the magnitude of deformation is large which leads to inaccurate results [49] [8]. There are two approaches to formulate a problem when dealing with Lagrangian meshes; (a) Total Lagrangian; and (b) Updated Lagrangian. In the Total Lagrangian approach, the derivatives and integrals are taken with respect to the material coordinates $\boldsymbol{X}$ whereas in Updated Lagrangian approach, the derivatives and integrals are taken with respect to the spatial (Eulerian) coordinates $\boldsymbol{x}$.

On the other hand, Eulerian algorithms deal with a fixed computational grid throughout the numerical simulation. This is advantageous when it comes to fluid simulations since it allows severe distortion of the continuum. However, a compromise has to be made since interface cannot be tracked accurately and approximate interface-tracking methods such as the Volume of Fluid (VOF) and level-set method are utilised. Moreover, since this approach allows for mass transfer across the element surface, discretisation of the convection term introduces strong numerical diffusion.

[8] An alternative strategy is to apply a remeshing technique to avoid highly distorted elements [54].

By combining advantages of both Lagrangian and Eulerian frameworks, a new variant Arbitrary Lagrangian-Eulerian (ALE) was introduced. In this methodology, the mesh is moved in an arbitrary manner such that mesh points are coincident with the material boundary [49, 54, 91]. In this thesis, a Total Lagrangian description of continuum is chosen in presenting the conservation laws, which will be discussed in the following sections.

2.3 Kinematics

Consider the motion of a continuum (see Fig. 2.3) which in its material configuration is defined by a domain $\Omega_0 \subset R^3$ of boundary $\partial\Omega_0$ with outward unit normal $\boldsymbol{N}$. During the motion, the continuum occupies a spatial configuration defined by a domain $\Omega \subset R^3$ of boundary $\partial\Omega$ with outward unit normal $\boldsymbol{n}$. The motion is defined by a time t dependent mapping field $\boldsymbol{\phi}$ which links a material particle from material configuration $\boldsymbol{X} \in \Omega_0$ to spatial configuration $\boldsymbol{x} \in \Omega$ according to $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t)$. The displacement $\boldsymbol{u}$ of a material particle can be described as

$$\boldsymbol{u}(\boldsymbol{X}, t) = \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X}. \tag{2.1}$$

The deformation gradient tensor (or fibre-map) $\boldsymbol{F}$ is defined as the material gradient of spatial configuration and can be expressed as

$$\boldsymbol{F} = \boldsymbol{\nabla}_0 \boldsymbol{x} = \frac{\partial \boldsymbol{\phi}(\boldsymbol{X}, t)}{\partial \boldsymbol{X}} = \boldsymbol{I} + \frac{\partial \boldsymbol{u}(\boldsymbol{X}, t)}{\partial \boldsymbol{X}}, \tag{2.2}$$

where $\boldsymbol{I}$ is a second order identity tensor. Note that $\boldsymbol{F}$ is a two-point tensor which has one basis defined in the current and the other in reference configuration. Geometrically, it maps an elemental material vector $d\boldsymbol{X}$ to its corresponding spatial vector $d\boldsymbol{x}$ as

$$d\boldsymbol{x} = \boldsymbol{F} d\boldsymbol{X}. \tag{2.3}$$

The volume elements in the reference and current configurations are related through the determinant of $\boldsymbol{F}$, denoted as J, and is expressed as

$$dv = JdV; \qquad J = \det(\boldsymbol{F}). \tag{2.4}$$

The cofactor of $\boldsymbol{F}$, expressed as $\boldsymbol{H}$, denotes the relation between area vector in the material and spatial configuration

$$d\boldsymbol{a} = \boldsymbol{H} d\boldsymbol{A}; \qquad \boldsymbol{H} = J\boldsymbol{F}^{-T}. \tag{2.5}$$

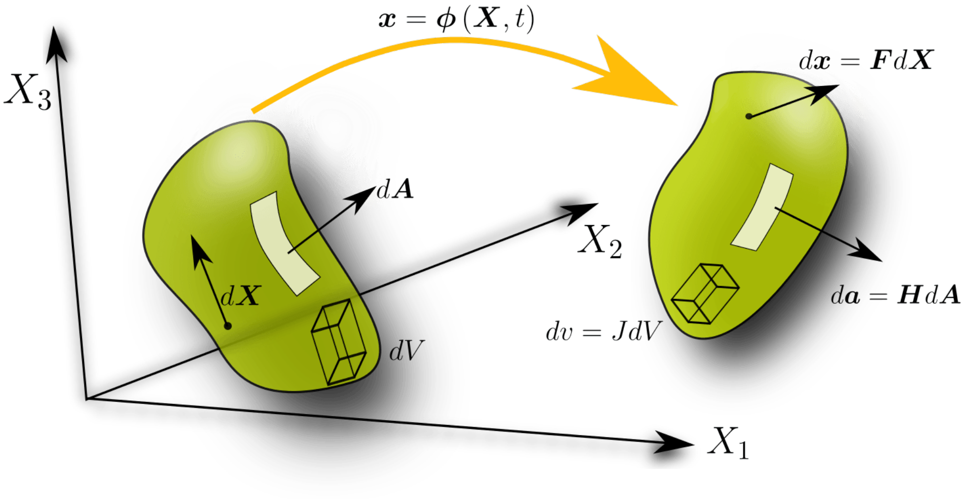

$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{X}, t)$
X_3
$d\boldsymbol{A}$
X_2
$d\boldsymbol{X}$
dV
X_1
$d\boldsymbol{x} = \boldsymbol{F}d\boldsymbol{X}$
$d\boldsymbol{a} = \boldsymbol{H}d\boldsymbol{A}$
$dv = JdV$

FIGURE 2.3: Motion of a continuum body

2.4 Governing equations

Traditionally mechanical systems are characterised by the balance principles of mass, linear momentum $\boldsymbol{p}$ and energy E. Insofar as a mixed formulation is being employed in this thesis, new geometric conservation laws $\{\boldsymbol{F}, \boldsymbol{H}, J\}$are introduced in this section.

2.4.1 Conservation of mass and momentum

The conservation of mass in a Lagrangian setting can be simply stated as

$$\frac{d}{dt}\int_{\Omega_0} \rho_0 \, d\Omega_0 = 0, \tag{2.6}$$

where the initial density of the material ρ_0 is constant and therefore need not be considered as part of the unknown conservation variables. The conservation of linear momentum per unit of undeformed volume can be expressed in the integral form as

$$\frac{d}{dt}\int_{\Omega_0} \boldsymbol{p} \, d\Omega_0 = \int_{\Omega_0} \rho_0 \, \boldsymbol{b} \, d\Omega_0 + \int_{\partial\Omega_0} \boldsymbol{t} \, dA, \tag{2.7}$$

where $\boldsymbol{p} = \rho_0 \boldsymbol{v}$ is the linear momenum per unit of undeformed volume Ω_0, $\boldsymbol{b}$ are the body forces per unit of mass, $\boldsymbol{t}$ denotes the traction vector defined as $\boldsymbol{t} = \boldsymbol{P}\boldsymbol{N}$, $\boldsymbol{P}$ is the first Piola Kirchoff stress tensor and $\boldsymbol{N}$ is the unit material outward normal and A is the surface area in the reference configuration. Applying the Gauss divergence theorem, the local differential linear momentum conservation law can be expressed as

$$\frac{\partial \boldsymbol{p}}{\partial t} = \text{DIV}\boldsymbol{P} + \rho_0 \boldsymbol{b}, \tag{2.8}$$

where DIV denotes the material divergence operator.

2.4.2 Conservation of deformation gradient

Generally, in computational mechanics the deformation gradient tensor is computed directly from the current coordinates $\boldsymbol{x}$ (see Eq. (2.2)). However, when accompanied by low order discretisations, this approach can lead to the appearance of locking, specially in the case of nearly incompressible simulations [92, 93]. Typically, the strains will converge at a rate one order less than the geometry. For this reason, we introduce flexibility into the formulation by considering a new conservation law for the deformation. Differentiation of Eq. (2.2) with respect to time yields

$$\frac{\partial \boldsymbol{F}}{\partial t} = \boldsymbol{\nabla}_0 \boldsymbol{v} = \boldsymbol{\nabla}_0 \left(\frac{\boldsymbol{p}}{\rho_0} \right). \tag{2.9}$$

The integral conservation law for the deformation gradient can be expressed by integrating Eq. (2.9)

$$\frac{d}{dt}\int_{\Omega_0} \boldsymbol{F}\, d\Omega_0 = \int_{\partial\Omega_0} \frac{1}{\rho_0}\boldsymbol{p}\otimes\boldsymbol{N}\, dA. \tag{2.10}$$

Application of the Gauss divergence theorem on the surface integral term in Eq. (2.10) yields the following local differential form

$$\frac{\partial \boldsymbol{F}}{\partial t} = \text{DIV}\left(\frac{1}{\rho_0}\boldsymbol{p}\otimes\boldsymbol{I}\right). \tag{2.11}$$

It is also clear from the definition of $\boldsymbol{F}$ that the conservation law is accompanied by the compatibility condition such as

$$\text{CURL}\boldsymbol{F} = \boldsymbol{0}. \tag{2.12}$$

2.4.3 Conservation of area map

The cofactor of deformation gradient tensor $\boldsymbol{H}$ can also be expressed as [94]

$$\boldsymbol{H} = \frac{1}{2}\boldsymbol{F}\times\boldsymbol{F} = \frac{1}{2}\text{CURL}\left(\boldsymbol{x}\times\boldsymbol{F}\right), \tag{2.13}$$

where $\times$ is defined as the tensor cross product presented in [16, 95, 96] (see Appendix A for more details). Conservation law for the area map can be obtained by integrating Eq. (2.13) as

$$\begin{aligned}\frac{d}{dt}\int_{\Omega_0} \boldsymbol{H}\, d\Omega_0 &= \int_{\Omega_0} \text{CURL}\left(\boldsymbol{v}\times\boldsymbol{F}\right) d\Omega_0 \\ &= -\int_{\partial\Omega_0} \left(\boldsymbol{v}\times\boldsymbol{F}\right)\times\boldsymbol{N}\, dA.\end{aligned} \tag{2.14}$$

It is also clear that the variable $\boldsymbol{H}$ must satisfy the following compatibility condition

$$\text{DIV}\boldsymbol{H} = \boldsymbol{0}. \tag{2.15}$$

In addition, the differential form can be expressed as

$$\frac{\partial \boldsymbol{H}}{\partial t} = \text{CURL}\left(\boldsymbol{v}\times\boldsymbol{F}\right). \tag{2.16}$$

2.4.4 Conservation of volume map

The Jacobian of deformation J is generally computed as follows

$$J = \det(\boldsymbol{F}) = \frac{1}{3}(\boldsymbol{H}:\boldsymbol{F}). \tag{2.17}$$

Another set of conservation laws for the volume map J can be obtained by differentiating Eq. (2.17) with respect to time such that

$$\frac{\partial J}{\partial t} = \text{DIV}\left(\boldsymbol{H}^T \frac{\boldsymbol{p}}{\rho_0}\right). \tag{2.18}$$

2.4.5 Conservation of total energy

Analogously to the CFD case of the Euler equations [54, 97], it is also possible to include the total energy conservation law into our mixed system as

$$\frac{d}{dt}\int_{\Omega_0} E \, d\Omega_0 = \int_{\partial\Omega_0} \frac{1}{\rho_0}\boldsymbol{p}\cdot\boldsymbol{t}\, dA - \int_{\partial\Omega_0} \boldsymbol{Q}\cdot\boldsymbol{N}\, dA + \int_{\Omega_0} s\, d\Omega_0, \tag{2.19}$$

where E is the total energy per unit of undeformed volume and $\boldsymbol{Q}$ and s represent the heat flux and the heat source terms respectively. The local form of the total energy conservation law can be expressed as

$$\frac{\partial E}{\partial t} - \text{DIV}\left(\frac{1}{\rho_0}\boldsymbol{P}^T\boldsymbol{p} - \boldsymbol{Q}\right) = s. \tag{2.20}$$

In the case of an adiabatic deformation, the heat flux $\boldsymbol{Q}$ and the heat source s are neglected. In addition, when dealing with non-thermomechanical materials, Eq. (2.20) is fully decoupled from the rest of the system of conservation laws. Even in this case, from the computational point of view, this equation is still very useful when evaluating the numerical diffusion (entropy) introduced by the algorithm and hence, will be retained in this work.

2.5 Conservation law formulation

It is now possible to combine Eqs. (2.8), (2.11), (2.16), (2.18) and (2.20) into a system of hyperbolic conservation laws expressed in the integral form as

$$\frac{d}{dt}\int_{\Omega_0} \boldsymbol{\mathcal{U}}\, d\Omega_0 + \int_{\partial\Omega_0} \boldsymbol{\mathcal{F}}\, dA = \int_{\Omega_0} \boldsymbol{\mathcal{S}}\, d\Omega_0. \tag{2.21}$$

Eq. (2.21) can be written as a set of first order differential equations

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \frac{\partial \boldsymbol{\mathcal{F}}_I}{\partial X_I} = \boldsymbol{\mathcal{S}}; \qquad \forall\, I = 1, 2, 3, \tag{2.22}$$

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

$$\boldsymbol{\mathcal{U}} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{F} \\ \boldsymbol{H} \\ J \\ E \end{bmatrix}; \qquad \boldsymbol{\mathcal{F}}_{\boldsymbol{N}} = \boldsymbol{\mathcal{F}}_I N_I = - \begin{bmatrix} \boldsymbol{t} \\ \frac{1}{\rho_0}\boldsymbol{p}\otimes\boldsymbol{N} \\ \boldsymbol{F} \times \left(\frac{1}{\rho_0}\boldsymbol{p}\otimes\boldsymbol{N}\right) \\ \boldsymbol{H} : \left(\frac{1}{\rho_0}\boldsymbol{p}\otimes\boldsymbol{N}\right) \\ \frac{1}{\rho_0}\boldsymbol{p}\cdot\boldsymbol{t} - \boldsymbol{Q}\cdot\boldsymbol{N} \end{bmatrix}; \qquad \boldsymbol{\mathcal{S}} = \begin{bmatrix} \rho_0\boldsymbol{b} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ 0 \\ s \end{bmatrix}. \tag{2.23}$$

In the presence of discontinuous solutions, the conservation laws are accompanied by appropriate Rankine-Hugoniot (jump) conditions valid across the discontinuity surface with normal $\boldsymbol{N}$ and propagating with speed c [6, 98, 99] such that

$$c\,[\![\boldsymbol{\mathcal{U}}]\!] = [\![\boldsymbol{\mathcal{F}}]\!]\,\boldsymbol{N}, \tag{2.24}$$

where $[\![\cdot]\!]$ is the jump operator (simply a difference between the two states on either side of an interface). For each individual conservation laws, the jump conditions can be expressed as

$$c\,[\![\boldsymbol{p}]\!] = -[\![\boldsymbol{P}]\!]\boldsymbol{N}; \tag{2.25a}$$

$$c\,[\![\boldsymbol{F}]\!] = -\frac{1}{\rho_0}[\![\boldsymbol{p}]\!]\otimes\boldsymbol{N}; \tag{2.25b}$$

$$c\,[\![\boldsymbol{H}]\!] = -\boldsymbol{F}^{\text{Ave}} \times \left(\frac{1}{\rho_0}[\![\boldsymbol{p}]\!]\otimes\boldsymbol{N}\right); \tag{2.25c}$$

$$c\,[\![J]\!] = -\boldsymbol{H}^{\text{Ave}} : \left(\frac{1}{\rho_0}[\![\boldsymbol{p}]\!]\otimes\boldsymbol{N}\right); \tag{2.25d}$$

$$c\,[\![E]\!] = -\left(\frac{1}{\rho_0}[\![\boldsymbol{P}^T\boldsymbol{p}]\!]\cdot\boldsymbol{N}\right) + [\![\boldsymbol{Q}]\!]\cdot\boldsymbol{N}, \tag{2.25e}$$

where $\boldsymbol{F}^{\text{Ave}}$ and $\boldsymbol{H}^{\text{Ave}}$ are defined as an average of the strain measures across the discontinuity surface.

2.6 Constitutive models

For closure of the system (2.22), it is necessary to introduce an appropriate constitutive model obeying both the laws of thermodynamics and the principle of objectivity [6, 14]. For reversible isothermal elasticity, the first Piola Kirchoff stress tensor can be defined as a function of the deformation gradient tensor such that

$$\boldsymbol{P}(\boldsymbol{F}) = \frac{\partial\psi}{\partial\boldsymbol{F}}, \tag{2.26}$$

where ψ is the strain energy functional per unit of undeformed volume [50, 100]. In general, the strain energy functional $\psi(\boldsymbol{F})$ is not convex in $\boldsymbol{F}$, therefore, the resulting stress-strain relationship is not one-to one. In order to overcome this, it is necessary to introduce constitutive laws defined by means of a polyconvex elastic strain energy. Polyconvexity [94] is a fundamental mathematical requirement that must be satisfied by admissible multi-variable strain energy functions used to describe large strain deformation of elastic materials. The strain energy ψ per unit of undeformed volume must be a function of $\boldsymbol{\nabla}_0\boldsymbol{x}$ via a convex multi-variable function W such that [17]

$$\psi(\boldsymbol{\nabla}_0\boldsymbol{x}) = W(\boldsymbol{F}, \boldsymbol{H}, J), \tag{2.27}$$

where W is convex with respect to its 19 variables. The three strain measures $\{\boldsymbol{F}, \boldsymbol{H}, J\}$ have their corresponding work conjugate stresses $\boldsymbol{\Sigma_F}$, $\boldsymbol{\Sigma_H}$, Σ_J defined by [94]

$$\boldsymbol{\Sigma_F}(\boldsymbol{F}, \boldsymbol{H}, J) = \frac{\partial W}{\partial \boldsymbol{F}}; \qquad \boldsymbol{\Sigma_H}(\boldsymbol{F}, \boldsymbol{H}, J) = \frac{\partial W}{\partial \boldsymbol{H}}; \qquad \Sigma_J(\boldsymbol{F}, \boldsymbol{H}, J) = \frac{\partial W}{\partial J}. \tag{2.28}$$

Moreover, the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ can now be expressed in terms of the geometric strains $\{\boldsymbol{F}, \boldsymbol{H}, J\}$ and the conjugate stresses $\{\boldsymbol{\Sigma_F}, \boldsymbol{\Sigma_H}, \Sigma_J\}$ [17, 96] as

$$\boldsymbol{P} = \boldsymbol{\Sigma_F} + \boldsymbol{\Sigma_H} \times \boldsymbol{F} + \Sigma_J \boldsymbol{H}. \tag{2.29}$$

2.6.1 Compressible polyconvex model

For a compressible Mooney–Rivlin model, an admissible polyconvex strain energy can be defined as [16]

$$W(\boldsymbol{F}, \boldsymbol{H}, J) = \alpha(\boldsymbol{F} : \boldsymbol{F}) + \gamma\,(\boldsymbol{H} : \boldsymbol{H}) + f(J), \tag{2.30}$$

where $J = -4\gamma J - 2\alpha \ln J + \frac{\lambda}{2}(J-1)^2$ and $\{\alpha, \gamma, \lambda\}$ are material parameters defined such that $\alpha + \gamma = \frac{\mu}{2}$ where μ represents the shear modulus. The conjugate stresses can then be expressed as

$$\boldsymbol{\Sigma_F} = 2\alpha\boldsymbol{F}; \qquad \boldsymbol{\Sigma_H} = 2\gamma\boldsymbol{H}; \qquad \Sigma_J = f'(J), \tag{2.31}$$

and utilizing Eq. (2.29), the first Piola-Kirchoff stress tensor can be written as

$$\boldsymbol{P} = 2\alpha\boldsymbol{F} + 2\beta\boldsymbol{H} \times \boldsymbol{F} + f'(J)\boldsymbol{H}. \tag{2.32}$$

2.6.2 Nearly incompressible polyconvex model

In the case of a polyconvex nearly incompressible Mooney-Rivlin material, the multi-variable strain energy W can be decomposed into the summation of deviatoric $\hat{W}(\boldsymbol{F}, \boldsymbol{H}, J)$ and volumetric $U(J)$ contributions [16, 17, 94]:

$$\psi(\nabla_0\boldsymbol{x}) = W(\boldsymbol{F}, \boldsymbol{H}, J) = \hat{W} + U, \tag{2.33}$$

with

$$\hat{W} = \zeta J^{-2/3}\,(\boldsymbol{F} : \boldsymbol{F}) + \xi J^{-2}\,(\boldsymbol{H} : \boldsymbol{H})^{3/2} - 3\zeta - 3^{3/2}\xi; \qquad U = \frac{\kappa}{2}(J-1)^2, \tag{2.34}$$

where ζ, ξ and κ (bulk modulus) are positive material parameters. By comparison of the tangent elasticity operator at the initial undeformed configuration with that of classical linear elasticity [94], appropriate values for the material parameters ζ and ξ can be defined in terms of the shear modulus μ, that is, $2\zeta + 3\sqrt{3}\xi = \mu$ [17–19]. Utilizing Eq. (2.28), the conjugate stresses yield

$$\boldsymbol{\Sigma}_{\boldsymbol{F}} := \frac{\partial \hat{W}}{\partial \boldsymbol{F}} = 2\zeta J^{-2/3}\boldsymbol{F}; \qquad \boldsymbol{\Sigma}_{\boldsymbol{H}} := \frac{\partial \hat{W}}{\partial \boldsymbol{H}} = 3\xi J^{-2}\left(\boldsymbol{H} : \boldsymbol{H}\right)^{1/2}\boldsymbol{H}, \tag{2.35}$$

and $\Sigma_J := \hat{\Sigma}_J + p$ with

$$\hat{\Sigma}_J := \frac{\partial \hat{W}}{\partial J} = -\frac{2}{3}\zeta J^{-5/3}(\boldsymbol{F} : \boldsymbol{F}) - 2\xi J^{-3}(\boldsymbol{H} : \boldsymbol{H})^{3/2}; \qquad p := \frac{\partial U}{\partial J} = \kappa(J-1). \tag{2.36}$$

It is worth pointing out that the multi-variable strain energy described in (2.34) degenerates to the case of a polyconvex nearly incompressible neo-Hookean model by imposing the values of $\zeta = \frac{\mu}{2}$ and $\xi = 0$ such that

$$\boldsymbol{P}_{dev} = \mu J^{-2/3}\boldsymbol{F} - \frac{\mu}{3}J^{-5/3}(\boldsymbol{F} : \boldsymbol{F})\boldsymbol{H}; \qquad \boldsymbol{P}_{vol} = p\boldsymbol{H}; \qquad p = \kappa(J-1). \tag{2.37}$$

Remark 1: The classical nearly incompressible neo-Hookean model [50] can be simply recovered by replacing J with $J_{\boldsymbol{F}}$ and $\boldsymbol{H}$ with $\boldsymbol{H}_{\boldsymbol{F}}$ in Eq. (2.37) such that

$$\boldsymbol{P}_{\text{dev}} = \mu J_{\boldsymbol{F}}^{-2/3}\left(\boldsymbol{F} - \frac{1}{3}(\boldsymbol{F} : \boldsymbol{F})\boldsymbol{F}^{-T}\right); \qquad \boldsymbol{P}_{\text{vol}} = p\boldsymbol{H}_{\boldsymbol{F}}; \qquad p = \kappa(J_{\boldsymbol{F}} - 1), \tag{2.38}$$

where $J_{\boldsymbol{F}} = \det \boldsymbol{F}$ is the Jacobian based on deformation gradient tensor $\boldsymbol{F}$ and $\boldsymbol{H}_{\boldsymbol{F}} = \frac{1}{2}\,\boldsymbol{F} \times \boldsymbol{F}$ is the cofactor of deformation based on $\boldsymbol{F}$.

2.6.3 Isothermal elasto-plastic model

In this study, consideration of irreversible processes is restricted to isothermal elasto-plastic materials typically used in metal forming applications [101]. In this particular work, thermal effects will be neglected (refer to [11] for the consideration of thermal effects). In order to model irrecoverable plastic behaviour, the standard rate-independent von Mises plasticity with isotropic hardening is used [9]. In this case, the deformation gradient tensor $\boldsymbol{F}$ can be decomposed

[9] The term ’rate-independent’ implies that the stress is not a function of the strain rate as would be the case for a viso-plastic model [50]. The term ’isotropic hardening’ means that the yield stress increases equally in all directions due to tensile or compressive loading [50].

into elastic $\boldsymbol{F}_e$ and plastic $\boldsymbol{F}_p$ components such that

$$\boldsymbol{F} = \boldsymbol{F}_e\,\boldsymbol{F}_p, \tag{2.39}$$

where the elastic left strain tensor is defined as $\boldsymbol{b}_e = \boldsymbol{F}\,\boldsymbol{C}_p^{-1}\boldsymbol{F}^T$ and the plastic right Cauchy Green tensor as $\boldsymbol{C}_p = \boldsymbol{F}_p^T\boldsymbol{F}_p$. Moreover, it is essential to define a strain energy functional in terms of the elastic principal stretches $(\lambda_e^1,\, \lambda_e^2,\, \lambda_e^3)$ as

$$\psi(\lambda_e^1,\, \lambda_e^2,\, \lambda_e^3, J) = \psi_{\text{dev}}(J^{-1/3}\lambda_e^1,\, J^{-1/3}\lambda_e^2,\, J^{-1/3}\lambda_e^3) + \psi_{\text{vol}}(J), \tag{2.40}$$

where

$$\psi_{\text{dev}} = \mu\left[(\ln\lambda_e^1)^2 + (\ln\lambda_e^2)^2 + (\ln\lambda_e^3)^2\right] - \frac{\mu}{3}(\ln J)^2, \tag{2.41}$$

and

$$\psi_{\text{vol}} = \frac{\kappa}{2}(\ln J)^2; \qquad \kappa = \lambda + \frac{2}{3}\mu; \qquad \ln J = \sum_{\alpha=1}^{3}\ln\lambda_e^\alpha. \tag{2.42}$$

The basic structure of this isothermal elasto-plastic model is summarised in Algorithm 2.1 (refer to [50] for further details).

2.7 Eigenvalue structure

A hyperbolic problem has a wave like solution [10]. We expect the information to propagate as waves moving along the characteristics. For a system to be considered hyperbolic it is necessary that, in the absence of source term $\boldsymbol{\mathcal{S}}$, the flux Jacobian matrix $\boldsymbol{\mathcal{A}}_{\boldsymbol{N}}$ is diagonalizable with real eigenvalues [97, 103]. From a physical point of view, the satisfaction of hyperbolicity (also called rank-one convexity [14]) ensures the existence of physical (real) waves travelling throughout the domain and hence, the well-posedness of the problem.

Hyperbolicity of the system of conservation laws (2.22) in combination with a constitutive model (see Section 2.6), is of paramount importance especially when considering elastic materials undergoing extremely large deformations [94]. For completeness, the study of eigenvalue structure of the mixed system (2.22) is presented to demonstrate the hyperbolicity of the system of conservation laws. In addition, the computation of the maximum (pressure) wave speed is necessary for the evaluation of the maximum time step of the explicit time integrator. Consider

[10] Hyperbolic problems do not include diffusive or viscous effects [102].

Algorithm 2.1: Time update of first Piola Kirchoff stress tensor

```
Input  : $\boldsymbol{F}^{n+1}, \left[\boldsymbol{C}_p^{-1}\right]^n, \varepsilon_p^n$
Output: $\boldsymbol{P}^{n+1}$
 (1) Obtain Jacobian of deformation:   $J_{\boldsymbol{F}}^{n+1} = \det\left(\boldsymbol{F}^{n+1}\right)$
 (2) Evaluate pressure:   $p^{n+1} = \kappa \frac{\ln\left(J_{\boldsymbol{F}}^{n+1}\right)}{J_{\boldsymbol{F}}^{n+1}}$
 (3) Compute trial elastic left strain tensor:   $\bar{\boldsymbol{b}}_e^{n+1} = \boldsymbol{F}^{n+1}\left[\boldsymbol{C}_p^{-1}\right]^n\left[\boldsymbol{F}^T\right]^{n+1}$
 (4) Spectral decomposition of $\bar{\boldsymbol{b}}_e^{n+1}$:   $\bar{\lambda}_e^i\ ,\ \bar{\boldsymbol{n}}_i \leftarrow \bar{\boldsymbol{b}}_e^{n+1} = \sum_{i=1}^3 \left(\bar{\lambda}_e^i\right)^2 \left(\bar{\boldsymbol{n}}_i \otimes \bar{\boldsymbol{n}}_i\right)$
 (5) Obtain trial deviatoric Kirchoff stress tensor:
     $\bar{\boldsymbol{\tau}}' = \sum_{i=1}^3 \bar{\tau}'_{ii}\left(\bar{\boldsymbol{n}}_i \otimes \bar{\boldsymbol{n}}_i\right);\qquad \bar{\tau}'_{ii} = 2\mu\ \ln\left(\bar{\lambda}_e^i\right) - \frac{2}{3}\mu\ \ln\left(J_{\boldsymbol{F}}^{n+1}\right)$
 (6) Obtain yield criterion:   $f\left(\bar{\boldsymbol{\tau}}', \varepsilon_p^n\right) = \sqrt{\frac{3}{2}(\bar{\boldsymbol{\tau}}' : \bar{\boldsymbol{\tau}}')} - \left(\tau_y^0 + H\varepsilon_p^n\right)$
 (7) Compute direction vector and plastic multiplier:
     if $f\left(\bar{\boldsymbol{\tau}}', \varepsilon_p^n\right) > 0$ then
         □ Direction vector:   $\boldsymbol{v}_i^{n+1} = \frac{\bar{\tau}'_{ii}}{\sqrt{\frac{2}{3}(\bar{\boldsymbol{\tau}}' : \bar{\boldsymbol{\tau}}')}}$
         □ Plastic multiplier:   $\mathit{\Delta}\gamma = \frac{f\left(\bar{\boldsymbol{\tau}}', \varepsilon_p^n\right)}{3\mu + H}$
     else
         □ $\boldsymbol{v}_i^{n+1} = \mathit{\Delta}\gamma = 0$
     end
 (8) Evaluate elastic stretch:   $\lambda_e^{i,n+1} = \exp\left(\ln(\bar{\lambda}_e^i) - \mathit{\Delta}\gamma\, \boldsymbol{v}_i^{n+1}\right)$
 (9) Set spatial normals:   $\boldsymbol{n}_i^{n+1} = \bar{\boldsymbol{n}}_i$
(10) Compute Kirchoff stresss tensor:   $\boldsymbol{\tau}^{n+1} = \sum_{i=1}^3 \tau_{ii}\left(\boldsymbol{n}_i^{n+1} \otimes \boldsymbol{n}_i^{n+1}\right)$
     $\tau_{ii} = \tau'_{ii} + J_{\boldsymbol{F}}^{n+1} p^{n+1};\qquad \tau'_{ii} = \left(1 - \frac{2\mu\,\mathit{\Delta}\gamma}{\sqrt{\frac{2}{3}(\bar{\boldsymbol{\tau}}' : \bar{\boldsymbol{\tau}}')}}\right)\bar{\tau}'_{ii}$
(11) Evaluate first Piola Kirchoff stress tensor:   $\boldsymbol{P}^{n+1} = \boldsymbol{\tau}^{n+1}\left[\boldsymbol{F}^{-T}\right]^{n+1}$
(12) Update elastic left strain tensor:   $\boldsymbol{b}_e^{n+1} = \sum_{i=1}^3 \left(\lambda_e^{i,n+1}\right)^2 \left(\boldsymbol{n}_i^{n+1} \otimes \boldsymbol{n}_i^{n+1}\right)$
(13) Compute plastic right Cauchy Green tensor:   $\left[\boldsymbol{C}_p^{-1}\right]^{n+1} = \left[\boldsymbol{F}^{-1}\right]^{n+1} \boldsymbol{b}_e^{n+1} \left[\boldsymbol{F}^{-T}\right]^{n+1}$
(14) Update plastic strain:   $\varepsilon_p^{n+1} = \varepsilon_p^n + \mathit{\Delta}\gamma$
```

the plane wave solutions of the form

$$\boldsymbol{\mathcal{U}} \;=\; \phi\left(\boldsymbol{X}\cdot\boldsymbol{N}-c_\alpha t\right)\boldsymbol{\mathcal{R}}_\alpha \;=\; \phi\left(\boldsymbol{X}\cdot\boldsymbol{N}-c_\alpha t\right)\begin{bmatrix}\boldsymbol{p}_\alpha^R\\ \boldsymbol{F}_\alpha^R\\ \boldsymbol{H}_\alpha^R\\ J_\alpha^R\end{bmatrix}, \tag{2.43}$$

where c_α are the wave speeds associated with the corresponding right eigenvectors $\boldsymbol{\mathcal{R}}_\alpha$ and the normalised direction of propagation $\boldsymbol{N}$. A compressible Mooney Rivlin constitutive model given by Eq. (2.30) is considered for simplicity. With the aid of Eq. (4.46) and Eq. (2.30), the modified eigen-problem by considering each individual component of this system becomes

$$c_\alpha\boldsymbol{p}_\alpha = -2\alpha\boldsymbol{F}_\alpha\boldsymbol{N} - \left(2\gamma\boldsymbol{F}\times\boldsymbol{H}_\alpha\right)\boldsymbol{N} - W_{JJ}J_\alpha\boldsymbol{H}\boldsymbol{N}; \tag{2.44a}$$

$$c_\alpha\boldsymbol{F}_\alpha = -\frac{1}{\rho_0}\left(\boldsymbol{p}_\alpha\otimes\boldsymbol{N}\right); \tag{2.44b}$$

$$c_\alpha\boldsymbol{H}_\alpha = -\frac{1}{\rho_0}\left[\boldsymbol{F}\times\left(\boldsymbol{p}_\alpha\otimes\boldsymbol{N}\right)\right]; \tag{2.44c}$$

$$c_\alpha J_\alpha = -\frac{1}{\rho_0}\boldsymbol{p}_\alpha\cdot\boldsymbol{H}\boldsymbol{N}, \tag{2.44d}$$

with $W_{JJ}=\lambda+\frac{2\alpha}{J^2}$. As a consequence of the high level of redundancy in the system of equations being considered, only six wave speeds are different from zero. These can be readily identified by substituting the last three geometric strain equations (4.49b-4.49d) into (4.49a) to give:

$$\left(2\alpha\boldsymbol{p}_\alpha + W_{JJ}(\boldsymbol{H}\boldsymbol{N}\otimes\boldsymbol{H}\boldsymbol{N})\boldsymbol{p}_\alpha + 2\gamma\left[\boldsymbol{F}\times\left(\boldsymbol{F}\times\left(\boldsymbol{p}_\alpha\otimes\boldsymbol{N}\right)\right)\right]\boldsymbol{N}\right) = \rho_0 c_\alpha^2\boldsymbol{p}_\alpha. \tag{2.45}$$

The double tensor cross product term can be simplified by repeated use of the third order alternating tensor product expression $\mathcal{E}_{ijk}\mathcal{E}_{lmk}=\delta_{il}\delta_{jm}-\delta_{im}\delta_{jl}$ and noting that

$$\boldsymbol{I}-\boldsymbol{N}\otimes\boldsymbol{N} = \boldsymbol{T}_1\otimes\boldsymbol{T}_1+\boldsymbol{T}_2\otimes\boldsymbol{T}_2 \tag{2.46}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{T}_{1,2}$ denote an arbitrary pair of orthogonal unit vectors on the reference propagation plane with normal $\boldsymbol{N}$. With the help of these expressions the eigenvalue problem becomes

$$\left[2\alpha\boldsymbol{p}_\alpha + W_{JJ}\mathit{\Lambda}_A^2\left(\boldsymbol{n}\otimes\boldsymbol{n}\right)\boldsymbol{p}_\alpha + 2\gamma\left(\mathit{\Lambda}_T^2\boldsymbol{I}-\boldsymbol{\Lambda}_T\right)\boldsymbol{p}_\alpha\right] = \rho_0 c_\alpha^2\boldsymbol{p}_\alpha, \tag{2.47}$$

where the following notations have been used

$$\begin{aligned}\mathit{\Lambda}_A\boldsymbol{n} &= \boldsymbol{H}\boldsymbol{N};\\ \mathit{\Lambda}_A^2 &= \boldsymbol{H}\boldsymbol{N}\cdot\boldsymbol{H}\boldsymbol{N};\\ \boldsymbol{\Lambda}_T &= \boldsymbol{F}\boldsymbol{T}_1\otimes\boldsymbol{F}\boldsymbol{T}_1+\boldsymbol{F}\boldsymbol{T}_2\otimes\boldsymbol{F}\boldsymbol{T}_2;\\ \mathit{\Lambda}_T^2 &= \boldsymbol{F}\boldsymbol{T}_1\cdot\boldsymbol{F}\boldsymbol{T}_1+\boldsymbol{F}\boldsymbol{T}_2\cdot\boldsymbol{F}\boldsymbol{T}_2 = \mathrm{tr}\boldsymbol{\Lambda}_T.\end{aligned} \tag{2.48}$$

Note that $\boldsymbol{n}$ is a unit vector orthogonal to the vectors $\boldsymbol{FT}_{1,2}$ which lie on the propagation surface.

The first set of eigenvalues corresponding to p-waves is obtained by taking $\boldsymbol{p}_\alpha = \boldsymbol{n}$ to give,

$$c_{1,2} = c_p; \qquad c_p = \pm\sqrt{\frac{\left(2\alpha + 2\gamma \mathit{\Lambda}_T^2 + W_{JJ}\mathit{\Lambda}_A^2\right)}{\rho_0}}. \tag{2.49}$$

The remaining four eigenvalues correspond to shear waves where the vibration takes place on the propagation plane. The corresponding velocity vectors are orthogonal to $\boldsymbol{n}$ and in the directions of the unit eigenvectors $\{\boldsymbol{t}_1,\ \boldsymbol{t}_2\}$ of the rank two tensor $\boldsymbol{\Lambda}_T$. The wave speeds are given by $c_{3,4} = c_{s_1}$ and $c_{5,6} = c_{s_2}$, where

$$c_{s_1} = \pm\sqrt{\frac{\left(2\alpha + 2\gamma\left(\mathit{\Lambda}_T^2 - \lambda_1^2\right)\right)}{\rho_0}}; \qquad c_{s_2} = \pm\sqrt{\frac{\left(2\alpha + 2\gamma\left(\mathit{\Lambda}_T^2 - \lambda_2^2\right)\right)}{\rho_0}} \tag{2.50}$$

and $\lambda_{1,2}^2$ are the eigenvalues of $\boldsymbol{\Lambda}_T$. In the case of linear elasticity (e.g. $\alpha = \frac{\mu}{2}, \gamma = 0$), both the elastic pressure and shear wave speeds in the undeformed configuration ($\boldsymbol{F} = \boldsymbol{H} = \boldsymbol{I}$) become

$$c_{1,2} = \sqrt{\frac{\lambda + 2\mu}{\rho_0}}; \qquad c_{3,4} = c_{5,6} = \sqrt{\frac{\mu}{\rho_0}}. \tag{2.51}$$

Utilising equations (4.49), the corresponding right eigenvectors are obtained as

$$\boldsymbol{\mathcal{R}}_{1,2} = \begin{bmatrix} \boldsymbol{n} \\ -\frac{1}{\rho_0 c_{1,2}}\boldsymbol{n}\otimes\boldsymbol{N} \\ -\frac{1}{\rho_0 c_{1,2}}\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{n}\otimes\boldsymbol{N}) \\ -\frac{\mathit{\Lambda}_A}{\rho_0 c_{1,2}} \end{bmatrix}; \qquad \boldsymbol{\mathcal{R}}_{3,4} = \begin{bmatrix} \boldsymbol{t}_1 \\ -\frac{1}{\rho_0 c_{3,4}}\boldsymbol{t}_1\otimes\boldsymbol{N} \\ -\frac{1}{\rho_0 c_{3,4}}\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{t}_1\otimes\boldsymbol{N}) \\ 0 \end{bmatrix};$$

$$\boldsymbol{\mathcal{R}}_{5,6} = \begin{bmatrix} \boldsymbol{t}_2 \\ -\frac{1}{\rho_0 c_{5,6}}\boldsymbol{t}_2\otimes\boldsymbol{N} \\ -\frac{1}{\rho_0 c_{5,6}}\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{t}_2\otimes\boldsymbol{N}) \\ 0 \end{bmatrix}. \tag{2.52}$$

Finally, the following set of left eigenvectors can also be obtained in an analogous manner

$$\boldsymbol{\mathcal{L}}_{1,2} = \begin{bmatrix} \boldsymbol{n} \\ -\frac{2\alpha}{c_{1,2}}\left[\boldsymbol{n}\otimes\boldsymbol{N}\right] \\ -\frac{2\gamma}{c_{1,2}}\left[\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{n}\otimes\boldsymbol{N})\right] \\ -\frac{W_{JJ}\Lambda_A}{c_{1,2}} \end{bmatrix}; \qquad \boldsymbol{\mathcal{L}}_{3,4} = \begin{bmatrix} \boldsymbol{t}_1 \\ -\frac{2\alpha}{c_{3,4}}\left[\boldsymbol{t}_1\otimes\boldsymbol{N}\right] \\ -\frac{2\gamma}{c_{3,4}}\left[\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{t}_1\otimes\boldsymbol{N})\right] \\ 0 \end{bmatrix}; \tag{2.53}$$
$$\boldsymbol{\mathcal{L}}_{5,6} = \begin{bmatrix} \boldsymbol{t}_2 \\ -\frac{2\alpha}{c_{5,6}}\left[\boldsymbol{t}_2\otimes\boldsymbol{N}\right] \\ -\frac{2\gamma}{c_{5,6}}\left[\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{t}_2\otimes\boldsymbol{N})\right] \\ 0 \end{bmatrix}.$$

Chapter 3

FINITE VOLUME SPATIAL DISCRETISATION

3.1 Preliminaries

In Chapter 2, the problem was formulated as a system of first order hyperbolic conservation laws (Section 2.5). The hyperbolic system can now be spatially discretised using any of the available CFD methodologies. In this work an upwind cell centred Finite Volume Method is presented in Section 3.2. Specifically, both the standard Godunov-type CCFVM and node-based FVM are discussed. In order to obtain second order spatial accuracy, a linear reconstruction procedure [104] is described in Section 3.4. The layout of this chapter is shown in Fig. 3.1.

3.2 Finite volume methodology

Finite Volume Method is a numerical technique for spatial discretisation through which an integral formulation of the conservation laws is directly discretised in the physical space. Finite Volume Methods were first introduced in 1970s by McDonald, MacCormack and Paullay for the numerical solution of fluid mechanics problem [21, 22]. Since then the Finite Volume Method is the most popular discretisation technique in CFD community. FVM, unlike FEM, is based on the notion of conservative discretisation which takes into account the integral formulation of the conservation laws. This is perhaps one of its biggest advantages, since it automatically satisfies the local conservation property of the primary variables such as mass, momentum

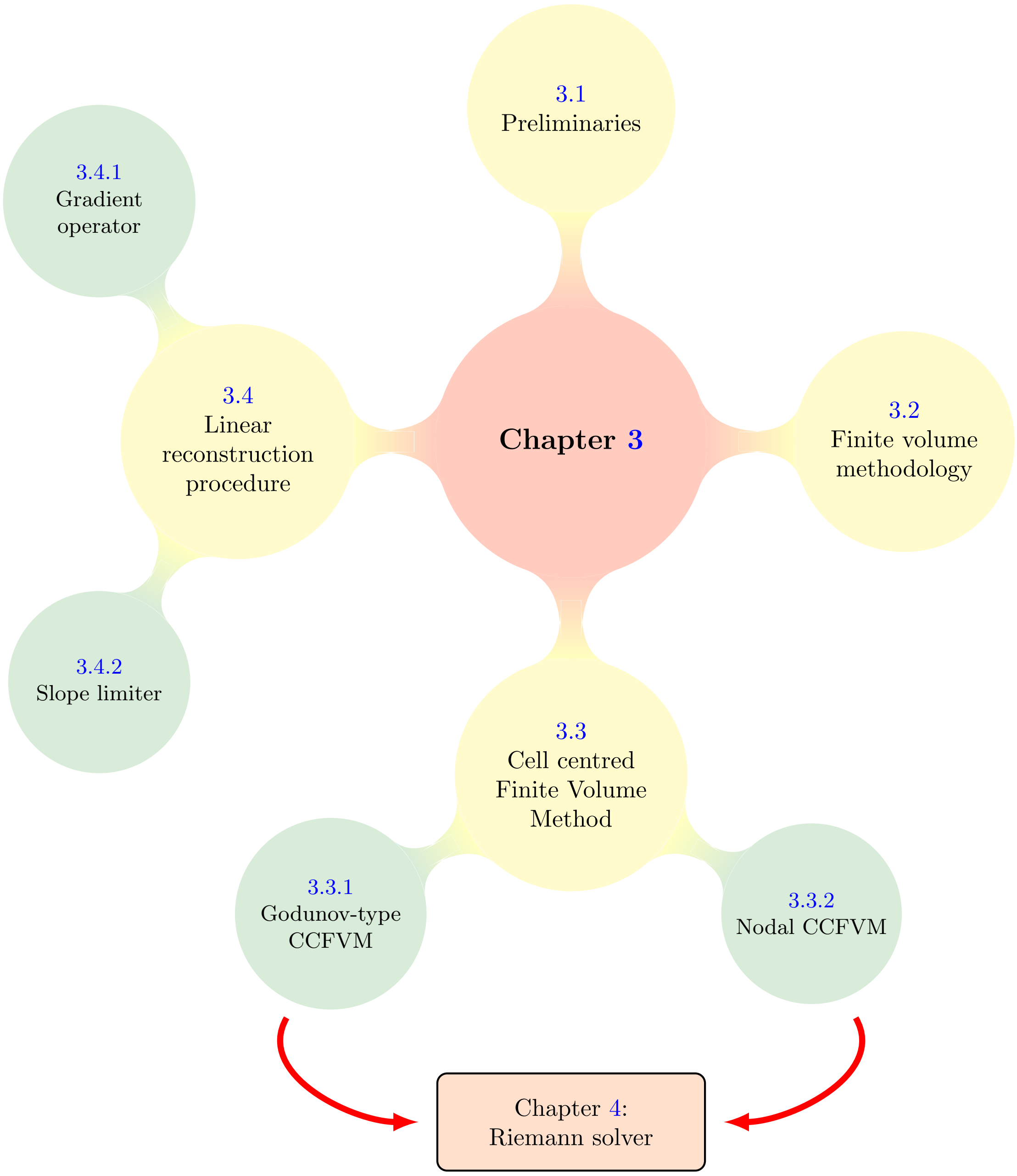
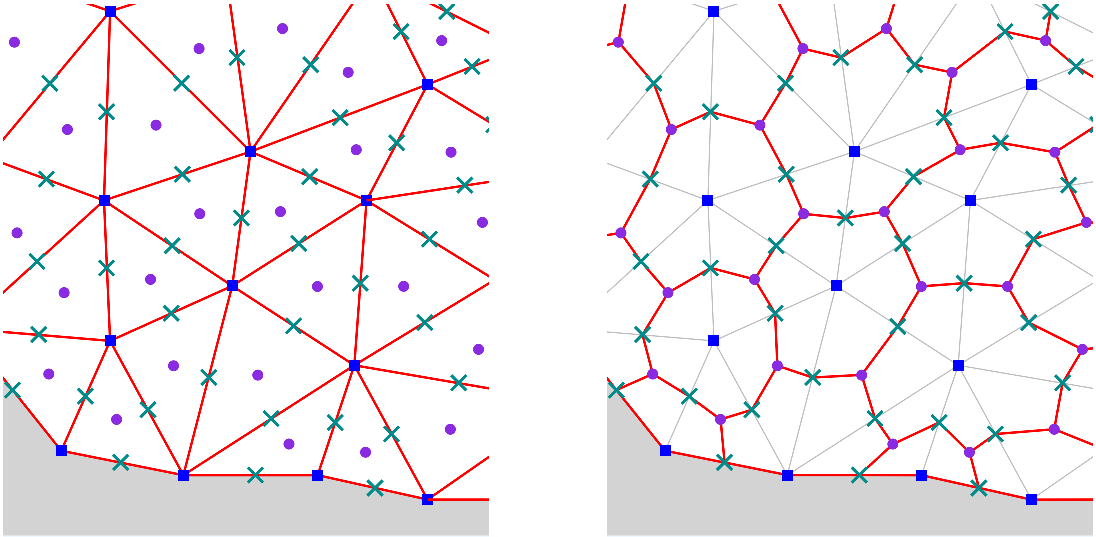


FIGURE 3.1: Structure of Chapter 3



(a) Cell centred FVM

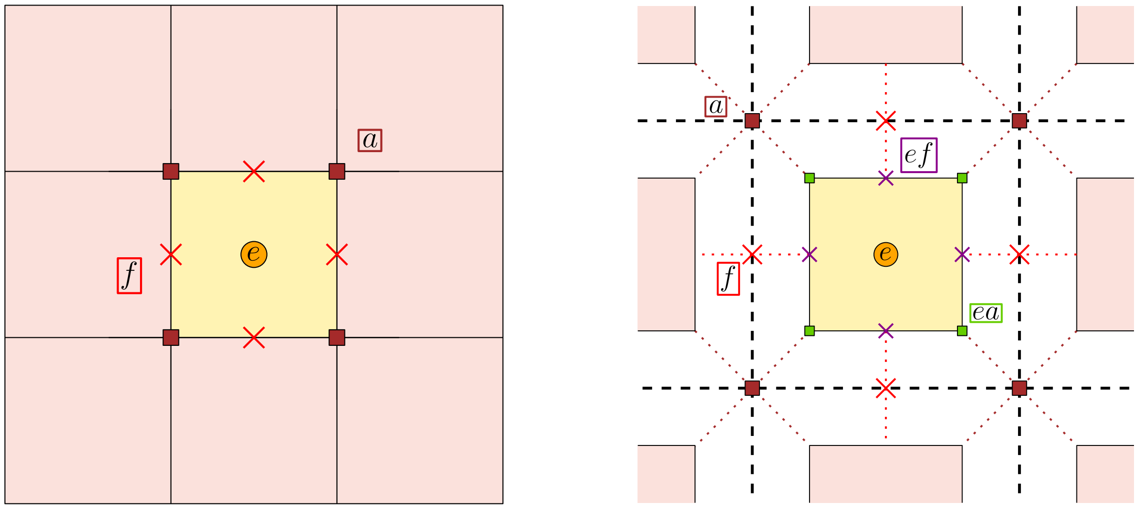
(b) Vertex centred FVM

Figure 3.2: The two categories of Finite Volume Methods: (a) cell centred FVM; and (b) vertex centred FVM.

and energy at the discrete level [21, 105, 106]. Moreover, the Finite Volume Method can be easily applied to arbitrary grids, either structured or unstructured. The Finite Volume Method discretises the governing equations by dividing the physical space into a number of arbitrary local finite volumes of the domain which are known as control volumes. Once established, it is at the center of these control volumes that the conservation variables are stored (cell-averaged values) which distinguishes it from finite difference and finite element methods, where the main numerical quantities are the local function values at the mesh points [21].

There are two major categories of defining these control volumes where the integral conservation laws are applied. The first approach is the more traditional cell centred scheme where the control volumes coincide with the grid cells. Since the conservation variables are stored at the center of the grid cells, hence it is given the name cell centred. The second approach is called the vertex centred (or cell vertex) scheme where the variables are stored at the grid points. In this case the control volume can then either be the union of all cells connected to a grid point giving rise to overlapping control volumes or some volume centred around the grid point thereby creating a dual (non-overlapping) control volume [21, 105]. Generally, construction of a dual mesh is achieved by median-dual tessellation, where sub-volumes are created by joining the centroids of the elements and midpoints of the edges [106].

Both cell centred and vertex centred schemes are currently used in practice with some researchers tilting their preference towards one or the other. The advantage of cell centred scheme is that dealing interfaces is much more easier since control volumes are clearly defined for different mediums. On the other hand, advantage of the vertex centred scheme is in its excellent handling of boundaries since the unknown quantities can be explicitly specified at the physical boundary. The focus of this study is aimed at cell centred FVM which will be further explored in the following section.



(a)

(b)

FIGURE 3.3: Nomenclature used for spatial discretisation

3.3 Cell centred Finite Volume Method

In the cell centred finite volume approach described in this thesis, the control volume coincides with the cell. Additionally, for the ease of understanding, a nomenclature is introduced in this chapter to elaborate on the finite volume discretisation. In Fig. 3.3a, a quadrilateral mesh is shown where e represents the cell under consideration, f denotes the face centre and a represents the node. Moreover, in order to correctly identify additional quantities, an exploded view of the same mesh is shown in Fig. 3.3b where ef represents the elemental surface (face f associated to cell e) and ea denotes elemental node (node a associated to cell e). Additionally, the subscript 'C' denotes the contact or interface flux obtained through a Riemann solver.

3.3.1 Godunov-type CCFVM

The system of conservation laws (2.22) is discretised using the standard face-based CCFVM for hexahedral meshes (see Fig. 3.4a). Unlike staggered approaches [92, 93], all the primary variables are computed at the centroid of each cell (also known as control volume). The spatial discretisation of this system for an arbitrary cell e can be generally approximated as

$$\begin{aligned} \frac{d\boldsymbol{\mathcal{U}}_e}{dt} &= -\frac{1}{\Omega_0^e}\int_{\Omega_0^e} \frac{\partial \boldsymbol{\mathcal{F}}_I}{\partial X_I}\, d\Omega_0 \\ &= -\frac{1}{\Omega_0^e}\int_{\partial\Omega_0^e} \boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\, dA \\ &\approx -\frac{1}{\Omega_0^e}\sum_{f\in\Lambda_e^f}\sum_{g\in\Lambda_f^g} \boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{eg}}(\boldsymbol{\mathcal{U}}^-_{eg},\boldsymbol{\mathcal{U}}^+_{eg})\,\|\boldsymbol{C}_{eg}\|, \end{aligned} \tag{3.1}$$

where use has been made of the Gauss divergence theorem and of equation. In the above equations, $\varOmega_0^e$ denotes the control volume associated to cell e, $\varLambda_e^f$ represents the set of surfaces f associated to cell e, $\varLambda_f^g$ represents the set of Gauss quadrature points g associated to surface f, $\boldsymbol{N}_{eg} := \boldsymbol{C}_{eg}/\|\boldsymbol{C}_{eg}\|$ and $\|\boldsymbol{C}_{eg}\|$ denote the material outward normal and the surface area associated to Gauss quadrature point g of cell e, respectively, and $\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{eg}}(\boldsymbol{\mathcal{U}}^-_{eg},\boldsymbol{\mathcal{U}}^+_{eg})$ represents the numerical (contact) flux computed using the left and right states of variable $\boldsymbol{\mathcal{U}}$ at Gauss quadrature point g, namely $\boldsymbol{\mathcal{U}}^-_{eg}$ and $\boldsymbol{\mathcal{U}}^+_{eg}$. When evaluating the surface flux integral Eq. (3.1), we will restrict ourselves to the consideration of a single Gauss quadrature point for exact integration, seeking computational efficiency of the overall scheme. In this case, the reduced form of the variable update Eq. (3.1) takes the form

$$\frac{d\boldsymbol{\mathcal{U}}_e}{dt} = -\frac{1}{\varOmega_0^e}\sum_{f\in\varLambda_e^f}\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}^-_{ef},\boldsymbol{\mathcal{U}}^+_{ef})\;\|\boldsymbol{C}_{ef}\|, \tag{3.2}$$

where $\boldsymbol{N}_{ef} := \boldsymbol{C}_{ef}/\|\boldsymbol{C}_{ef}\|$ and $\|\boldsymbol{C}_{ef}\|$ represent the material outward normal and the surface area associated to face f of cell e and $\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}^-_{ef},\boldsymbol{\mathcal{U}}^+_{ef})$ represents the numerical (contact) flux computed using the left and right states of variable $\boldsymbol{\mathcal{U}}$ at face f, namely $\boldsymbol{\mathcal{U}}^-_{ef}$ and $\boldsymbol{\mathcal{U}}^+_{ef}$. Above expression can now be particularised for the individual components of $\boldsymbol{\mathcal{U}}$, yielding

$$\frac{d\boldsymbol{p}_e}{dt} = \frac{1}{\varOmega_0^e}\sum_{f\in\varLambda_e^f}\boldsymbol{t}_f^C\|\boldsymbol{C}_{ef}\|; \tag{3.3a}$$

$$\frac{d\boldsymbol{F}_e}{dt} = \frac{1}{\varOmega_0^e}\sum_{f\in\varLambda_e^f}\frac{\boldsymbol{p}_f^C}{\rho_0}\otimes\boldsymbol{C}_{ef}; \tag{3.3b}$$

$$\frac{d\boldsymbol{H}_e}{dt} = \boldsymbol{F}_e \boldsymbol{\times} \frac{1}{\varOmega_0^e}\sum_{f\in\varLambda_e^f}\frac{\boldsymbol{p}_f^C}{\rho_0}\otimes\boldsymbol{C}_{ef}; \tag{3.3c}$$

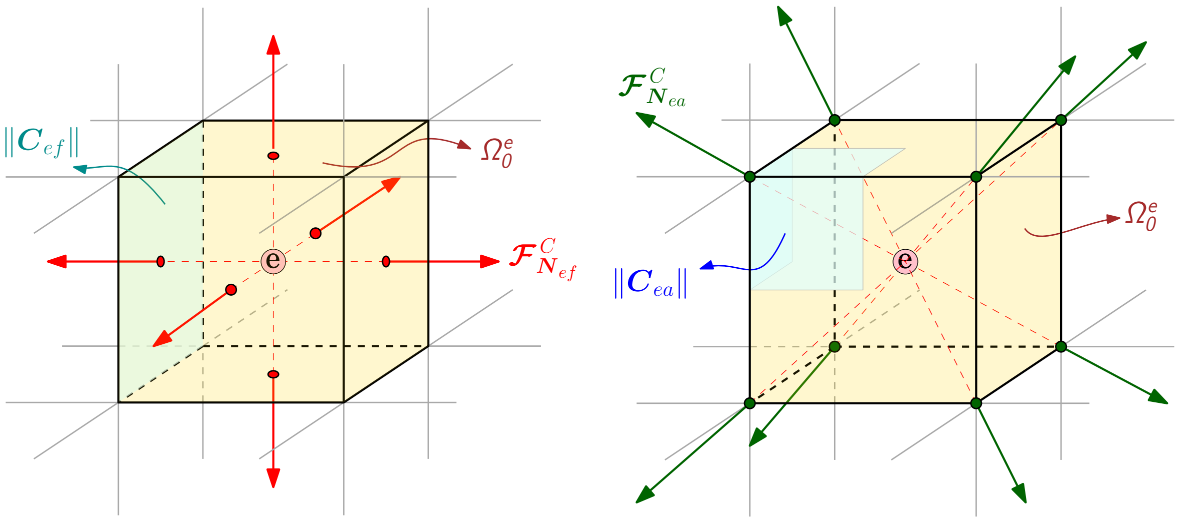
$$\frac{dJ_e}{dt} = \boldsymbol{H}_e : \frac{1}{\varOmega_0^e}\sum_{f\in\varLambda_e^f}\frac{\boldsymbol{p}_f^C}{\rho_0}\otimes\boldsymbol{C}_{ef}; \tag{3.3d}$$

$$\frac{dE_e}{dt} = \frac{1}{\varOmega_0^e}\sum_{f\in\varLambda_e^f}\left(\frac{\boldsymbol{p}_f^C}{\rho_0}\cdot\boldsymbol{t}_f^C\right)\|\boldsymbol{C}_{ef}\|, \tag{3.3e}$$

where $\boldsymbol{t}_f^C = \boldsymbol{P}_f^C\boldsymbol{N}_{ef}$ is the contact traction. Moreover, the geometry can be updated through the linear momentum as

$$\frac{d\boldsymbol{x}_e}{dt} = \frac{\boldsymbol{p}_e}{\rho_0}. \tag{3.4}$$

One of the most challenging issues in updating Eqs. (3.3a) to (3.3d) is the ability to control spurious mechanisms over a long term response. In order to tackle this problem, two different strategies are proposed in this study where the geometric strain updates described in Eqs. (3.3b) to (3.3d) are solved subjected to the fulfilment of specific involutions (Eqs. (2.12) and (2.15)) [6, 8, 107]. The first algorithm, named as Constrained-TOUCH (abbreviated to C-TOUCH), is to replace the Godunov-type numerical linear momentum $\boldsymbol{p}_f^C$ described in


$\|\boldsymbol{C}_{ef}\|$
Ω_0^e
e
$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ef}}$
$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ea}}$
$\|\boldsymbol{C}_{ea}\|$
e
Ω_0^e

(a) Godunov-type CCFVM

(b) Nodal CCFVM

FIGURE 3.4: The two categories of cell centred Finite Volume Methods: (a) standard Godunov-type CCFVM; and (b) nodal CCFVM.

Eqs. (3.3b) to (3.3e) with a projected linear momentum $\tilde{\boldsymbol{p}}^C_f$. On the contrary, the second algorithm, named as Penalised-TOUCH (abbreviated to P-TOUCH), relies on the explicit addition of a residual based artificial dissipation to the fibre map Eq. (3.3b) and area map Eq. (3.3c) equations. Specifically, a geometrical penalisation of $\boldsymbol{\nabla}_0\boldsymbol{x}-\boldsymbol{F}$ is added into Eq. (3.3b), whereas $\frac{1}{2}\left(\boldsymbol{\nabla}_0\boldsymbol{x}\times\boldsymbol{\nabla}_0\boldsymbol{x}\right)-\boldsymbol{H}$ for Eq. (3.3c). Detailed discussion on C-TOUCH and P-TOUCH schemes satisfying involutions of the system under consideration can be found in Chapter 5.

3.3.2 Nodal CCFVM

Alternatively, the Total Lagrangian spatial discretisation of Eq. (3.1) can also be expressed in terms of the node-based flux approximation $\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ea}}$ (see Fig. 3.4b) [5, 7, 77] such that

$$\frac{d\boldsymbol{\mathcal{U}}_e}{dt} = -\frac{1}{\Omega_0^e}\sum_{a\in\Lambda_e^a}\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ea}}(\boldsymbol{\mathcal{U}}^-_{ea},\boldsymbol{\mathcal{U}}^+_{ea})\,\|\boldsymbol{C}_{ea}\|, \tag{3.5}$$

with the help of $\boldsymbol{C}_{ea} = \sum\limits_{f\in\Lambda_a^f}\frac{\boldsymbol{C}_{ef}}{\Lambda_f^a}$ [5, 7, 76–78, 108]. Here, Λ_e^a represents the set of nodes a of cell e, Λ_f^a represents the number of nodes associated with face f, $\boldsymbol{N}_{ea} := \boldsymbol{C}_{ea}/\|\boldsymbol{C}_{ea}\|$ and $\boldsymbol{C}_{ea}$ represent the material outward nodal unit normal vector and material nodal area normal vector. For completeness, the system of semi-discrete nodal updates for the enhanced $\{\boldsymbol{p},\boldsymbol{F},\boldsymbol{H},J,E\}$

X-GLACE methodology can be presented as

$$\frac{d\boldsymbol{p}_e}{dt} = \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \boldsymbol{t}_{ea}^C \|\boldsymbol{C}_{ea}\|; \tag{3.6a}$$

$$\frac{d\boldsymbol{F}_e}{dt} = \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \frac{\boldsymbol{p}_a^C}{\rho_0} \otimes \boldsymbol{C}_{ea}; \tag{3.6b}$$

$$\frac{d\boldsymbol{H}_e}{dt} = \boldsymbol{F}_e \boldsymbol{\times} \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \frac{\boldsymbol{p}_a^C}{\rho_0} \otimes \boldsymbol{C}_{ea}; \tag{3.6c}$$

$$\frac{dJ_e}{dt} = \boldsymbol{H}_e : \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \frac{\boldsymbol{p}_a^C}{\rho_0} \otimes \boldsymbol{C}_{ea}; \tag{3.6d}$$

$$\frac{dE_e}{dt} = \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \left(\frac{\boldsymbol{p}_a^C}{\rho_0} \cdot \boldsymbol{t}_{ea}^C \right) \|\boldsymbol{C}_{ea}\|; \tag{3.6e}$$

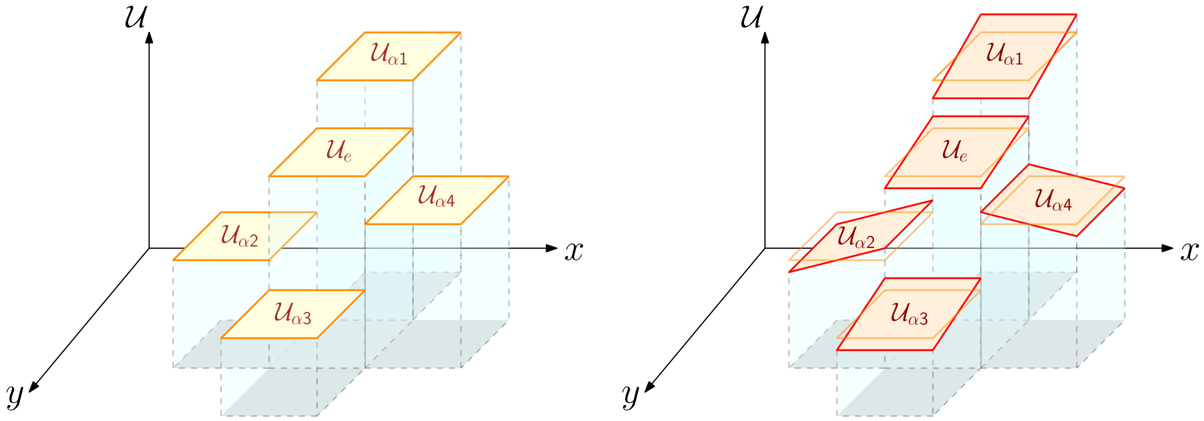
where the geometry can be obtained as

$$\frac{d\boldsymbol{x}_e}{dt} = \frac{\boldsymbol{p}_e}{\rho_0}; \qquad \frac{d\boldsymbol{x}_a}{dt} = \frac{\boldsymbol{p}_a^C}{\rho_0}. \tag{3.7}$$

3.4 Linear reconstruction procedure

It is worth noting that Eq. (3.2) or its nodal equivalent Eq. (3.5) leads to a first order solution in space as long as $\boldsymbol{\mathcal{U}}_{ef}^{-,+}$ or $\boldsymbol{\mathcal{U}}_{ea}^{-,+}$ are modelled following a piecewise constant representation within every cell (see Fig. 3.5a). This is Godonov's original approach which leads to poor accuracy and smeared results, especially in the presence of discontinuities [102, 105]. This is due to the fact that first order accurate numerical methods introduce excessive numerical dissipation/viscosity into the system in a similar manner as physical viscosity would do but to an unrealistically large extent [102]. Therefore the physics of the problem can no longer be accurately captured unless excessively fine meshes are used, which is clearly undesirable. To overcome this drawback, a reconstruction procedure is necessary such that it is locally conservative within each element [97]. In this study, reconstruction is obtained by considering linear variation of the solution within the element (polynomial of degree 1), thereby leading to a second order accurate method (see Fig. 3.5b) [11]. The evaluation of Riemann contact flux requires the computation of interfacial states on either side of the interface.

[11] Alternatively, polynomials of higher degree can also be employed to achieve higher order spatial accuracy but at the expense of computational cost [97].



(a) Constant reconstruction

(b) Linear reconstruction

FIGURE 3.5: Solution representation using (a) piecewise constant; and (b) piecewise linear elements on quadrilateral domain.

3.4.1 Gradient operator

An appropriate local gradient operator $\boldsymbol{G}_e$ can be obtained using a least squares minimisation process [6, 108]. The error $\Delta\mathcal{U}$ can be expressed as the difference between neighbouring value $\mathcal{U}_\alpha$ and the reconstructed value

$$\Delta\mathcal{U} = \mathcal{U}_\alpha - (\mathcal{U}_e + \boldsymbol{G}_e \cdot \boldsymbol{d}_{e\alpha})\,; \qquad \boldsymbol{d}_{e\alpha} = \boldsymbol{X}_\alpha - \boldsymbol{X}_e, \tag{3.8}$$

where $\boldsymbol{d}_{e\alpha}$ represents a material position vector pointing from the centroid of cell e, $\boldsymbol{X}_e$, to the position of neighbour α, $\boldsymbol{X}_\alpha$. An objective functional Π is introduced where the owner $\mathcal{U}_e$ and the gradient vector $\boldsymbol{G}_e$ are the unknowns and can be expressed as

$$\Pi(\mathcal{U}_e, \boldsymbol{G}_e) = \sum_{\alpha\in\Lambda_e^\alpha} \left[\mathcal{U}_\alpha - (\mathcal{U}_e + \boldsymbol{G}_e \cdot \boldsymbol{d}_{e\alpha})\right]^2 . \tag{3.9}$$

The directional derivative of the functional Π with respect to $\mathcal{U}_e$ and $\boldsymbol{G}_e$ can be expressed as

$$D\Pi[\delta\,\mathcal{U}_e] = \sum_{\alpha\in\Lambda_e^\alpha} -2\left[\mathcal{U}_\alpha - (\mathcal{U}_e + \boldsymbol{G}_e \cdot \boldsymbol{d}_{e\alpha})\right]\delta\,\mathcal{U}_e = \boldsymbol{0}; \tag{3.10a}$$

$$D\Pi[\delta\,\boldsymbol{G}_e] = \sum_{\alpha\in\Lambda_e^\alpha} -2\left[\mathcal{U}_\alpha - (\mathcal{U}_e + \boldsymbol{G}_e \cdot \boldsymbol{d}_{e\alpha})\right](\boldsymbol{d}_{e\alpha} \cdot \delta\boldsymbol{G}_e) = \boldsymbol{0}. \tag{3.10b}$$

For arbitrary values of $\delta\,\mathcal{U}_e$ and $\delta\,\boldsymbol{G}_e$ we get

$$\sum_{\alpha\in\Lambda_e^\alpha} \mathcal{U}_e + \sum_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha} \cdot \boldsymbol{G}_e = \sum_{\alpha\in\Lambda_e^\alpha} \mathcal{U}_\alpha; \tag{3.11a}$$

$$\mathcal{U}_e \sum_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha} + \sum_{\alpha\in\Lambda_e^\alpha} (\boldsymbol{d}_{e\alpha} \otimes \boldsymbol{d}_{e\alpha})\,\boldsymbol{G}_e = \sum_{\alpha\in\Lambda_e^\alpha} \mathcal{U}_\alpha \boldsymbol{d}_{e\alpha}. \tag{3.11b}$$

Equations (3.11) can be solved for the unknown quantities $\mathcal{U}_e$ and $\boldsymbol{G}_e$ by solving the following system of equations

$$\begin{bmatrix} n_\alpha & \left[\sum\limits_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha}\right]^T \\ \sum\limits_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha} & \sum\limits_{\alpha\in\Lambda_e^\alpha} (\boldsymbol{d}_{ea}\otimes\boldsymbol{d}_{e\alpha}) \end{bmatrix} \begin{bmatrix} \mathcal{U}_e \\ \boldsymbol{G}_e \end{bmatrix} = \begin{bmatrix} \sum\limits_{\alpha\in\Lambda_e^\alpha} \mathcal{U}_\alpha \\ \sum\limits_{\alpha\in\Lambda_e^\alpha} \mathcal{U}_\alpha\, \boldsymbol{d}_{ea} \end{bmatrix}. \tag{3.12}$$

In the case where position $\boldsymbol{X}_e$ is the centre of gravity (barycentre) of neighbouring positions $\boldsymbol{X}_\alpha$ we have

$$\boldsymbol{X}_{cg} = \boldsymbol{X}_e = \frac{1}{\alpha}\sum_{\alpha\in\Lambda_e^\alpha} \boldsymbol{X}_\alpha; \qquad \sum_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha} = \boldsymbol{0}. \tag{3.13}$$

Then, solution of the system (3.12) can be obtained as

$$\mathcal{U}_e = \frac{1}{\alpha}\sum_{\alpha\in\Lambda_e^\alpha} \mathcal{U}_\alpha; \qquad \boldsymbol{G}_e = \left[\sum_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha}\otimes\boldsymbol{d}_{e\alpha}\right]^{-1} \sum_{\alpha\in\Lambda_e^\alpha} \mathcal{U}_\alpha\, \boldsymbol{d}_{e\alpha}. \tag{3.14}$$

Remark 2: Alternatively, if $\mathcal{U}_e$ is known beforehand, the objective functional $\mathit{\Pi}$ only depends on $\boldsymbol{G}_e$ such that

$$\mathit{\Pi}(\boldsymbol{G}_e) = \sum_{\alpha\in\Lambda_e^\alpha} \left[\mathcal{U}_\alpha - (\mathcal{U}_e + \boldsymbol{G}_e\cdot\boldsymbol{d}_{e\alpha})\right]^2. \tag{3.15}$$

Minimisation of this functional results in the following expression for gradient operator

$$\boldsymbol{G}_e = \left[\sum_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha}\otimes\boldsymbol{d}_{e\alpha}\right]^{-1} \sum_{\alpha\in\Lambda_e^\alpha} (\mathcal{U}_\alpha - \mathcal{U}_e)\, \boldsymbol{d}_{e\alpha}. \tag{3.16}$$

This expression is identical to Eq. (3.14) provided that $\sum\limits_{\alpha\in\Lambda_e^\alpha} \boldsymbol{d}_{e\alpha} = \boldsymbol{0}$. This result has been reported earlier in [6, 10, 11].

Fig. 3.6 highlights the neighbours used for calculation of the gradient operator. In the case of an interior cell (see Fig. 3.6a), the immediate cell neighbours $\{\alpha_1 \dots \alpha_4\}$ are sufficient enough to approximate the gradient in cell e [62]. However, for boundary cells, this approach will lead to inaccurate gradient calculation due to truncated neighbours $\{\alpha_1 \dots \alpha_3\}$ of cell e (see Fig. 3.6b). For cells attached to the fixed boundary, this can however be avoided, for instance, during the gradient calculation of the linear momentum field where on the boundary $\boldsymbol{p}_B = \boldsymbol{0}$, by taking into account the fixed boundary information $\{\alpha_4 \dots \alpha_6\}$. This enhancement in computing gradient has shown a significant deal of improvement as shown in the numerical examples. Finally, with the aid of Equations (3.14) or (3.16), the reconstructed solution $\mathcal{U}_{e\beta}$ at the flux integration

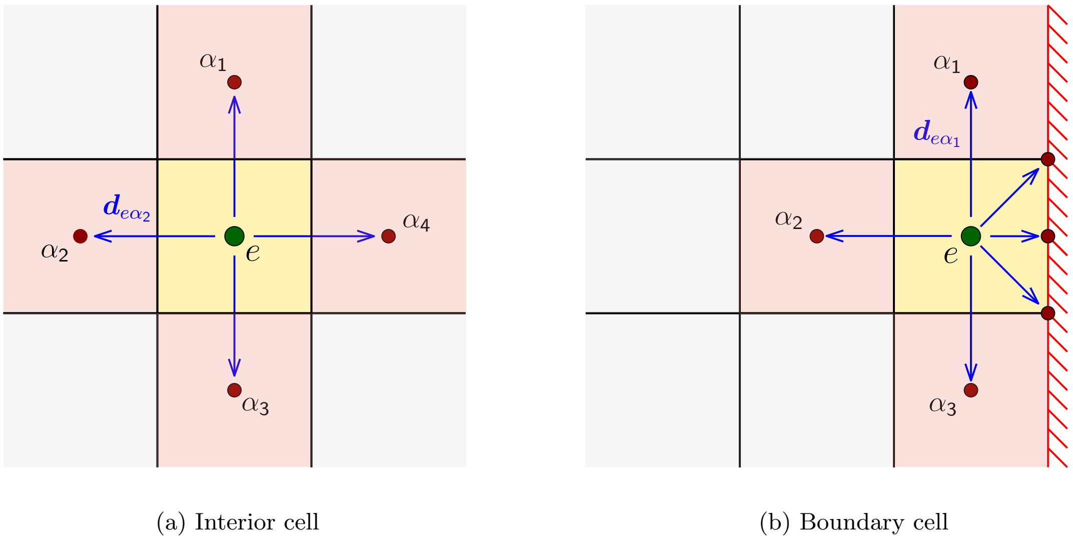


FIGURE 3.6: Stencil associated with (a) interior cell; and (b) boundary cell for gradient calculation on a two dimensional domain.

point β associated to cell e and can be written as [12]

$$\mathcal{U}_{e\beta} = \mathcal{U}_e + \boldsymbol{G}_e \cdot (\boldsymbol{X}_\beta - \boldsymbol{X}_e). \tag{3.17}$$

It is also worth pointing out that the reconstruction procedure is conservative if and only if the gradient is obtained at the cell centroid location $\boldsymbol{X}_e$.

3.4.2 Slope limiter

Although, the computation of gradient in Section 3.4.1 ensures second order accuracy in space, the solution reconstruction at flux integration points through Eq. (3.17) exhibits overshoots and undershoots. Therefore the use of a second order (or higher) spatial discretisation requires the use of so-called slope limiters to prevent the generation of oscillations and spurious modes in the regions of large gradients (eg. shocks) [21, 103, 105, 106]. This is necessary to ensure a monotonicity preserving numerical scheme [13]. For this reason, the inclusion of a local slope limiter ϕ_e is necessary to avoid the creation of a new local extrema at flux integration point β [109]

$$\mathcal{U}_{e\beta} = \mathcal{U}_e + \phi_e \boldsymbol{G}_e \cdot (\boldsymbol{X}_\beta - \boldsymbol{X}_e). \tag{3.18}$$

Algorithm 3.1 describes the conventional Barth and Jespersen limiter [104] used in this study for the satisfaction of monotonicity.

[12] Note that for a Godunov-type CCFVM, β denotes the face center location f, whereas in the case of a nodal CCFVM, it denotes the nodal location a.

[13] For a scheme to preserve monotonicity (a) a local extrema must not be created during the time evolution; and (b) an existing local minimum must be non-decreasing and a local maximum non-increasing. A detailed discussion on monotonicity of numerical schemes can be found in [21].

Algorithm 3.1: Barth and Jespersen slope limiter

```
Input  : $\mathcal{U}_e$, $\mathcal{U}_\alpha$
Output: $\phi_e$

(1) Compute minimum and maximum values:
    $\mathcal{U}_e^{\min} = \min_{\alpha \in \Lambda_e^\alpha} (\mathcal{U}_e, \mathcal{U}_\alpha)$;    $\mathcal{U}_e^{\max} = \max_{\alpha \in \Lambda_e^\alpha} (\mathcal{U}_e, \mathcal{U}_\alpha)$.

(2) Compute an unlimited reconstruction at the flux integration point $\beta$:
    $\mathcal{U}_{e\beta} = \mathcal{U}_e + \boldsymbol{G}_e \cdot (\boldsymbol{X}_\beta - \boldsymbol{X}_e)$;    $\forall \beta \in \Lambda_e^\beta$.

(3) Obtain a maximum allowable value of $\phi_{e\beta}$ at each flux integration point:
    $\phi_{e\beta} = \begin{cases} \min\left(1, \frac{\mathcal{U}_e^{\max} - \mathcal{U}_e}{\mathcal{U}_{e\beta} - \mathcal{U}_e}\right), & \text{if} \quad \mathcal{U}_{e\beta} - \mathcal{U}_e > 0; \\ \min\left(1, \frac{\mathcal{U}_e^{\min} - \mathcal{U}_e}{\mathcal{U}_{e\beta} - \mathcal{U}_e}\right), & \text{if} \quad \mathcal{U}_{e\beta} - \mathcal{U}_e < 0; \\ 1, & \text{if} \quad \mathcal{U}_{e\beta} - \mathcal{U}_e = 0. \end{cases}$

(4) Select the limiter associated to the cell:
    $\phi_e = \min_{\beta \in \Lambda_e^\beta} (\phi_{e\beta})$.
```

Chapter 4

RIEMANN SOLVER

4.1 Preliminaries

In Chapter 3, the computational domain was spatially discretised using a Finite Volume Method where the only unknown quantities were the numerical fluxes. Due to the cell centred nature of the scheme, discontinuities are present at the cell interface which motivate the use of a Riemann solver [102, 103, 110]. In Section 4.2, a contact scenario is presented along with an upwind Riemann solver through the introduction of an upwind bias into the evaluation of numerical fluxes. These fluxes are presented in terms of the Godunov-type Riemann solver (Section 4.3) and the nodal Riemann solver (Section 4.4). The former Riemann solver is further explored to handle contact mechanic problems in Section 4.5. Moreover, the performance of this acoustic Riemann solver is further improved for nearly incompressible materials by introducing preconditioned dissipation in the acoustic Riemann solver (Section 4.6). The layout of this chapter is summarised in Fig. 4.1.

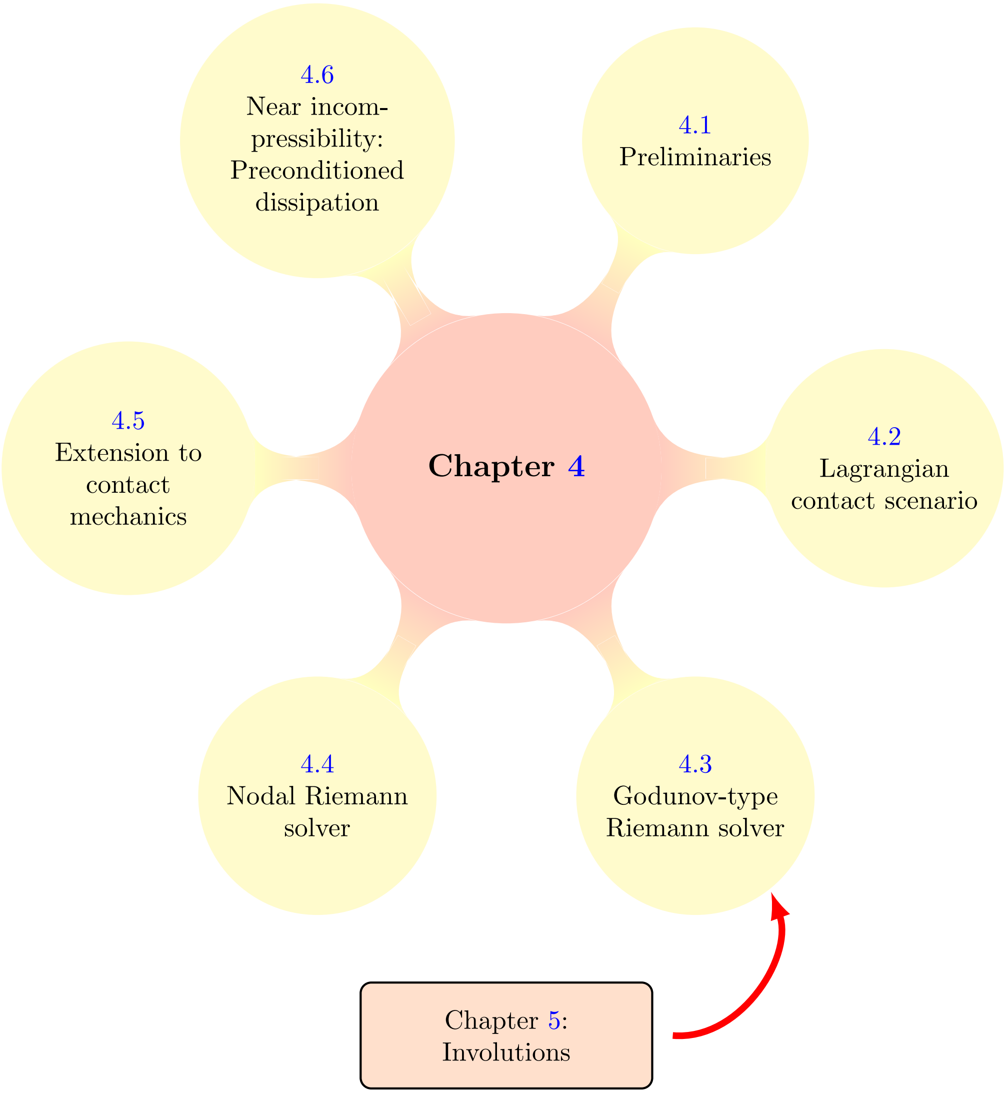

4.6
Near incompressibility:
Preconditioned dissipation
4.1
Preliminaries
4.5
Extension to contact mechanics
Chapter 4
4.2
Lagrangian contact scenario
4.4
Nodal Riemann solver
4.3
Godunov-type Riemann solver
Chapter 5:
Involutions

Figure 4.1: Structure of Chapter 4

4.2 Lagrangian contact scenario

In Lagrangian dynamics, it is often possible for two domains (i.e. Ω_0^+ and Ω_0^-) to come into contact with each other after time t. The impact would typically generate two types of shock waves (a) pressure or longitudinal wave travelling with speed c_p and (b) shear or transverse wave travelling with speed c_s within the domains, as shown in Fig. 4.2. Similarly, this scenario can also be applied to two parts of the same body in contact with each other but having a discontinuity between them. Numerically, contacts may arise due to the use of a discontinuous representation of problem variables at the flux integration point such as in Godunov-type or discontinuous Galerkin frameworks [97, 103].

The upwind finite volume spatial discretisation requires an approximate Riemann solver [97] to numerically compute contact fluxes $\boldsymbol{t}_f^C$ and $\boldsymbol{p}_f^C$ in Eqs. (3.3a) and (3.3b). There could be several ways to calculate these fluxes. In the classical approach, interface flux across a surface defined by the outward material unit normal vector $\boldsymbol{N}_{ef}$, can be expressed as [103, 105]

$$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ef}} = \underbrace{\frac{1}{2}\left[\boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}_f^-) + \boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}_f^+)\right]}_{\text{Unstable flux}} \underbrace{- \frac{1}{2}\int_{\boldsymbol{\mathcal{U}}_f^-}^{\boldsymbol{\mathcal{U}}_f^+} \left|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ef}}\right| \,\mathrm{d}\boldsymbol{\mathcal{U}},}_{\text{Upwinding stabilisation}} \tag{4.1}$$

where the above integral is taken along an arbitrary path from $\boldsymbol{\mathcal{U}}_f^-$ to $\boldsymbol{\mathcal{U}}_f^+$. The first term denotes the unstable flux (simple arithmetic average of the left and right states), implying no consideration for wave directional character. The second (upwinding stabilisation) term can be interpreted as a numerical diffusion that damps the instabilities arising from the first term. Eq. (4.1) could be simplified if an acoustic Riemann solver is employed where the flux Jacobian matrix is considered constant across the interface such that [14]

$$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ef}} = \underbrace{\frac{1}{2}\left[\boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}_f^-) + \boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}_f^+)\right]}_{\text{Unstable flux}} \underbrace{-\frac{1}{2}\left|\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ef}}\right|\left(\boldsymbol{\mathcal{U}}_f^+ - \boldsymbol{\mathcal{U}}_f^-\right)}_{\text{Upwinding stabilisation}}. \tag{4.2}$$

This approach has been presented earlier in [8]. Therefore, in this study an alternative strategy is employed for flux evaluation and is explained in the following section.

[14] In reality, the wave speeds c_p and c_s are a non-linear function of the jump in conservation variables.

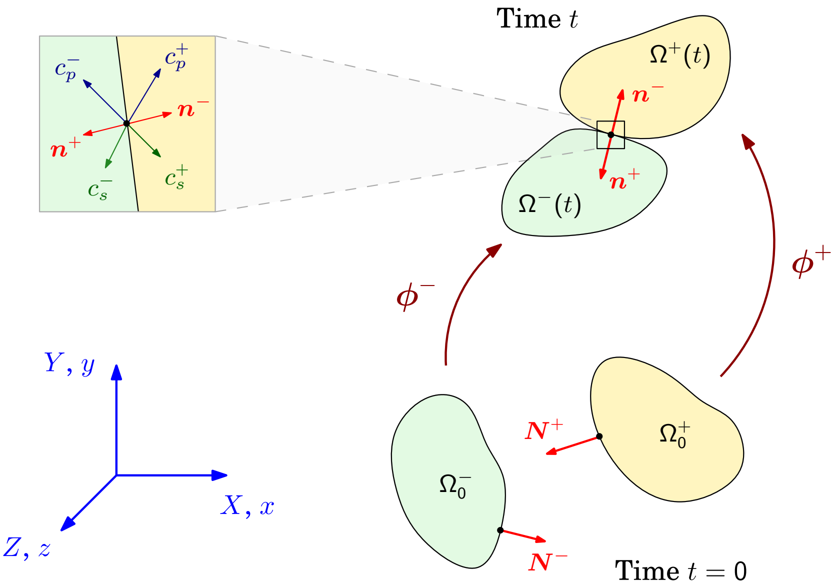


FIGURE 4.2: Contact scenario

4.3 Godunov-type Riemann solver

The acoustic Riemann solver can also be derived based on the Rankine-Hugonoit jump condition for the conservation of linear momentum (see Eq. (2.25a)) as shown earlier in [6]. This jump can be split into its normal and tangential components (see Fig. 4.3) such that

$$c_p\,(\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\,[\![\boldsymbol{p}]\!] = -(\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\,[\![\boldsymbol{t}]\!], \tag{4.3a}$$

$$c_s\,(\boldsymbol{I}-\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\,[\![\boldsymbol{p}]\!] = -(\boldsymbol{I}-\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\,[\![\boldsymbol{t}]\!], \tag{4.3b}$$

where $(\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})$ and $(\boldsymbol{I}-\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})$ are the projection tensors used to express the normal and tangential components respectively. Here, $\boldsymbol{n}$ is the spatial outward unit normal vector (push forward of the material outward unit normal $\boldsymbol{N}$), which is given by

$$\boldsymbol{n}_{ef} = \frac{\boldsymbol{F}_f^{-T}\boldsymbol{N}_{ef}}{\|\boldsymbol{F}_f^{-T}\boldsymbol{N}_{ef}\|}, \tag{4.4}$$

where $\boldsymbol{F}_f$ is the deformation gradient tensor at the face obtained via linear interpolation from the cell values. Considering Eq. (4.3a) can be decomposed into

$$c_p^-\,(\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\left(\boldsymbol{p}_{ef}^- - \boldsymbol{p}_f^C\right) = (\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\left(\boldsymbol{t}_{ef}^- - \boldsymbol{t}_f^C\right), \tag{4.5a}$$

$$c_p^+\,(\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\left(\boldsymbol{p}_{ef}^+ - \boldsymbol{p}_f^C\right) = -(\boldsymbol{n}_{ef}\otimes\boldsymbol{n}_{ef})\left(\boldsymbol{t}_{ef}^+ - \boldsymbol{t}_f^C\right). \tag{4.5b}$$

Solving Eqs. (4.5a) and (4.5b) for the normal components of contact linear momentum $\boldsymbol{p}^C_{f,n}$ and contact traction $\boldsymbol{t}^C_{f,n}$ results in

$$\boldsymbol{p}^C_{f,n} = (\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_p^- \boldsymbol{p}^-_{ef} + c_p^+ \boldsymbol{p}^+_{ef}}{c_p^- + c_p^+} + \frac{\boldsymbol{t}^+_{ef} - \boldsymbol{t}^-_{ef}}{c_p^- + c_p^+} \right], \tag{4.6a}$$

$$\boldsymbol{t}^C_{f,n} = (\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_p^+ \boldsymbol{t}^-_{ef} + c_p^- \boldsymbol{t}^+_{ef}}{c_p^- + c_p^+} + \frac{c_p^- c_p^+ (\boldsymbol{p}^+_{ef} - \boldsymbol{p}^-_{ef})}{c_p^- + c_p^+} \right]. \tag{4.6b}$$

Analogously, the tangential components of contact linear momentum $\boldsymbol{p}^C_{f,t}$ and contact traction $\boldsymbol{t}^C_{f,t}$ can be expressed as

$$\boldsymbol{p}^C_{f,t} = (\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_s^- \boldsymbol{p}^-_{ef} + c_s^+ \boldsymbol{p}^+_{ef}}{c_s^- + c_s^+} + \frac{\boldsymbol{t}^+_{ef} - \boldsymbol{t}^-_{ef}}{c_s^- + c_s^+} \right], \tag{4.7a}$$

$$\boldsymbol{t}^C_{f,t} = (\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_s^+ \boldsymbol{t}^-_{ef} + c_s^- \boldsymbol{t}^+_{ef}}{c_s^- + c_s^+} + \frac{c_s^- c_s^+ (\boldsymbol{p}^+_{ef} - \boldsymbol{p}^-_{ef})}{c_s^- + c_s^+} \right]. \tag{4.7b}$$

Combining Eqs. (4.6a) and (4.7a) results in the expression for linear momentum flux

$$\begin{aligned} \boldsymbol{p}^C_{f} = {} & (\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_p^- \boldsymbol{p}^-_{ef} + c_p^+ \boldsymbol{p}^+_{ef}}{c_p^- + c_p^+} + \frac{\boldsymbol{t}^+_{ef} - \boldsymbol{t}^-_{ef}}{c_p^- + c_p^+} \right] \\ & + (\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_s^- \boldsymbol{p}^-_{ef} + c_s^+ \boldsymbol{p}^+_{ef}}{c_s^- + c_s^+} + \frac{\boldsymbol{t}^+_{ef} - \boldsymbol{t}^-_{ef}}{c_s^- + c_s^+} \right], \end{aligned} \tag{4.8}$$

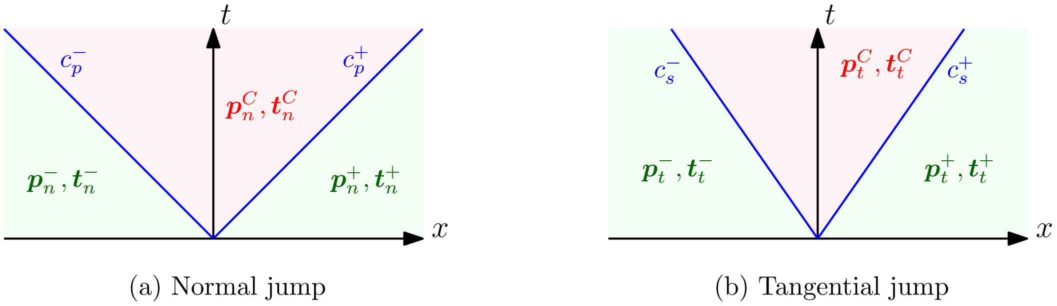
and similarly Eqs. (4.6b) and (4.7b) lead to the computation of traction flux

$$\begin{aligned} \boldsymbol{t}^C_{f} = {} & (\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_p^+ \boldsymbol{t}^-_{ef} + c_p^- \boldsymbol{t}^+_{ef}}{c_p^- + c_p^+} + \frac{c_p^- c_p^+ (\boldsymbol{p}^+_{ef} - \boldsymbol{p}^-_{ef})}{c_p^- + c_p^+} \right] \\ & + (\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\frac{c_s^+ \boldsymbol{t}^-_{ef} + c_s^- \boldsymbol{t}^+_{ef}}{c_s^- + c_s^+} + \frac{c_s^- c_s^+ (\boldsymbol{p}^+_{ef} - \boldsymbol{p}^-_{ef})}{c_s^- + c_s^+} \right]. \end{aligned} \tag{4.9}$$

In the case of homogeneous linear elasticity ($c_p = c_p^- = c_p^+$ and $c_s = c_s^- = c_s^+$) Eqs. (4.8) and (4.9) can be simplified to

$$\boldsymbol{p}^C_f = \frac{1}{2}(\boldsymbol{p}^-_{ef} + \boldsymbol{p}^+_{ef}) + \frac{1}{2}\boldsymbol{S}^{\boldsymbol{t}}_{ef}\,(\boldsymbol{t}^+_{ef} - \boldsymbol{t}^-_{ef}); \tag{4.10}$$

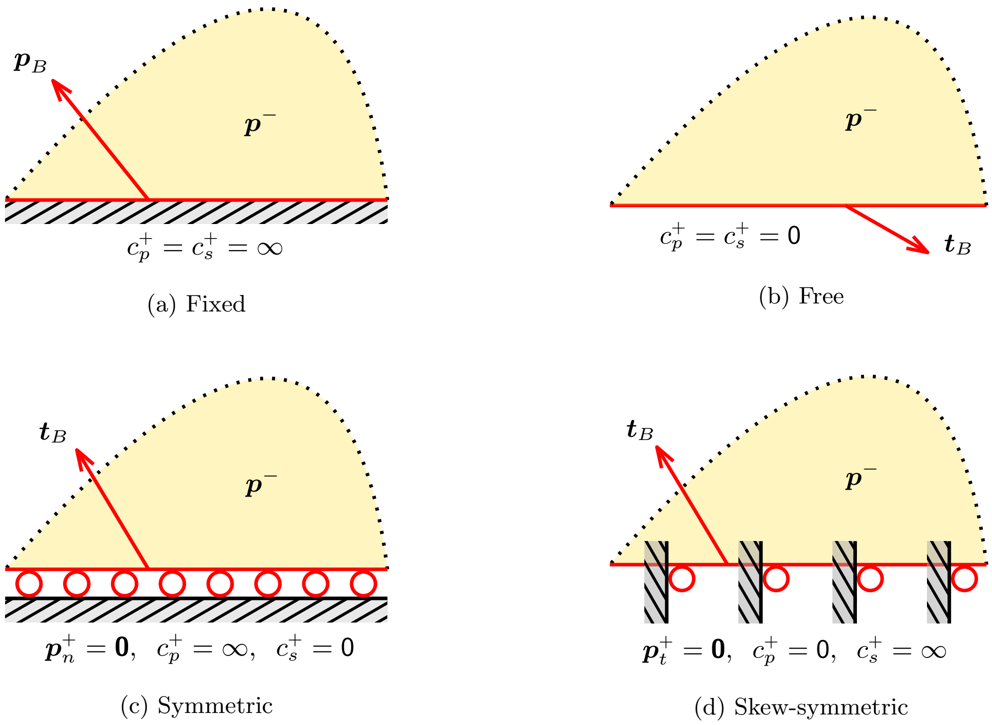
$$\boldsymbol{t}^C_f = \frac{1}{2}(\boldsymbol{t}^-_{ef} + \boldsymbol{t}^+_{ef}) + \frac{1}{2}\boldsymbol{S}^{\boldsymbol{p}}_{ef}\,(\boldsymbol{p}^+_{ef} - \boldsymbol{p}^-_{ef}), \tag{4.11}$$



(a) Normal jump

(b) Tangential jump

FIGURE 4.3: Representation of (a) normal; and (b) tangential jumps in contact fluxes.



(a) Fixed

(b) Free

(c) Symmetric

(d) Skew-symmetric

FIGURE 4.4: Schematic depicting boundary conditions for (a) fixed; (b) free; (c) symmetric; and (d) skew-symmetric boundaries.

where the stabilisation matrices can be expressed as

$$\boldsymbol{S}^{\boldsymbol{t}}_{ef} = \frac{1}{c_p}(\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) + \frac{1}{c_s}(\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}), \tag{4.12a}$$

$$\boldsymbol{S}^{\boldsymbol{p}}_{ef} = c_p(\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) + c_s(\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}). \tag{4.12b}$$

Furthermore, boundary conditions can be derived through Eqs. (4.8) and (4.9) [15] by applying appropriate values in the outer domain.

[15] The interior domain is denoted by a negative superscript and the outer domain with a positive superscript.

4.3.1 Moving boundary

In case of a moving boundary (see Fig. 4.4a), the linear momentum in the outer domain can be expressed as

$$\boldsymbol{p}_{ef}^{+} = \boldsymbol{p}_B. \tag{4.13}$$

Since no deformation is allowed in this particular case either in the longitudinal or the transverse directions, the wave speeds in the outer domain can be written as

$$c_p^{+} = c_s^{+} \approx \infty. \tag{4.14}$$

In such a scenario we expect the following in Eqs. (4.8) and (4.9)

$$\begin{aligned} \frac{c_p^-}{c_p^- + c_p^+} = \frac{c_s^-}{c_s^- + c_s^+} \approx 0; \qquad & \frac{c_p^+}{c_p^- + c_p^+} = \frac{c_s^+}{c_s^- + c_s^+} \approx 1; \\ \frac{c_p^- c_p^+}{c_p^- + c_p^+} \approx c_p^-; \qquad & \frac{c_s^- c_s^+}{c_s^- + c_s^+} \approx c_s^-. \end{aligned} \tag{4.15}$$

By making use of equations Eqs. (4.13) to (4.15) in Eqs. (4.8) and (4.9), the interface numerical fluxes for a moving boundary can be expressed as

$$\begin{aligned} \boldsymbol{p}_f^C &= \boldsymbol{p}_B; \\ \boldsymbol{t}_f^C &= \boldsymbol{t}_{ef}^- + \boldsymbol{S}_{ef}^{\boldsymbol{p}} \left(\boldsymbol{p}_B - \boldsymbol{p}_{ef}^- \right). \end{aligned} \tag{4.16}$$

For a fixed boundary (more commonly know as a wall boundary in the CFD community), the linear momentum at the boundary is zero ($\boldsymbol{p}_B = \boldsymbol{0}$).

4.3.2 Traction boundary

When dealing with a traction boundary condition (see Fig. 4.4b) we can conclude that

$$\boldsymbol{t}_{ef}^{+} = \boldsymbol{t}_B; \qquad c_p^{+} = c_s^{+} = 0, \tag{4.17}$$

where $\boldsymbol{t}_B$ is the applied boundary traction. Eqs. (4.8) and (4.9) then reduce to

$$\begin{aligned} \boldsymbol{p}_f^C &= \boldsymbol{p}_{ef}^- + \boldsymbol{S}_{ef}^{\boldsymbol{t}} \left(\boldsymbol{t}_B - \boldsymbol{t}_{ef}^- \right); \\ \boldsymbol{t}_f^C &= \boldsymbol{t}_B. \end{aligned} \tag{4.18}$$

In the case of a free boundary, the boundary traction is considered zero ($\boldsymbol{t}_B = \boldsymbol{0}$).

4.3.3 Symmetric boundary

For a symmetric boundary (restricted normal displacements) with applied traction $\boldsymbol{t}_{ef}^{+} = \boldsymbol{t}_B$ (see Fig. 4.4c), the interior domain merely slides tangentially along the boundary. Physically this can be interpreted as a roller support with restricted normal displacements. Since deformation is not allowed in the normal direction we have

$$\boldsymbol{p}_n^+ = \boldsymbol{0}; \qquad c_p^+ \approx \infty; \qquad c_s^+ = 0. \tag{4.19}$$

By making use of the above in Eqs. (4.8) and (4.9) we obtain

$$\begin{aligned}
\boldsymbol{p}_f^C &= (\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\boldsymbol{p}_{ef}^- + \frac{1}{c_s^-}(\boldsymbol{t}_B - \boldsymbol{t}_{ef}^-)\right]; \\
\boldsymbol{t}_f^C &= (\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\boldsymbol{t}_{ef}^- - c_p^- \boldsymbol{p}_{ef}^-\right] + (\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef})\boldsymbol{t}_B.
\end{aligned} \tag{4.20}$$

4.3.4 Skew-symmetric boundary

As opposed to a symmetric boundary condition, a skew-symmetric boundary with applied traction $\boldsymbol{t}_{ef}^+ = \boldsymbol{t}_B$ (see Fig. 4.4d) only allows for normal displacements. This implies that

$$\boldsymbol{p}_t^+ = \boldsymbol{0}; \qquad c_p^+ = 0; \qquad c_s^+ \approx \infty. \tag{4.21}$$

Eqs. (4.8) and (4.9) then reduce to

$$\begin{aligned}
\boldsymbol{p}^C &= (\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\boldsymbol{p}_{ef}^- + \frac{1}{c_s^-}(\boldsymbol{t}_B - \boldsymbol{t}_{ef}^-)\right]; \\
\boldsymbol{t}^C &= (\boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef})\boldsymbol{t}_B + (\boldsymbol{I} - \boldsymbol{n}_{ef} \otimes \boldsymbol{n}_{ef}) \left[\boldsymbol{t}_{ef}^- - c_p^- \boldsymbol{p}_{ef}^-\right].
\end{aligned} \tag{4.22}$$

4.4 Nodal Riemann solver

An alternate well known approach, known as the nodal cell centred finite volume framework [7, 20, 76–78, 108] has been successfully used in the numerical analysis of gas dynamic problems. Moreover, extension this nodal solver to solid mechanics applications was presented in [5]. In particular, an Updated Lagrangian $\{\boldsymbol{p}, \boldsymbol{F}, E, \boldsymbol{x}\}$ mixed formulation was spatially discretised using a Godunov-like Lagrangian Cell centred Scheme (GLACE) for hyperelasticity. In this section, this nodal scheme is adapted from an Updated Lagrangian framework to a Total Lagrangian formalism, thus allowing an alternative approach for comparison purposes. Following the works of Després *et al.* [5] and Maire *et al.* [7], the nodal linear momentum $\boldsymbol{p}_a^C$ and elemental nodal first Piola Kirchhoff stress $\boldsymbol{P}_{ea}^C$ are computed using the linear momentum jump conditon. For consistency, as shown earlier for the Godunov-type Riemann solver in Section 4.3, the corresponding jump can be decomposed into the summation of a normal and tangential contribution defined by

$$c_p(\boldsymbol{n}_{ea} \otimes \boldsymbol{n}_{ea}) \left[\boldsymbol{p}_a^C - \boldsymbol{p}_{ea}\right] = -(\boldsymbol{n}_{ea} \otimes \boldsymbol{n}_{ea}) \left[\boldsymbol{P}_{ea}^C - \boldsymbol{P}_{ea}\right] \boldsymbol{N}_{ea}, \tag{4.23a}$$

$$c_s(\boldsymbol{I} - \boldsymbol{n}_{ea} \otimes \boldsymbol{n}_{ea}) \left[\boldsymbol{p}_a^C - \boldsymbol{p}_{ea}\right] = -(\boldsymbol{I} - \boldsymbol{n}_{ea} \otimes \boldsymbol{n}_{ea}) \left[\boldsymbol{P}_{ea}^C - \boldsymbol{P}_{ea}\right] \boldsymbol{N}_{ea}. \tag{4.23b}$$

To complete the system (4.23), it is necessary to consider an additional nodal equilibrium (kinetic) condition such that the summation of all elemental nodal forces is zero, that is

$$\sum_{e\in\Lambda_a^e} \boldsymbol{P}_{ea}^C \boldsymbol{C}_{ea} = \boldsymbol{0}. \tag{4.24}$$

One of the most critical aspects of the nodal framework is the computation of nodal quantities such as nodal area vectors $\boldsymbol{C}_{ea}$. Generally, normals are associated with planar faces and therefore can be uniquely defined. However, this is not the same as defining a unique normal at a node. Després and collaborators [5, 76] proposed a methodology to compute nodal area vectors

$$\boldsymbol{C}_{ea} = \sum_{f\in\Lambda_a^f} \frac{\boldsymbol{C}_{ef}}{\Lambda_f^a}, \tag{4.25}$$

where $\boldsymbol{C}_{ef}$ is the area normal vector of face f associated with cell e and Λ_f^a are the number of nodes associated with face f. Moreover, the spatial nodal normals used in Eqs. (4.23a) and (4.23b) can be obtained from the area map $\boldsymbol{H}$ such that

$$\boldsymbol{n}_{ea} = \frac{\boldsymbol{c}_{ea}}{\|\boldsymbol{c}_{ea}\|}; \qquad \boldsymbol{c}_{ea} = \sum_{f\in\Lambda_a^f} \frac{\boldsymbol{c}_{ef}}{\Lambda_f^a}; \qquad \boldsymbol{c}_{ef} = \boldsymbol{H}_f \boldsymbol{C}_{ef}. \tag{4.26}$$

Addition of Eq. (4.23a) and Eq. (4.23b) gives

$$\boldsymbol{\mathcal{S}}_{ea}^{\boldsymbol{p}} \boldsymbol{p}_a^C = \boldsymbol{\mathcal{S}}_{ea}^{\boldsymbol{p}} \boldsymbol{p}_{ea} + \left(\boldsymbol{P}_{ea} - \boldsymbol{P}_{ea}^C\right) \boldsymbol{N}_{ea}, \tag{4.27}$$

where $\boldsymbol{\mathcal{S}}_{ea}^{\boldsymbol{p}}$ is the dissipation tensor defined as

$$\boldsymbol{\mathcal{S}}_{ea}^{\boldsymbol{p}} = c_p(\boldsymbol{n}_{ea} \otimes \boldsymbol{n}_{ea}) + c_s(\boldsymbol{I} - \boldsymbol{n}_{ea} \otimes \boldsymbol{n}_{ea}). \tag{4.28}$$

Multiplying Eq. (4.27) by nodal areas $\|\boldsymbol{C}_{ea}\|$, summing over all elements attached to a node and utilising Eq. (4.24), we can express the nodal linear momentum $\boldsymbol{p}_a^C$ as

$$\boxed{\boldsymbol{p}_a^C = \boldsymbol{A}_a^{-1} \boldsymbol{b}_a\,,} \tag{4.29}$$

where

$$\boldsymbol{A}_a = \sum_{e\in\Lambda_a^e} \|\boldsymbol{C}_{ea}\| \boldsymbol{\mathcal{S}}_{ea}^{\boldsymbol{p}}; \qquad \boldsymbol{b}_a = \sum_{e\in\Lambda_a^e} \|\boldsymbol{C}_{ea}\| \boldsymbol{\mathcal{S}}_{ea}^{\boldsymbol{p}} \boldsymbol{p}_{ea} + \sum_{e\in\Lambda_a^e} \boldsymbol{P}_{ea}\boldsymbol{C}_{ea}. \tag{4.30}$$

It is now possible to evaluate the elemental nodal traction $\boldsymbol{t}_{ea}^C$ by combining Eq. (4.23a) and Eq. (4.23b), to give:

$$\boxed{\boldsymbol{t}_{ea}^C := \boldsymbol{P}_{ea}^C \boldsymbol{N}_{ea} = \boldsymbol{P}_{ea} \boldsymbol{N}_{ea} + \boldsymbol{\mathcal{S}}_{ea}^{\boldsymbol{p}} \left(\boldsymbol{p}_{ea} - \boldsymbol{p}_a^C\right).} \tag{4.31}$$

The evaluation of nodal fluxes $\boldsymbol{p}_a^C$ and $\boldsymbol{t}_{ea}^C$ is summarised in Algorithm 4.1. Once these quantities are obtained, the conservation variables of interest can be obtained from nodal updates Eqs. (3.6a) to (3.6) presented earlier. As discussed in [5, 7, 20, 77], these nodal updates satisfy the local entropy inequality by construction when utilising a first order spatial discretisation (piecewise constant reconstruction), which is necessary for correct computation of discontinuities.

Algorithm 4.1: Computation of fluxes using nodal Riemann solver

```
Input  : $\boldsymbol{p}_e$, $\boldsymbol{P}_e$
Output: $\boldsymbol{p}_a^C$, $\boldsymbol{t}_{ea}^C$

(1) Compute gradient of linear momentum $\boldsymbol{p}_e$:   $\boldsymbol{G}_e(\boldsymbol{p}_e) \leftarrow$ Eq. (3.16)
(2) Compute gradient of first Piola Kirchoff stress $\boldsymbol{P}_e$:   $\boldsymbol{G}_e(\boldsymbol{P}_e) \leftarrow$ Eq. (3.16)
(3) Obtain elemental nodal linear momentum:   $\boldsymbol{p}_{ea} \leftarrow$ Eq. (3.18)
(4) Obtain elemental first Piola Kirchoff stress:   $\boldsymbol{P}_{ea} \leftarrow$ Eq. (3.18)
(5) Calculate elemental nodal spatial normals:   $\boldsymbol{n}_{ea} \leftarrow$ Eq. (4.25)
(6) Evaluate nodal contact linear momentum:   $\boldsymbol{p}_a^C \leftarrow$ Eq. (4.29)
(7) Evaluate elemental nodal contact traction:   $\boldsymbol{t}_{ea}^C \leftarrow$ Eq. (4.31)
```

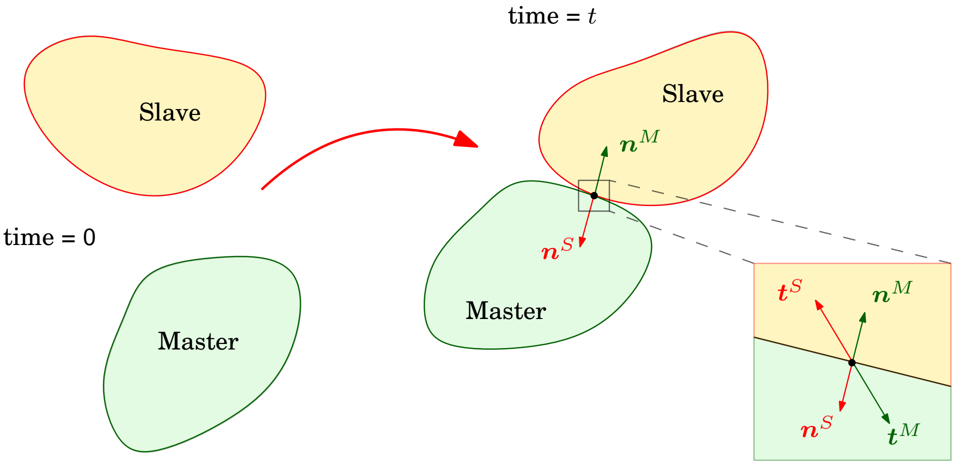


FIGURE 4.5: Contact scenario for contact applications

4.5 Extension to contact mechanics

In this section, extension of the Godunov-type Riemann solver (see Section 4.3) to contact problems is discussed. The two bodies expected to be in contact are designated as master (denoted by superscript M) and slave (denoted by a superscript S) bodies. Initially separated (see Fig. 4.5), the bodies come into contact at time t at a unique location. In the case of an ideal contact, the spatial normals at the master and slave boundaries align such that $\boldsymbol{n}^S = -\boldsymbol{n}^M$. In addition, traction of the same magnitude acts on both bodies such that it obeys the action-reaction principle $\boldsymbol{t}^S = -\boldsymbol{t}^M$. In this work, contact problems have been simplified by considering contact of one body (slave) with another rigid, planar body (master) [16].

For contact problems, it is essential to determine the separation gap d_n, also known as non-penetration, between the two bodies (see Fig. 4.6). The gap can be simply calculated as

$$d_n = (\boldsymbol{x}^S - \boldsymbol{x}^M) \cdot \boldsymbol{n}^M. \tag{4.32}$$

The gap d_n will be positive if the master and slave bodies are separated, zero in the case of an ideal contact and negative in the case of penetration which is physically non-admissible,

$$\begin{aligned} d_n > 0 &\quad\longrightarrow\quad \text{separation} \\ d_n = 0 &\quad\longrightarrow\quad \text{ideal contact} \\ d_n < 0 &\quad\longrightarrow\quad \text{penetration.} \end{aligned} \tag{4.33}$$

[16] More general self and multi-body contact will be studied in future [111, 112].

Moreover, the normal pressure experienced by both the bodies can be expressed as

$$t_n = \boldsymbol{t}^S \cdot \boldsymbol{n}^S = \boldsymbol{t}^M \cdot \boldsymbol{n}^M. \tag{4.34}$$

According to the sign convention used in Eq 4.34, t_n will be negative during compression and positive otherwise. Finally, the classical set of Kuhn-Tucker-Karush (KKT) relations [111, 112], in the absence of cohesion, can be expressed as

$$d_n \geq 0 \qquad \text{(kinematic constraint);} \tag{4.35}$$

$$t_n \leq 0 \qquad \text{(kinetic constraint);} \tag{4.36}$$

$$t_n\, d_n = 0 \qquad \text{(complementary condition).} \tag{4.37}$$

With the definition of KKT conditions, all machinery is now available for implementation of the contact algorithm which can be described in two steps: (a) contact detection followed by (b) contact boundary conditions.

4.5.1 Contact detection

Effective contact detection is one the most integral aspects of any contact algorithm. For many simple contact problems, part of the slave body which is expected to be in contact with the rigid planar master body can be predicted beforehand. Therefore, only a section of the slave body is designated as the slave boundary where a dynamic switch of boundary conditions is applied. This is necessary to reduce the computation time spent in contact detection.

In order to detect contact, it is necessary to decide which points on the slave boundary will be considered. The obvious choice that comes to mind is to consider the face centers on the slave boundaries $\boldsymbol{x}_f^S$. We can further enhance contact detection by using more Gauss points at the face (with spatial coordinates $\boldsymbol{x}_g^S$) rather than just the face center. The location of the Gauss points can easily be obtained by using the finite element shape functions as discussed in Eq. (B.3). Now, the gap at each slave boundary face Gauss point $\boldsymbol{x}_g^S$ can be obtained as

$$d_n^g = (\boldsymbol{x}_g^S - \boldsymbol{x}_f^M) \cdot \boldsymbol{n}_{ef}^M. \tag{4.38}$$

4.5.2 Contact boundary conditions

Once the separation gap has been computed, it is possible to apply the corresponding boundary condition according to Eq. (4.33). Initially, when both bodies are separated, free boundary condition needs to be applied on the slave boundary Gauss points for the contact linear momentum

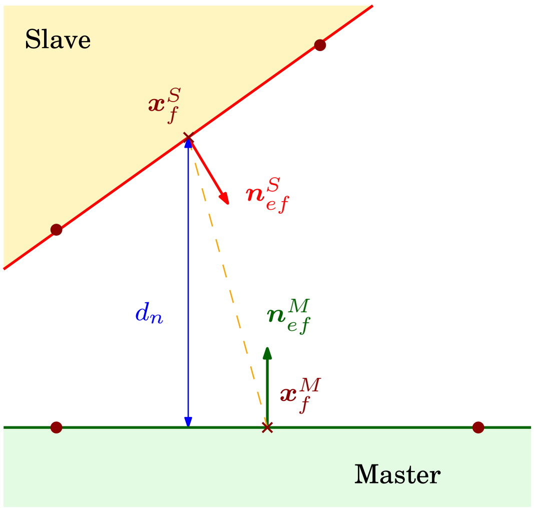


FIGURE 4.6: Contact detection

$[\boldsymbol{p}_g^C]^S$ and contact traction $[\boldsymbol{t}_g^C]^S$ such that

$$\left[\boldsymbol{p}_g^C\right]^S = \boldsymbol{p}_{eg}^S + \left[\frac{1}{c_p}\left(\boldsymbol{n}_{eg}^S \otimes \boldsymbol{n}_{eg}^S\right) + \frac{1}{c_s}\left(\boldsymbol{I} - \boldsymbol{n}_{eg}^S \otimes \boldsymbol{n}_{eg}^S\right)\right] \boldsymbol{t}_{eg}^S; \tag{4.39}$$

$$\left[\boldsymbol{t}_g^C\right]^S = \boldsymbol{0}, \tag{4.40}$$

where $\boldsymbol{p}_{eg}^S$ and $\boldsymbol{t}_{eg}^S$ are the reconstructed linear momentum and traction at the Gauss point under consideration [17]. In the event of contact detection at a particular Gauss point associated to the slave boundary face (i.e. $d_n^g \leq 0$), the slave boundary is switched from free to frictionless roller supports (symmetric boundary condition). In this case it is important to point out that the gradient of linear momentum used in the reconstruction process are obtained through a predictor-corrector step as detailed in Algorithm 4.2. The frictionless contact boundary conditions for linear momentum $[\boldsymbol{p}_g^C]^S$ and traction $[\boldsymbol{t}_g^C]^S$ can be expressed as

$$\left[\boldsymbol{p}_g^C\right]^S = \left(\boldsymbol{I} - \boldsymbol{n}_{eg}^S \otimes \boldsymbol{n}_{eg}^S\right)\left[\boldsymbol{p}_{eg}^S - \frac{1}{c_p}\boldsymbol{t}_{eg}^S\right]; \tag{4.41}$$

$$\left[\boldsymbol{t}_g^C\right]^S = \left(\boldsymbol{n}_{eg}^S \otimes \boldsymbol{n}_{eg}^S\right)\left[\boldsymbol{t}_{eg}^S - c_p\,\boldsymbol{p}_{eg}^S\right]. \tag{4.42}$$

Once these fluxes are obtained at the Gauss points they can easily be transferred to the face centre as

$$[\boldsymbol{p}_f^C]^S = \sum_{g \in \Lambda_f^g} [\boldsymbol{p}_g^C]^S\, w_g; \qquad [\boldsymbol{t}_f^C]^S = \sum_{g \in \Lambda_f^g} [\boldsymbol{t}_g^C]^S\, w_g, \tag{4.43}$$

where w_g are the weights associated to Gauss point g. Additionally, when contact occurs strong boundary conditions for the slave boundary linear momentum $\boldsymbol{p}_a^S$ must be enforced such that

[17] The procedure to obtain these linearly reconstructed states is the same as outlined in Section 3.4.

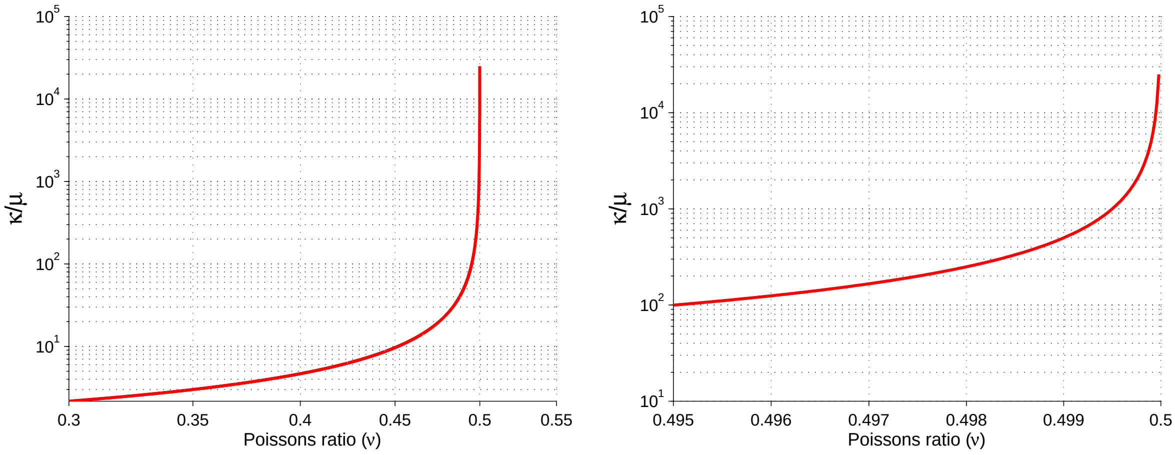


FIGURE 4.7: Variation of κ/μ with Poisson's ratio ν.

it is kept orthogonal to the normal of the rigid master body $\boldsymbol{n}^M$ such that [18].

$$\boldsymbol{p}_a^S = \left(\boldsymbol{I} - \boldsymbol{n}^M \otimes \boldsymbol{n}^M\right) \boldsymbol{p}_a^S. \tag{4.44}$$

We are also aware that when the slave body is in contact, it experiences a compressive force where contact pressure t_n is negative as discussed earlier. Once in contact, it is imperative to monitor the sign of t_n which allows for the appropriate instance for release of the slave body. This is achieved by applying the next switch in boundary conditions (i.e. contact to free) when contact pressure is positive $t_n \geq 0$. The complete contact algorithm is summarised in Algorithm 4.2.

4.6 Near incompressibility: Preconditioned dissipation

Unfortunately, the performance of the acoustic Riemann solver presented in Sections 4.3 to 4.5 seems to be inefficient when simulating materials near the incompressibility regime ($\kappa/\mu \geq 500$). This is due to the fact that the ratio κ/μ starts shooting up exponentially with the increase in Poisson's ratio $\nu \geq 0.495$ [19] (see Fig. 4.7). Since this ratio is directly related to the pressure wave speed c_p we see a ten-fold increase in c_p at $\nu = 0.4999$ when compared to c_p at $\nu = 0.49$ (see Table 4.1). This causes the Riemann solver to become unstable due to inappropriate introduction of numerical dissipation.

In order to extend the range of applications near the incompressibility limit, upwinding stabilisation of the aforementioned flux approximation $\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ef}}$ in Eq. (4.2) must be modified by

[18] Detailed discussion on the computation of nodal linear momentum $\boldsymbol{p}_a$ is discussed in Section 5.2.

[19] A rubber-like material with density $\rho_0 = 1100$ kg/m^3 and Young's Modulus $E = 17$ MPa has been used to obtain these results.

Algorithm 4.2: Contact algorithm

```
Input  : $\boldsymbol{x}_a^S$, $\boldsymbol{p}_e$, $\boldsymbol{P}_e$
Output: $[\boldsymbol{p}_f^C]^S$, $[\boldsymbol{t}_f^C]^S$

(1) Predictor step:
    for ( $g = 0;\ g < \Lambda_f^g;\ g++$ ) do
        (1.1) Obtain spatial Gauss point location:  $\boldsymbol{x}_g^S \leftarrow$ Eq. (B.3)
        (1.2) Calculate gap at Gauss point:  $d_n^g \leftarrow$ Eq. (4.38)
        (1.3) Compute spatial normal at Gauss point:  $\boldsymbol{n}_{eg}^S \leftarrow$ Eq. (B.2)
        (1.4) Reconstruction at Gauss point:  $\boldsymbol{t}_{eg}^S$, $\boldsymbol{p}_{eg}^S \leftarrow$ Eq. (3.18)
        (1.5) Apply boundary conditions on slave boundary
              if ( $d_n \geq 0$ ) then
                  (1.5.1) Gap detected:
                          Apply free boundary conditions:  $[\boldsymbol{p}_g^C]^S$, $[\boldsymbol{t}_g^C]^S \leftarrow$ Eq. (4.39)
              else
                  (1.5.2) Contact detected:
                          Apply frictionless contact conditions:  $[\boldsymbol{p}_g^C]^S$, $[\boldsymbol{t}_g^C]^S \leftarrow$ Eq. (4.41)
                  (1.5.3) Check for rebound.
                          if ( $t_n \geq 0$ ) then
                              Rebound imminent:
                              Switch to free boundary conditions:  $[\boldsymbol{p}_g^C]^S$, $[\boldsymbol{t}_g^C]^S \leftarrow$ Eq. (4.39)
                          else
                              Contact persists:
                              Impose strong boundary conditions:  $\boldsymbol{p}_a^S \leftarrow$ Eq. (4.44)
                          end
              end
    end

(2) Compute minimum separation distance at the face:  $[d_n^f]_{\min} = \min_{g \in \Lambda_f^g} (\, d_n^g \,)$.

(3) Corrector step:
    if ( $[d_n^f]_{\min} \leq 0$ ) then
        for  ( $g = 0;\ g < \Lambda_f^g;\ g++$ ) do
            (3.1) Recompute cell gradients using boundary linear momentum (see Fig. 3.6b)
                  $\boldsymbol{G}_e\,(\boldsymbol{p}) \leftarrow$ Fig. 3.6b
            (3.2) Repeat Steps 1.4 and 1.5.
        end
    end

(4) Obtain fluxes at slave contact face:  $[\boldsymbol{p}_f^C]^S$, $[\boldsymbol{t}_f^C]^S \leftarrow$ Eq. (4.43)
```

Poisson's ratio	**Pressure wave (c_p)**	**Shear wave (c_s)**	**Ratio (c_p/c_s)**
0.49	514.28	72.0145	7.14
0.499	1607.06	71.7980	22.38
0.4999	5075.87	71.7764	70.72
0.49999	16049.38	71.7742	223.61
0.499999	50571.98	71.7741	704.60
0.4999999	160491.69	71.7741	2236.07

Table 4.1: Variation of pressure and shear wave speeds along with Poisson's ratio.

preconditioning [113–115], resulting in

$$\boldsymbol{\mathcal{F}}^C_{\boldsymbol{N}_{ef}} = \frac{1}{2}\left[\boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}_f^-) + \boldsymbol{\mathcal{F}}_{\boldsymbol{N}_{ef}}(\boldsymbol{\mathcal{U}}_f^+)\right] - \frac{1}{2}\boldsymbol{\mathcal{P}}^{-1}|\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{A}}_{\boldsymbol{N}_{ef}}|\left(\boldsymbol{\mathcal{U}}_f^+ - \boldsymbol{\mathcal{U}}_f^-\right), \tag{4.45}$$

where $\boldsymbol{\mathcal{P}}$ is the preconditioning matrix. With respect to the original flux computation Eq. (4.2), only the upwinding stabilisation terms (also known as high order corrections) are altered and thus, the finite volume spatial discretisation presented in Chapter 3 remains a consistent approximation for the enhanced $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ system. In this study, we use a simple (diagonal) preconditioner defined as

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \beta^2\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \beta^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{1}{\beta^2} \end{bmatrix}, \tag{4.46}$$

where the dimensionless parameter β is defined as $\beta := \frac{\tilde{\kappa}}{\kappa}$ and $\tilde{\kappa}$ is a user defined material constant usually taken in the neighbourhood of the bulk modulus κ of the material. The aim of this diagonal preconditioner Eq. (4.46) is to re-scale the stabilisation coefficients of the system (i.e. pressure wave speed c_p and shear wave speed c_s), but not the characteristic structure (i.e. streamline direction) of the upwinding method. Referring to Section 2.7, three pairs of non-zero eigenvalues corresponding to the volumetric and shear waves are then obtained as

$$c_{1,2} = \pm\tilde{c}_p, \quad \tilde{c}_p = \frac{1}{\beta}\sqrt{\frac{\kappa}{\rho_0}}; \qquad c_{3,4} = c_{5,6} = \pm\tilde{c}_s, \quad \tilde{c}_s = \beta\sqrt{\frac{\mu}{\rho_0}}. \tag{4.47}$$

In near incompressibility (i.e. $\kappa/\mu \geq 100$), the stabilisation matrices, as shown in Eq. (4.12a), need to be rescaled by replacing $\{c_p, c_s\}$ in Eq. (4.56) with $\{\tilde{c}_p, \tilde{c}_s\}$ (4.47) in either Eqs. (4.10) and (4.11) or Eqs. (4.29) and (4.31). Notice that the the original elastic pressure wave speed c_p and shear wave speed c_s can be simply recovered by equating the value of $\tilde{\kappa}$ with the bulk modulus κ of the material.

For completeness, the study of eigenvalue structure of the mixed system (2.22) is presented to demonstrate the hyperbolicity of the system of conservation laws. In addition, the computation of the maximum (pressure) wave speed is necessary for the evaluation of the maximum time step of the explicit time integrator. Consider the plane wave solutions of the form

$$\boldsymbol{\mathcal{U}} = \phi\left(\boldsymbol{X}\cdot\boldsymbol{N} - c_\alpha t\right)\boldsymbol{\mathcal{R}}_\alpha = \phi\left(\boldsymbol{X}\cdot\boldsymbol{N} - c_\alpha t\right)\begin{bmatrix} \boldsymbol{p}_\alpha^R \\ \boldsymbol{F}_\alpha^R \\ \boldsymbol{H}_\alpha^R \\ J_\alpha^R \end{bmatrix}, \tag{4.48}$$

where c_α are the wave speeds associated with the corresponding right eigenvectors $\boldsymbol{\mathcal{R}}_\alpha$ and the normalised direction of propagation $\boldsymbol{N}$. A compressible Mooney Rivlin constitutive model given by Eq. (2.30) is considered for simplicity. With the aid of Eq. (4.46) and Eq. (2.30), the modified eigen-problem by considering each individual component of this system becomes

$$c_\alpha \boldsymbol{p}_\alpha = -2\alpha \boldsymbol{F}_\alpha \boldsymbol{N} - \left(2\gamma \boldsymbol{F} \times \boldsymbol{H}_\alpha\right)\boldsymbol{N} - W_{JJ} J_\alpha \boldsymbol{H}\boldsymbol{N}; \tag{4.49a}$$

$$c_\alpha \boldsymbol{F}_\alpha = -\frac{\beta^2}{\rho_0}\left(\boldsymbol{p}_\alpha \otimes \boldsymbol{N}\right); \tag{4.49b}$$

$$c_\alpha \boldsymbol{H}_\alpha = -\frac{\beta^2}{\rho_0}\left[\boldsymbol{F} \times \left(\boldsymbol{p}_\alpha \otimes \boldsymbol{N}\right)\right]; \tag{4.49c}$$

$$c_\alpha J_\alpha = -\frac{1}{\beta^2 \rho_0}\boldsymbol{p}_\alpha \cdot \boldsymbol{H}\boldsymbol{N}, \tag{4.49d}$$

with $W_{JJ} = \lambda + \frac{2\alpha}{J^2}$. As a consequence of the high level of redundancy in the system of equations being considered, only six wave speeds are different from zero. These can be readily identified by substituting the last three geometric strain equations (4.49b-4.49d) into (4.49a) to give:

$$\left(2\alpha\beta^2 \boldsymbol{p}_\alpha + \frac{W_{JJ}}{\beta^2}(\boldsymbol{H}\boldsymbol{N} \otimes \boldsymbol{H}\boldsymbol{N})\boldsymbol{p}_\alpha + 2\gamma\beta^2\left[\boldsymbol{F} \times \left(\boldsymbol{F} \times \left(\boldsymbol{p}_\alpha \otimes \boldsymbol{N}\right)\right)\right]\boldsymbol{N}\right) = \rho_0 c_\alpha^2 \boldsymbol{p}_\alpha. \tag{4.50}$$

The double tensor cross product term can be simplified by repeated use of the third order alternating tensor product expression $\mathcal{E}_{ijk}\mathcal{E}_{lmk} = \delta_{il}\delta_{jm} - \delta_{im}\delta_{jl}$ and noting that

$$\boldsymbol{I} - \boldsymbol{N} \otimes \boldsymbol{N} = \boldsymbol{T}_1 \otimes \boldsymbol{T}_1 + \boldsymbol{T}_2 \otimes \boldsymbol{T}_2 \tag{4.51}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{T}_{1,2}$ denote an arbitrary pair of orthogonal unit vectors on the reference propagation plane with normal $\boldsymbol{N}$. With the help of these expressions the eigenvalue problem becomes

$$\left[2\alpha\beta^2 \boldsymbol{p}_\alpha + \frac{W_{JJ}}{\beta^2}\mathit{\Lambda}_A^2\left(\boldsymbol{n} \otimes \boldsymbol{n}\right)\boldsymbol{p}_\alpha + 2\gamma\beta^2\left(\mathit{\Lambda}_T^2 \boldsymbol{I} - \boldsymbol{\Lambda}_T\right)\boldsymbol{p}_\alpha\right] = \rho_0 c_\alpha^2 \boldsymbol{p}_\alpha, \tag{4.52}$$

where the following notations have been used

$$\begin{aligned} \mathit{\Lambda}_A \boldsymbol{n} &= \boldsymbol{H}\boldsymbol{N}; \\ \mathit{\Lambda}_A^2 &= \boldsymbol{H}\boldsymbol{N} \cdot \boldsymbol{H}\boldsymbol{N}; \\ \boldsymbol{\Lambda}_T &= \boldsymbol{F}\boldsymbol{T}_1 \otimes \boldsymbol{F}\boldsymbol{T}_1 + \boldsymbol{F}\boldsymbol{T}_2 \otimes \boldsymbol{F}\boldsymbol{T}_2; \\ \mathit{\Lambda}_T^2 &= \boldsymbol{F}\boldsymbol{T}_1 \cdot \boldsymbol{F}\boldsymbol{T}_1 + \boldsymbol{F}\boldsymbol{T}_2 \cdot \boldsymbol{F}\boldsymbol{T}_2 = \mathrm{tr}\boldsymbol{\Lambda}_T. \end{aligned} \tag{4.53}$$

Note that $\boldsymbol{n}$ is a unit vector orthogonal to the vectors $\boldsymbol{FT}_{1,2}$ which lie on the propagation surface.

The first set of eigenvalues corresponding to p-waves is obtained by taking $\boldsymbol{p}_\alpha = \boldsymbol{n}$ to give,

$$c_{1,2} = c_p; \qquad c_p = \pm\sqrt{\frac{\left(2\alpha\beta^2 + 2\gamma\beta^2 \mathit{\Lambda}_T^2 + \frac{W_{JJ}}{\beta^2}\mathit{\Lambda}_A^2\right)}{\rho_0}}. \tag{4.54}$$

The remaining four eigenvalues correspond to shear waves where the vibration takes place on the propagation plane. The corresponding velocity vectors are orthogonal to $\boldsymbol{n}$ and in the directions of the unit eigenvectors $\{\boldsymbol{t}_1,\ \boldsymbol{t}_2\}$ of the rank two tensor $\boldsymbol{\mathit{\Lambda}}_T$. The wave speeds are given by $c_{3,4} = c_{s_1}$ and $c_{5,6} = c_{s_2}$, where

$$c_{s_1} = \pm\beta\sqrt{\frac{\left(2\alpha + 2\gamma\left(\mathit{\Lambda}_T^2 - \lambda_1^2\right)\right)}{\rho_0}}; \qquad c_{s_2} = \pm\beta\sqrt{\frac{\left(2\alpha + 2\gamma\left(\mathit{\Lambda}_T^2 - \lambda_2^2\right)\right)}{\rho_0}} \tag{4.55}$$

and $\lambda_{1,2}^2$ are the eigenvalues of $\boldsymbol{\mathit{\Lambda}}_T$. For a neo-Hookean model (e.g. $\alpha = \frac{\mu}{2}, \gamma = 0$), both the elastic pressure and shear wave speeds in the undeformed configuration ($\boldsymbol{F} = \boldsymbol{H} = \boldsymbol{I}$) become

$$c_{1,2} = \frac{1}{\beta}\sqrt{\frac{\lambda + \mu(\beta^4 + 1)}{\rho_0}}; \qquad c_{3,4} = c_{5,6} = \beta\sqrt{\frac{\mu}{\rho_0}}, \tag{4.56}$$

where β is a dimensionless parameter defined as $\beta := \frac{\kappa}{\tilde{\kappa}}$ and $\tilde{\kappa}$ is the fictitious bulk modulus. By substituting the expression of $\lambda = \kappa - \frac{2\mu}{3}$ into Eq. (4.56), the above wave speeds $c_{1,2}$ under nearly incompressible regime ($\kappa \gg \mu$) can be approximated as

$$c_{1,2} \approx \frac{1}{\beta}\sqrt{\frac{\kappa}{\rho_0}}. \tag{4.57}$$

Utilising equations (4.49), the corresponding right eigenvectors are obtained as

$$\boldsymbol{\mathcal{R}}_{1,2} = \begin{bmatrix} \boldsymbol{n} \\ -\frac{\beta^2}{\rho_0 c_{1,2}}\boldsymbol{n}\otimes\boldsymbol{N} \\ -\frac{\beta^2}{\rho_0 c_{1,2}}\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{n}\otimes\boldsymbol{N}) \\ -\frac{\mathit{\Lambda}_A}{\beta^2\rho_0 c_{1,2}} \end{bmatrix}; \qquad \boldsymbol{\mathcal{R}}_{3,4} = \begin{bmatrix} \boldsymbol{t}_1 \\ -\frac{\beta^2}{\rho_0 c_{3,4}}\boldsymbol{t}_1\otimes\boldsymbol{N} \\ -\frac{\beta^2}{\rho_0 c_{3,4}}\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{t}_1\otimes\boldsymbol{N}) \\ 0 \end{bmatrix};$$

$$\boldsymbol{\mathcal{R}}_{5,6} = \begin{bmatrix} \boldsymbol{t}_2 \\ -\frac{\beta^2}{\rho_0 c_{5,6}}\boldsymbol{t}_2\otimes\boldsymbol{N} \\ -\frac{\beta^2}{\rho_0 c_{5,6}}\boldsymbol{F}\boldsymbol{\times}(\boldsymbol{t}_2\otimes\boldsymbol{N}) \\ 0 \end{bmatrix}. \tag{4.58}$$

Finally, the following set of left eigenvectors can also be obtained in an analogous manner

$$\boldsymbol{\mathcal{L}}_{1,2} = \begin{bmatrix} \boldsymbol{n} \\ -\frac{2\alpha}{c_{1,2}}\left[\boldsymbol{n}\otimes\boldsymbol{N}\right] \\ -\frac{2\gamma}{c_{1,2}}\left[\boldsymbol{F}\times(\boldsymbol{n}\otimes\boldsymbol{N})\right] \\ -\frac{W_{JJ}\Lambda_A}{c_{1,2}} \end{bmatrix}; \qquad \boldsymbol{\mathcal{L}}_{3,4} = \begin{bmatrix} \boldsymbol{t}_1 \\ -\frac{2\alpha}{c_{3,4}}\left[\boldsymbol{t}_1\otimes\boldsymbol{N}\right] \\ -\frac{2\gamma}{c_{3,4}}\left[\boldsymbol{F}\times(\boldsymbol{t}_1\otimes\boldsymbol{N})\right] \\ 0 \end{bmatrix}; \tag{4.59}$$

$$\boldsymbol{\mathcal{L}}_{5,6} = \begin{bmatrix} \boldsymbol{t}_2 \\ -\frac{2\alpha}{c_{5,6}}\left[\boldsymbol{t}_2\otimes\boldsymbol{N}\right] \\ -\frac{2\gamma}{c_{5,6}}\left[\boldsymbol{F}\times(\boldsymbol{t}_2\otimes\boldsymbol{N})\right] \\ 0 \end{bmatrix}.$$

Chapter 5

INVOLUTIONS

5.1 Preliminaries

Many evolutionary equations in computational mechanics are supplemented with intrinsic constraints which must be satisfied throughout the whole evolutionary process. One of the greatest challenges in designing a robust computational framework is the fulfilment of these constraints over a long term response. In this work, the evolution of $\boldsymbol{F}$ and $\boldsymbol{H}$ must satisfy some compatibility conditions known as involutions. To be more precise, the deformation gradient $\boldsymbol{F}$ must be curl-free in order to guarantee the existence of a single-valued continuous displacement field and $\boldsymbol{H}$ must be divergence-free, in order to guarantee the existence of a single-valued continuous displacement field such that [94]

$$\text{CURL}\,\boldsymbol{F} = \boldsymbol{0}; \qquad \text{DIV}\,\boldsymbol{H} = \boldsymbol{0}. \tag{5.1}$$

If these involutions are not satisfied, spurious modes can accumulate which eventually lead to breakdown of the numerical scheme. In the case of linear elasticity, these involutions are known as Saint Venant compatibility conditions [116] [20]. Moreover, the time rate of equations (5.1) leads to the following expressions

$$\text{CURL}\,\dot{\boldsymbol{F}} = \boldsymbol{0}; \qquad \text{DIV}\dot{\boldsymbol{H}} = \boldsymbol{0}, \tag{5.2}$$

[20] In the case of linear elasticity, the compatibility equations can be written as $\text{curl}(\text{curl}\,\boldsymbol{\varepsilon}) = \boldsymbol{0}$ (in indicial notation $\mathcal{E}_{nml}\mathcal{E}_{ijk}\frac{\partial^2 \varepsilon_{lk}}{\partial x_j \partial x_m} = 0$), where $\boldsymbol{\varepsilon}$ represents the small strain tensor and the operator curl denotes the spatial curl operator.

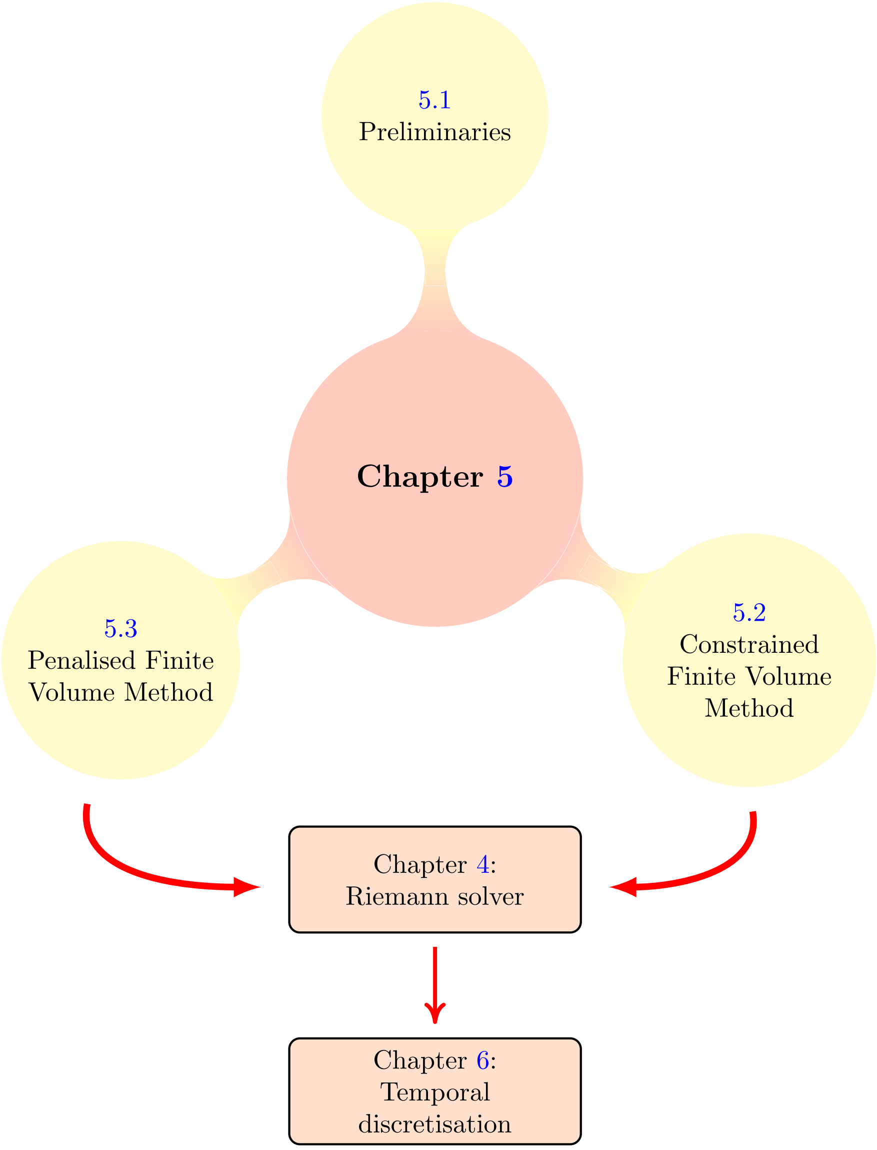


FIGURE 5.1: Structure of Chapter 5

which establish an equivalent set of involutions on the update of deformation gradient tensor in Eq. (3.3b) and cofactor of deformation in Eq. (3.3c). These 18 differential conditions in $\mathbb{R}^3$ need to be satisfied by the space-time evolution operator provided that they are met by the initial deformation gradient and cofactor of deformation [110]. This implies that the compatibility conditions Eq. (5.1) or Eq. (5.2), as opposed to classical constraints, are not necessary to close the system of conservation laws (2.22) but must be an inherent property of the space-time evolution operator.

In the context Magnetohydrodynamics (MHD), a wide variety of numerical methodologies are available for the construction of transport schemes fulfilling the satisfaction of specific involutions [107, 117–119]. For instance, one strategy is based on the use of a local involution-free projection scheme leading to the solution of a Poisson-type equation. This approach is known as elliptic correction [120] and relies upon the computation of an extra Lagrange multiplier parameter. A second strategy amends the conservation law equations via a convection-type correction [118], designed to advect spurious errors out of the computational domain. The third approach is based on the use of constrained transport algorithms [107, 117, 118], where the spatial discretisation is tailor-made to exactly satisfy the involutions by construction. Within the latter approach, a general framework for a locally curl-preserving Finite Volume Method on two-dimensional structured quadrilateral grids was proposed by Torrilhon *et al.* [107, 117] in the context of shallow water equations. This methodology was later explored and adapted by Lee *et al.* [6] for the curl preservation of deformation gradient tensor in the context of two-dimensional structured quadrilateral grids.

In this study, two new algorithms are introduced where the system of conservation laws is solved subjected to the fulfilment of involutions (5.2). The first algorithm, named as constrained-TOUCH (abbreviated to C-TOUCH), relies on the extension of the constrained transport method presented in [6] to three dimensional hexahedral elements where the overall scheme is recast into a Godunov-type method (see Section 5.2). The second algorithm, named as penalised-TOUCH (abbreviated to P-TOUCH), follows some of the ideas presented in [13] leading to a penalisation-based finite volume algorithm (see Section 5.3). The roadmap to this chapter is represented in Fig. 5.1.

5.2 Constrained Finite Volume Method

The satisfaction of involutions (5.2) can be guaranteed by considering the nodal linear momentum $\boldsymbol{p}_a$ and linear interpolation functions for the update of $\boldsymbol{F}$ and $\boldsymbol{H}$ such that [107, 117, 119]

$$\frac{d\boldsymbol{F}_e}{dt} = \frac{1}{\Omega_0^e} \sum_{a\in\Lambda_e^a} \frac{\boldsymbol{p}_a}{\rho_0} \otimes \boldsymbol{\nabla}_0 N_a(\boldsymbol{X}_e); \tag{5.3a}$$

$$\frac{d\boldsymbol{H}_e}{dt} = \boldsymbol{F}_e \boldsymbol{\times} \frac{1}{\Omega_0^e} \sum_{a\in\Lambda_e^a} \frac{\boldsymbol{p}_a}{\rho_0} \otimes \boldsymbol{\nabla}_0 N_a(\boldsymbol{X}_e), \tag{5.3b}$$

where $N_a(\boldsymbol{X}_e)$ represents the standard material finite element nodal shape function evaluated at the centroid of the cell e [21]. Eqs. (5.3a) and (5.3b) are naturally curl-free as the evolution of $\boldsymbol{F}$ and $\boldsymbol{H}$ is formulated in terms of a material discrete gradient of a continuous linear momentum field. Analogously, Eqs. (5.3a) and (5.3b) can also be expressed in terms of the outward nodal

[21] The shape functions and its derivatives for a hexahedral element can be found in Section B.3.

area vector $\boldsymbol{C}_{ea}$ as previously shown in Section 3.3.2 and is repeated here for convenience

$$\frac{d\boldsymbol{F}_e}{dt} = \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \frac{\boldsymbol{p}_a^C}{\rho_0} \otimes \boldsymbol{C}_{ea}; \tag{5.4a}$$

$$\frac{d\boldsymbol{H}_e}{dt} = \boldsymbol{F}_e \boldsymbol{\times} \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \frac{\boldsymbol{p}_a^C}{\rho_0} \otimes \boldsymbol{C}_{ea}. \tag{5.4b}$$

It is worth noticing that the nodal update for the deformation gradient tensor Eq. (5.4a) and the cofactor of deformation Eq. (5.4b) could be recast into a Godunov-type framework by using the definition of nodal area vectors Eq. (4.25) which is repeated here for convenience

$$\boldsymbol{C}_{ea} = \sum_{f \in \Lambda_a^f} \frac{\boldsymbol{C}_{ef}}{\Lambda_f^a}, \tag{5.5}$$

Using Eq. (5.5) the nodal update for $\boldsymbol{F}$, Eq. (5.4a), could be re-expressed in terms of the face area vector $\boldsymbol{C}_{ef}$ such that

$$\begin{aligned} \frac{d\boldsymbol{F}_e}{dt} &= \frac{1}{\Omega_0^e} \sum_{a \in \Lambda_e^a} \frac{\boldsymbol{p}_a}{\rho_0} \otimes \left(\sum_{f \in \Lambda_a^f} \frac{\boldsymbol{C}_{ef}}{\Lambda_f^a} \right) \\ &= \frac{1}{\Omega_0^e} \sum_{f \in \Lambda_e^f} \left(\frac{1}{\Lambda_f^a} \sum_{a \in \Lambda_f^a} \frac{\boldsymbol{p}_a}{\rho_0} \right) \otimes \boldsymbol{C}_{ef} \\ &= \frac{1}{\Omega_0^e} \sum_{f \in \Lambda_e^f} \frac{\tilde{\boldsymbol{p}}_f^C}{\rho_0} \otimes \boldsymbol{C}_{ef}. \end{aligned} \tag{5.6}$$

Analogously, the same applies for the update of $\boldsymbol{H}$ leading to

$$\frac{d\boldsymbol{H}_e}{dt} = \boldsymbol{F}_e \boldsymbol{\times} \frac{1}{\Omega_0^e} \sum_{f \in \Lambda_e^f} \frac{\tilde{\boldsymbol{p}}_f^C}{\rho_0} \otimes \boldsymbol{C}_{ef}, \tag{5.7}$$

where $\tilde{\boldsymbol{p}}_f^C$ is the projected contact linear momentum calculated based on the nodal linear momentum $\boldsymbol{p}_a$ which is yet to be defined.

One of the strategies proposed in [6, 25] is to obtain the nodal linear momentum by applying a weighted average of the contact linear momentum emerging from the Riemann solver $\boldsymbol{p}_a = \frac{1}{\Lambda_a^f} \sum_{f \in \Lambda_a^f} \boldsymbol{p}_f^C$. In the context of two-dimensional structured quadrilateral grids, it has been shown in [6] that certain boundary corrections are essential to enhance the accuracy near boundaries. However, in a three-dimensional scenario such corrections are more sophisticated and therefore undesirable.

In this study, an alternative but simpler approach entitled Constrained-TOUCH (C-TOUCH) is presented that does not require any such boundary corrections [8]. It is based on the use of a constrained transport algorithm [8], where the spatial discretisation is tailor-made to discretely satisfy the involutions by construction. In this approach, the evolution of $\boldsymbol{F}$ and $\boldsymbol{H}$

is re-formulated in terms of a material discrete gradient of a continuous velocity field. This can be achieved by replacing the Godunov-type numerical linear momentum $\boldsymbol{p}_f^C$ described in Eqs. (3.3b) and (3.3c) with a projected linear momentum $\tilde{\boldsymbol{p}}_f^C$, ensuring that this specific update exactly coincides with the classical finite element discretisation when considering linear interpolation with only one Gauss quadrature point at the centroid of the element [6, 8]. This methodology can be understood by the two-dimensional schematic shown in Fig. 5.2. It relies on the evaluation of a localised cell gradient $\boldsymbol{G}_e(\boldsymbol{p}_f^C)$, obtained through Eq. (3.14), which is based on a least square minimisation procedure (see Fig. 5.2a). This gradient requires to compute the cell averaged linear momentum $\bar{\boldsymbol{p}}_e$ which can be expressed as

$$\bar{\boldsymbol{p}}_e = \frac{1}{\Lambda_e^f} \sum_{f \in \Lambda_e^f} \boldsymbol{p}_f^C. \tag{5.8}$$

The neighbouring values for the computation of this gradient $\boldsymbol{G}_e(\boldsymbol{p}_f^C)$ are contact linear momentum $\boldsymbol{p}_f^C$ obtained through the Riemann solver. To ensure a monotonicity preserving numerical scheme, the gradient must then be further corrected through the procedure outlined in Section 3.4.2. This corrected gradient is then reconstructed at the nodes locally within cell e according to Eq. (3.18) leading to a discontinuous elemental nodal linear momentum $\boldsymbol{p}_{ea}$ as can be seen in Fig. 5.2b. A continuous nodal linear momentum $\boldsymbol{p}_a$ can be easily obtained by following a weighted average procedure (see Fig. 5.2c)

$$\boldsymbol{p}_a = \frac{1}{\Lambda_a^e} \sum_{e \in \Lambda_a^e} \boldsymbol{p}_{ea}. \tag{5.9}$$

The nodal linear momentum obtained in Eq. (5.9) may not be true at specific boundaries. Therefore, appropriate strong boundary conditions on $\boldsymbol{p}_a$ need to be enforced in the case of fixed, symmetric/contact and skew-symmetric boundaries such that

$$\text{Moving}: \quad \boldsymbol{p}_a = \boldsymbol{p}_B; \tag{5.10a}$$

$$\text{Symmetric/contact}: \quad \boldsymbol{p}_a = (\boldsymbol{I} - \boldsymbol{N}_{ef} \otimes \boldsymbol{N}_{ef})\, \boldsymbol{p}_a; \qquad \forall\ f \in \Lambda_a^f; \tag{5.10b}$$

$$\text{Skew-symmetric}: \quad \boldsymbol{p}_a = (\boldsymbol{N}_{ef} \otimes \boldsymbol{N}_{ef})\, \boldsymbol{p}_a; \qquad \forall\ f \in \Lambda_a^f; \tag{5.10c}$$

The procedure to obtain continuous nodal linear momentum is summarised in Algorithm 5.1. Finally, the nodal linear momentum $\boldsymbol{p}_a$ obtained through Eq. (5.9) can be projected back to the faces (see Fig. 5.2d)

$$\tilde{\boldsymbol{p}}_f^C = \frac{1}{\Lambda_f^a} \sum_{a \in \Lambda_f^a} \boldsymbol{p}_a, \tag{5.11}$$

to update $\boldsymbol{F}$ and $\boldsymbol{H}$ in a Godunov-type manner, similar to Eqs. (3.3b) and (3.3c), such that

$$\frac{d\boldsymbol{F}_e}{dt} = \frac{1}{\Omega_0^e} \sum_{f\in\Lambda_e^f} \frac{\tilde{\boldsymbol{p}}_f^C}{\rho_0} \otimes \boldsymbol{C}_{ef}; \tag{5.12}$$

$$\frac{d\boldsymbol{H}_e}{dt} = \boldsymbol{F}_e \boldsymbol{\times} \frac{1}{\Omega_0^e} \sum_{f\in\Lambda_e^f} \frac{\tilde{\boldsymbol{p}}_f^C}{\rho_0} \otimes \boldsymbol{C}_{ef}. \tag{5.13}$$

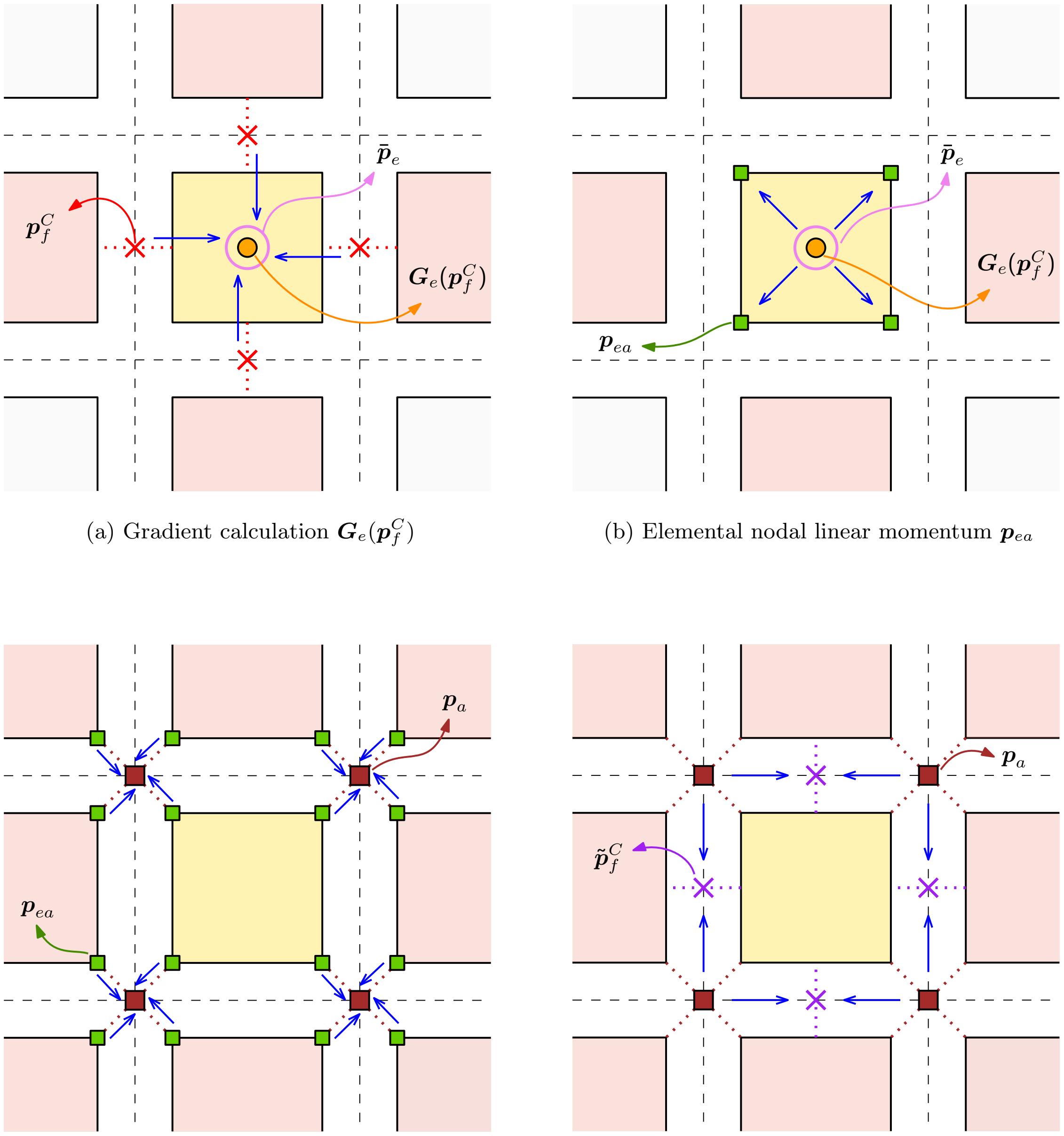
Algorithm 5.1: Evaluation of nodal linear momentum $\boldsymbol{p}_a$

```
Input  : $\boldsymbol{p}_f^C$
Output: $\boldsymbol{p}_a$
(1) Compute cell averaged linear momentum: $\bar{\boldsymbol{p}}_e \leftarrow$ Eq. (5.8)
(2) Compute cell gradient of $\boldsymbol{p}_f^C$: $\boldsymbol{G}_e(\boldsymbol{p}_f^C) \leftarrow$ Eq. (3.14)
(3) Apply slope limiter: $\phi_e \leftarrow$ Algorithm 3.1
(4) Obtain elemental nodal linear momentum: $\boldsymbol{p}_{ea} \leftarrow$ Eq. (3.18)
(5) Compute continuous nodal linear momentum: $\boldsymbol{p}_a \leftarrow$ Eq. (5.9)
(6) Apply strong boundary conditions on $\boldsymbol{p}_a$: $\boldsymbol{p}_a \leftarrow$ Eqs. (5.10)
```

5.3 Penalised Finite Volume Method

Following [13–16], an alternative penalisation-based methodology is presented in the context of the Finite Volume Method [8]. This approach is called the P-TOUCH scheme where a residual-based artificial diffusion is added into the fibre and volume map evolutions, Eqs. (2.11) and (2.16). The aim of this technique is to control the accumulation of non-physical curl/divergence errors, while still preserving the standard finite volume update for $\boldsymbol{F}$ Eq. (3.3b) and $\boldsymbol{H}$ Eq. (3.3c), without resorting to the computation of projected contact linear momentum $\tilde{\boldsymbol{p}}_f^C$ as in the C-TOUCH scheme (see Section 5.2).

For this reason, within each stage of the two-stage TVD Runge Kutta time integrator Eqs. (6.1) to (6.3), an augmented deformation gradient and its cofactor are introduced by incorporating



(a) Gradient calculation $\boldsymbol{G}_e(\boldsymbol{p}_f^C)$

(b) Elemental nodal linear momentum $\boldsymbol{p}_{ea}$

(c) Nodal linear momentum $\boldsymbol{p}_a$

(d) Projected contact linear momentum $\tilde{\boldsymbol{p}}_f^C$

Figure 5.2: Evaluation of projected contact linear momentum $\tilde{\boldsymbol{p}}_f^C$ in the C-TOUCH scheme.

additional geometrical penalisation, with the aim at eliminating spurious curl modes as follows

$$\boldsymbol{F}_e^{\chi} = \boldsymbol{F}_e^{\gamma} + \frac{\Delta t}{\Omega_0^e} \sum_{f \in \Lambda_e^f} \left(\frac{\boldsymbol{p}_f^{C,\gamma}}{\rho_0} \right) \otimes \boldsymbol{C}_{ef} + \underbrace{\xi_{\boldsymbol{F}} \left(\boldsymbol{F}_{\boldsymbol{x},e}^{\gamma} - \boldsymbol{F}_e^{\gamma} \right)}_{\text{Penalisation}}; \tag{5.14}$$

$$\boldsymbol{H}_e^{\chi} = \boldsymbol{H}_e^{\gamma} + \boldsymbol{F}_e^{\gamma} \times \frac{\Delta t}{\Omega_0^e} \sum_{f \in \Lambda_e^f} \left(\frac{\boldsymbol{p}_f^{C,\gamma}}{\rho_0} \right) \otimes \boldsymbol{C}_{ef} + \underbrace{\xi_{\boldsymbol{H}} \left(\boldsymbol{H}_{\boldsymbol{x},e}^{\gamma} - \boldsymbol{H}_e^{\gamma} \right)}_{\text{Penalisation}}, \tag{5.15}$$

where $\chi = \{*, **\}$ and $\gamma = \{n, *\}$. Moreover, $\boldsymbol{F}_{\boldsymbol{x}}$ represents the deformation gradient tensor and $\boldsymbol{H}_{\boldsymbol{x}}$, the cofactor of deformation based on geometry. They can be computed as

$$\boldsymbol{F}_{\boldsymbol{x}.e}^{\gamma} = \frac{1}{\Omega_0^e} \sum_{f \in \Lambda_e^f} \boldsymbol{x}_f^{\gamma} \otimes \boldsymbol{C}_{ef}; \qquad \boldsymbol{H}_{\boldsymbol{x},e}^{\gamma} = \frac{1}{2} \boldsymbol{F}_e^{\gamma} \times \boldsymbol{F}_e^{\gamma}. \tag{5.16}$$

The non-dimensional penalisation parameters $\xi_{\boldsymbol{F}}$ and $\xi_{\boldsymbol{H}}$ are usually defined in the range of $[0-0.5]$ [13–16]. It is worth pointing out that zero penalisation (i.e. $\xi_{\boldsymbol{F}} = \xi_{\boldsymbol{H}} = 0$) in Eqs. (5.14) and (5.15) recovers the standard finite volume update for the geometric strains $\boldsymbol{F}$ and $\boldsymbol{H}$ (see Eqs. (3.3b) and (3.3c)).

Finally, the P-TOUCH scheme is summarised in Algorithm 5.2. Notice that although P-TOUCH does not require the calculation of projected contact linear momentum $\tilde{\boldsymbol{p}}_f^C$ (see Eq. (5.11)), it still requires the evaluation of nodal linear momentum $\boldsymbol{p}_a$ as outlined in Algorithm 5.1 to calculate $\boldsymbol{F}_{\boldsymbol{x}}$ and for visualisation purposes.

Algorithm 5.2: Penalisation based $\boldsymbol{F}_e$ and $\boldsymbol{H}_e$ update

```
Input  : $\boldsymbol{F}_e^{\gamma}$, $\boldsymbol{H}_e^{\gamma}$, $\boldsymbol{p}_f^{C,\gamma}$, $\boldsymbol{x}_a^{\gamma}$    where $\gamma = \{n, *\}$
Output: $\boldsymbol{F}_e^{\chi}$, $\boldsymbol{H}_e^{\chi}$                 where $\chi = \{*, **\}$

(1) Calculate face centre coordinates:  $\boldsymbol{x}_f^{\gamma} = \frac{1}{\Lambda_f^a} \sum_{a \in \Lambda_f^a} \boldsymbol{x}_a^{\gamma}$
(2) Compute deformation gradient based on geometry:  $\boldsymbol{F}_{\boldsymbol{x},e}^{\gamma} \leftarrow$ Eq. (5.16)
(3) Compute cofactor of deformation gradient based on $\boldsymbol{F}$:  $\boldsymbol{H}_{\boldsymbol{F},e}^{\gamma} \leftarrow$ Eq. (5.16)
(4) Update deformation gradient:  $\boldsymbol{F}_e^{\chi} \leftarrow$ Eq. (5.14)
(5) Update cofactor of deformation gradient:  $\boldsymbol{H}_e^{\chi} \leftarrow$ Eq. (5.15)
```

Chapter 6

TEMPORAL DISCRETISATION

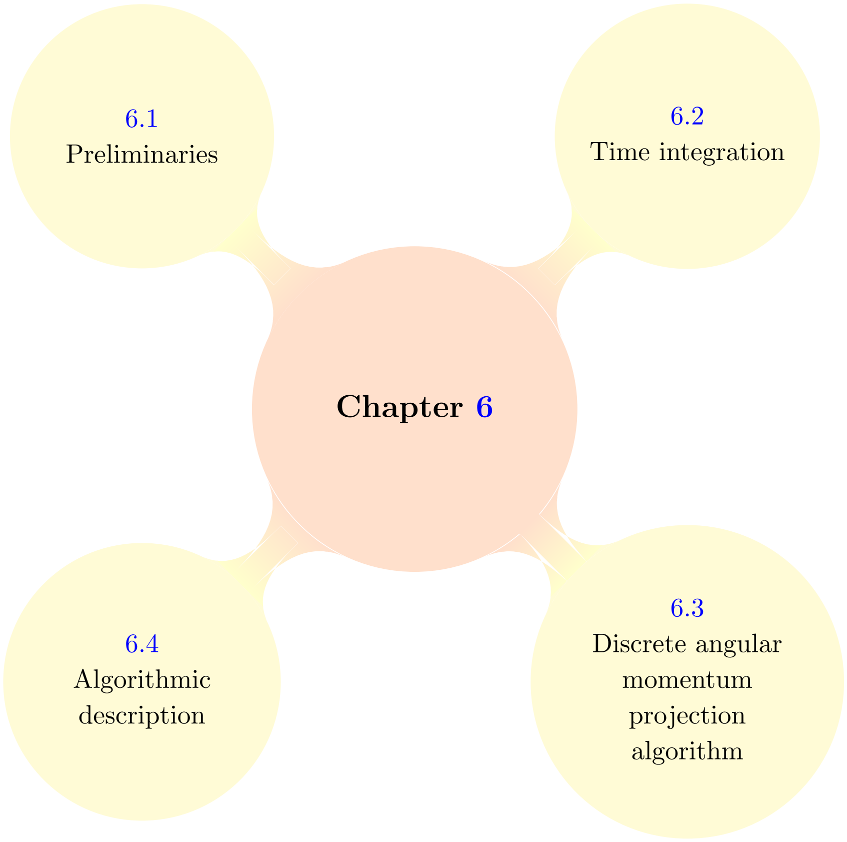


Figure 6.1: Structure of Chapter 6

6.1 Preliminaries

In Section 3.4, a linear reconstruction procedure was introduced to increase the spatial accuracy from first order to second. For dynamic (time-dependent) problems this higher order discretisation in space should also be supplemented by a high order discretisation in time [102] [22]. Therefore, in Section 6.2, a one-step two-stage Total Variation Diminishing (TVD) Runge-Kutta time integration scheme is introduced to ensure second order accuracy in time. Moreover, for completeness, a global angular momentum projection algorithm is presented in Section 6.3 to preserve the angular momentum of the system. Finally, in Section 6.4, a complete algorithmic description of the proposed C-TOUCH, P-TOUCH and X-GLACE schemes is outlined.

6.2 Time integration

As discussed earlier, the objective of this thesis is to simulate fast-transient solid dynamic problems which implies that small time increments must be utilised to accurately capture the deformation process. This becomes even more apparent in the case of time dependent constitutive models such as plasticity. Additionally, the resulting set of semi-discrete equations is rather large that it will only be suitable to employ an explicit time integrator.

A popular family of explicit time marching schemes are the multistage Runge-Kutta schemes. For simplicity, an explicit one-step two-stage Total Variation Diminishing Runge-Kutta (TVD-RK) scheme has been used as already explored in [6, 8, 10, 121] [23]. This is described by the following time update equations from time step t^n to t^{n+1}

$$\boldsymbol{\mathcal{U}}_e^{\star} = \boldsymbol{\mathcal{U}}_e^{n} + \Delta t\, \dot{\boldsymbol{\mathcal{U}}}_e^{n}(\boldsymbol{\mathcal{U}}_e^{n}, t^n) \tag{6.1}$$

$$\boldsymbol{\mathcal{U}}_e^{\star\star} = \boldsymbol{\mathcal{U}}_e^{\star} + \Delta t\, \dot{\boldsymbol{\mathcal{U}}}_e^{\star}(\boldsymbol{\mathcal{U}}_e^{\star}, t^{n+1}) \tag{6.2}$$

$$\boldsymbol{\mathcal{U}}_e^{n+1} = \frac{1}{2}(\boldsymbol{\mathcal{U}}_e^{n} + \boldsymbol{\mathcal{U}}_e^{\star\star}). \tag{6.3}$$

[22] For steady state problems, a second order spatial discretisation accompanied by a first order temporal discretisation will eventually converge to a second order accurate approximation despite not being second order accurate in time [102].

[23] The concept of Total Variation Diminishing (TVD) schemes was first introduced by Harten in 1983. The aim was to prevent the occurrence of new extrema within the solution domain (i.e. monotonicity preserving). A formulation with TVD properties allows shock capturing without the appearance of any spurious oscillations in the solution [105].

It is worthwhile pointing out that in this study, the geometry is also updated through the TVD-RK algorithm. This results in a monolithic time integration procedure where the conservation variables $\boldsymbol{\mathcal{U}} = \{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J, E\}$ and geometry $\boldsymbol{x}$ are all updated through Eqs. (6.1) to (6.3). The maximum time step $\varDelta t = t^{n+1} - t^n$ is governed by a standard Courant-Friedrichs-Lewy (CFL) condition [122] to ensure the correct physical speeds for information propagation and thus avoiding any wave interaction within the element [24].

$$\varDelta t = \alpha_{\text{CFL}} \frac{h_{\text{min}}}{c_{p,\text{max}}}, \tag{6.4}$$

where $c_{p,\,\text{max}}$ is the maximum p-wave speed (refer to Section 2.7 for its evaluation), h_{min} is the minimum (or characteristic) length within the computational domain and α_{CFL} is the CFL stability number. For the numerical computations presented in this study, a value of $\alpha_{\text{CFL}} = 0.3$ has been chosen, unless otherwise stated, to ensure both accuracy and stability.

6.3 Discrete angular momentum projection algorithm

The proposed mixed methodologies presented in this study, namely (a) C-TOUCH scheme Eqs. (3.3a), (3.3d), (3.3e), (5.12) and (5.13); (b) P-TOUCH scheme Eqs. (3.3a), (3.3d), (3.3e), (5.14) and (5.15); and (c) X-GLACE nodal solver Eqs. (3.6a) to (3.6e), do not intrinsically fulfil conservation of angular momentum, since the deformation gradient $\boldsymbol{F}$ is no longer 'strongly' obtained as the material gradient of the current geometry (e.g. $\boldsymbol{F} \neq \boldsymbol{F_x} := \boldsymbol{\nabla}_0 \boldsymbol{x}$). To rectify this, a projection-based method was presented in [6, 10] where the interface contact tractions were appropriately modified to guarantee global angular momentum preservation. This approach, however, implies a modification of the interface contact tractions which can affect the overall stability/accuracy of the scheme as a result of a reduction/increase in the numerical dissipation introduced by the interface fluxes. An alternative approach has been proposed by Després and Labourasse [108], which incorporates an additional angular momentum conservation law into the finite volume nodal solver system Eqs. (3.6a) to (3.6e). The use of this extra conservation law in conjunction with a local linear reconstruction procedure within every cell is shown to lead, via the use of a Lagrange multiplier, to the preservation of the angular momentum.

In this study, following the work in [10], a new variant of the discrete angular momentum projection algorithm is carried out and applied to the TOUCH schemes. Unlike the previous approaches proposed in [6, 15], in this study the local linear momentum update $\dot{\boldsymbol{p}}$ is modified (in the least squares sense) in order to preserve the total angular momentum, whilst still ensuring the conservation of global linear momentum. The conservation of discrete angular momentum

[24] The CFL condition is extremely important for any finite volume or finite difference method to ensure stability and convergence. The Courant number measures the fraction of a grid cell that information propagates through in one time step.

after a time step can be written as

$$\sum_e \Omega_0^e \, \boldsymbol{x}_e^{n+1} \times \boldsymbol{p}_e^{n+1} - \sum_e \Omega_0^e \, \boldsymbol{x}_e^{n} \times \boldsymbol{p}_e^{n} = \boldsymbol{T}_{\text{tor}}, \tag{6.5}$$

where $\boldsymbol{T}_{\text{tor}}$ is the external torque applied to the system. By taking into account the one-step two-stage TVD Runge-Kutta time integrator (see Section 6.2) for the time integration of the geometry, Eq. (6.5) can be rewritten as [25]

$$\sum_e \Omega_0^e \, \boldsymbol{x}_e^{n+1/2} \times \left(\boldsymbol{p}_e^{n+1} - \boldsymbol{p}_e^{n}\right) + \frac{\Delta t}{2\rho_0} \sum_e \Omega_0^e \, \boldsymbol{p}_e^{\star} \times \boldsymbol{p}_e^{n+1} = \boldsymbol{T}_{\text{tor}}, \tag{6.6}$$

where

$$\boldsymbol{x}_e^{n+1/2} = \boldsymbol{x}_e^{n} + \frac{\Delta t}{2\rho_0} \boldsymbol{p}_e^{n}. \tag{6.7}$$

Equivalently, Eq. (6.6) can be re-expressed as

$$\sum_e \Omega_0^e \left(\boldsymbol{x}_e^{n+1/2} + \frac{\Delta t}{2\rho_0} \boldsymbol{p}_e^{\star}\right) \times \left(\boldsymbol{p}_e^{n+1} - \boldsymbol{p}_e^{n}\right) - \frac{\Delta t}{2\rho_0} \sum_e \Omega_0^e \, \boldsymbol{p}_e^{\star} \times \left(\boldsymbol{p}_e^{\star} - \boldsymbol{p}_e^{n}\right) = \boldsymbol{T}_{\text{tor}}. \tag{6.8}$$

Considering the time integration as presented in Eqs. (6.1) to (6.3), the linear momentum increment reads

$$\boldsymbol{p}_e^{n+1} - \boldsymbol{p}_e^{n} = \frac{\Delta t}{2} \left(\dot{\boldsymbol{p}}_e^{n} + \dot{\boldsymbol{p}}_e^{\star}\right); \qquad \boldsymbol{p}_e^{\star} - \boldsymbol{p}_e^{n} = \Delta t \, \dot{\boldsymbol{p}}_e^{n}, \tag{6.9}$$

Substituting Eq. (6.9) into Eq. (6.8), the following equation is obtained after some algebraic manipulations

$$\sum_e \Omega_0^e \, \boldsymbol{x}_e^{n} \times \dot{\boldsymbol{p}}_e^{n} + \sum_e \Omega_0^e \left(\boldsymbol{x}_e^{n+1/2} + \frac{\Delta t}{2\rho_0} \boldsymbol{p}_e^{\star}\right) \times \dot{\boldsymbol{p}}_e^{\star} = \boldsymbol{T}_{\text{tor}}. \tag{6.10}$$

Sufficient conditions to satisfy Eq. (6.10) within a time step are enforced at each stage of the two-stage Runge Kutta time integrator described as

$$\sum_e \Omega_0^e \left(\boldsymbol{\mathcal{X}}_e^{\chi} \times \dot{\boldsymbol{p}}_e^{\chi}\right) = \boldsymbol{T}_{\text{tor}}; \qquad \forall \chi = \{n, \star\}, \tag{6.11}$$

where

$$\boldsymbol{\mathcal{X}}_e = \begin{cases} \boldsymbol{x}_e^{n}, & \chi = n \\ \boldsymbol{x}_e^{n+1/2} + \frac{\Delta t}{2\rho_0} \boldsymbol{p}_e^{\star}, & \chi = \star, \end{cases} \tag{6.12}$$

and

$$\dot{\boldsymbol{p}}_e^{\chi} = \sum_{f \in \Lambda_e^f} \boldsymbol{t}_f^{C,\chi} \, \|\boldsymbol{C}_{ef}\|. \tag{6.13}$$

A Lagrangian projection procedure is used to ensure the satisfaction of angular momentum

[25] As reported in [6, 10, 11, 13–16], Eq. (6.6) simplifies to to $\sum_e \Omega_0^e \, \boldsymbol{x}_e^{n+1/2} \times \left(\boldsymbol{p}_e^{n+1} - \boldsymbol{p}_e^{n}\right) = \boldsymbol{T}_{\text{Torque}}$, when using the trapezoidal rule for time integration of geometry.

constraint Eq. (6.11), as well as the conservation of linear momentum Eq. (2.7). This can be achieved by considering the minimisation of the following functional $\mathit{\Pi}$ as[26]

$$\begin{aligned}\mathit{\Pi}(\tilde{\dot{\boldsymbol{p}}}_e, \boldsymbol{\lambda}_{\text{ang}}, \boldsymbol{\lambda}_{\text{lin}}) = \frac{1}{2}\sum_e \mathit{\Omega}_0^e \left(\tilde{\dot{\boldsymbol{p}}}_e - \dot{\boldsymbol{p}}_e\right) \cdot \left(\tilde{\dot{\boldsymbol{p}}}_e - \dot{\boldsymbol{p}}_e\right) + \boldsymbol{\lambda}_{\text{ang}} \cdot \left[\sum_e \left(\boldsymbol{T}^e_{\text{tor}} - \mathit{\Omega}_0^e(\boldsymbol{\mathcal{X}}_e \times \tilde{\dot{\boldsymbol{p}}}_e)\right)\right] \\ + \boldsymbol{\lambda}_{\text{lin}} \cdot \left[\sum_e \left(\boldsymbol{T}^e_{\text{for}} - \mathit{\Omega}_0^e \tilde{\dot{\boldsymbol{p}}}_e\right)\right].\end{aligned} \tag{6.14}$$

Here, $\tilde{\dot{\boldsymbol{p}}}_e$ indicates the enhanced time variation of the elemental linear momentum, $\{\boldsymbol{\lambda}_{\text{ang}}, \boldsymbol{\lambda}_{\text{lin}}\}$ are two global Lagrange multiplier vectors associated with the angular momentum constraint and the linear momentum constraint. $\boldsymbol{T}^e_{\text{tor}}$ is the elemental torque and $\boldsymbol{T}^e_{\text{for}}$ is the elemental force which can be computed as

$$\boldsymbol{T}^e_{\text{tor}} = \sum_{f \in \mathit{\Lambda}_e^f} \left(\boldsymbol{x}_f \times \boldsymbol{t}_f^C\right) \|\boldsymbol{C}_{ef}\|, \tag{6.15}$$

$$\boldsymbol{T}^e_{\text{for}} = \sum_{f \in \mathit{\Lambda}_e^f} \boldsymbol{t}_f^C \|\boldsymbol{C}_{ef}\| + \rho_0 \boldsymbol{b}_e. \tag{6.16}$$

The directional derivative of the functional $\mathit{\Pi}$ with respect to $\tilde{\dot{\boldsymbol{p}}}$ gives

$$\frac{\partial \mathit{\Pi}}{\partial \tilde{\dot{\boldsymbol{p}}}} = \boldsymbol{0} = \sum_e \mathit{\Omega}_0^e \left(\tilde{\dot{\boldsymbol{p}}}_e - \dot{\boldsymbol{p}}_e\right) - \boldsymbol{\lambda}_{\text{ang}} \times \sum_e \mathit{\Omega}_0^e \boldsymbol{\mathcal{X}}_e - \boldsymbol{\lambda}_{\text{lin}} \sum_e \mathit{\Omega}_0^e. \tag{6.17}$$

The enhanced time variation of the linear momentum can therefore be obtained as

$$\boxed{\tilde{\dot{\boldsymbol{p}}}_e = \dot{\boldsymbol{p}}_e + \boldsymbol{\lambda}_{\text{ang}} \times \boldsymbol{\mathcal{X}}_e + \boldsymbol{\lambda}_{\text{lin}}.} \tag{6.18}$$

The Lagrange multipliers $\{\boldsymbol{\lambda}_{\text{ang}}, \boldsymbol{\lambda}_{\text{lin}}\}$ are the solutions of the following system of equations

$$\begin{bmatrix} \boldsymbol{\lambda}_{\text{ang}} \\ \\ \boldsymbol{\lambda}_{\text{lin}} \end{bmatrix} = \begin{bmatrix} \sum_e \mathit{\Omega}_0^e \left[\boldsymbol{\mathcal{X}}_e \otimes \boldsymbol{\mathcal{X}}_e - (\boldsymbol{\mathcal{X}}_e \cdot \boldsymbol{\mathcal{X}}_e)\boldsymbol{I}\right] & -\sum_e \mathit{\Omega}_0^e \hat{\boldsymbol{\mathcal{X}}}_e \\ \sum_e \mathit{\Omega}_0^e \hat{\boldsymbol{\mathcal{X}}}_e & -\sum_e \mathit{\Omega}_0^e \end{bmatrix}^{-1} \begin{bmatrix} \sum_e \mathit{\Omega}_0^e (\boldsymbol{\mathcal{X}}_e \times \dot{\boldsymbol{p}}_e) - \boldsymbol{T}^e_{\text{tor}} \\ \sum_e (\mathit{\Omega}_0^e \dot{\boldsymbol{p}}_e - \boldsymbol{T}^e_{\text{for}}) \end{bmatrix}, \tag{6.19}$$

with the indicial notation $\left[\hat{\boldsymbol{\mathcal{X}}}_e\right]_{ik} = \mathcal{E}_{ijk}\left[\boldsymbol{\mathcal{X}}_e\right]_j$. The angular momentum projection algorithm is summarised in Algorithm 6.1.

[26] The upper indices indicating time step (e.g. $n, n+1/2, *$) have been removed for simplicity.

Algorithm 6.1: Angular momentum projection algorithm

```
Input  : $\boldsymbol{p}_e$
Output: $\tilde{\dot{\boldsymbol{p}}}_e$

(1) Calculate rate of linear momentum:  $\dot{\boldsymbol{p}}_e \leftarrow$ Eq. (6.13)
(2) Calculate elemental torque :  $\boldsymbol{T}^e_{\text{tor}} \leftarrow$ Eq. (6.15)
(3) Calculate elemental force :  $\boldsymbol{T}^e_{\text{for}} \leftarrow$ Eq. (6.16)
(4) Compute coordinates for AMPA:  $\boldsymbol{\mathcal{X}}_e \leftarrow$ Eq. (6.12)
(5) Obtain Lagrange multipliers:  $\boldsymbol{\lambda}_{\text{Ang}}, \boldsymbol{\lambda}_{\text{Lin}} \leftarrow$ Eq. (6.19)
(6) Compute enhanced rate of linear momentum:  $\tilde{\dot{\boldsymbol{p}}}_e \leftarrow$ Eq. (6.18)
```

Remark 3: Similarly, angular momentum projection algorithm (Algorithm 6.1) can also be applied to the X-GLACE scheme by solving the global system (6.19) but with appropriate computation of elemental torque and elemental forces such that

$$\boldsymbol{T}^e_{\text{tor}} = \sum_{a \in \Lambda^a_e} \left(\boldsymbol{x}_a \times \boldsymbol{t}^C_{ea}\right) \|\boldsymbol{C}_{ea}\|; \qquad \boldsymbol{T}^e_{\text{for}} = \sum_{a \in \Lambda^a_e} \boldsymbol{t}^C_{ea} \|\boldsymbol{C}_{ea}\| + \rho_0 \boldsymbol{b}_e. \tag{6.20}$$

6.4 Algorithmic description

For ease of understanding, Algorithm 6.2 summarises the complete algorithmic description of the following $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J, E\}$ methodologies: C-TOUCH, P-TOUCH and X-GLACE.

Algorithm 6.2: Time update of conservation variables

```
Input  : $\boldsymbol{\mathcal{U}}_e^n$      where $\boldsymbol{\mathcal{U}} = [\boldsymbol{p}\ \boldsymbol{F}\ \boldsymbol{H}\ J\ E]^T$
Output: $\boldsymbol{\mathcal{U}}_e^{n+1}$, $\boldsymbol{P}_e^{n+1}$

  (1) Calculate time increment:   $\varDelta t^n \leftarrow$ Eq. (6.4)

  (2) Store conservation variables:   $\boldsymbol{\mathcal{U}}_e^{\text{old}} = \boldsymbol{\mathcal{U}}_e^n$

  (3) Loop over Runge-Kutta stages
      for $\mathrm{R-K}$ stage $= 1$ to $2$ do
      |   (3.1) Evaluate wave speeds:   $\tilde{c}_p,\ \tilde{c}_s \leftarrow$ Eq. (4.47)
      |
      |   (3.2) Apply linear reconstruction procedure (see Section 3.4)
      |
      |   (3.3) Apply acoustic Riemann solver:
      |         if ( algorithm = TOUCH ) then
      |         |     • Calculate Godunov-type fluxes:   $\boldsymbol{p}_f^C,\ \boldsymbol{t}_f^C \leftarrow$ Eqs. (4.10) and (4.11)
      |         |     • Compute nodal linear momentum:  $\boldsymbol{p}_a \leftarrow$ Algorithm 5.1
      |         |     • Compute projected contact linear momentum:   $\tilde{\boldsymbol{p}}_f^C \leftarrow$ Eq. (5.11)
      |         else if ( algorithm = X−GLACE ) then
      |         |     • Calculate nodal fluxes:   $\boldsymbol{p}_a^C,\ \boldsymbol{t}_{ea}^C \leftarrow$ Eqs. (4.29) and (4.31)
      |         end
      |
      |   (3.4) Enforce boundary conditions on nodal linear momentum:   $\boldsymbol{p}_a$ or $\boldsymbol{p}_a^C \leftarrow$ Eq. (5.10)
      |
      |   (3.5) Apply angular momentum projection algorithm (see Algorithm 6.1).
      |
      |   (3.6) Solve governing equations:
      |         if ( algorithm = C−TOUCH ) then
      |         |     • $\boldsymbol{\mathcal{U}}_e = \boldsymbol{\mathcal{U}}_e + \varDelta t^n\,\dot{\boldsymbol{\mathcal{U}}}_e\left(\tilde{\boldsymbol{p}}_f^C, \boldsymbol{t}_f^C\right)$
      |         else if ( algorithm = P−TOUCH ) then
      |         |     • $\boldsymbol{\mathcal{U}}_e = \boldsymbol{\mathcal{U}}_e + \varDelta t^n\,\dot{\boldsymbol{\mathcal{U}}}_e\left(\boldsymbol{p}_f^C, \boldsymbol{t}_f^C\right)$
      |         |     • $\boldsymbol{F}_e = \boldsymbol{F}_e\,(1-\xi_{\boldsymbol{F}}) + \xi_{\boldsymbol{F}}\,(\boldsymbol{\nabla}_0\boldsymbol{x})_e$
      |         |     • $\boldsymbol{H}_e = \boldsymbol{H}_e\,(1-\xi_{\boldsymbol{H}}) + \frac{1}{2}\xi_{\boldsymbol{H}}\,(\boldsymbol{\nabla}_0\boldsymbol{x} \times \boldsymbol{\nabla}_0\boldsymbol{x})_e$
      |         else if ( algorithm = X−GLACE ) then
      |         |     • $\boldsymbol{\mathcal{U}}_e = \boldsymbol{\mathcal{U}}_e + \varDelta t^n\,\dot{\boldsymbol{\mathcal{U}}}_e\ (\boldsymbol{p}_a^C,\ \boldsymbol{t}_{ea}^C)$
      |         end
      end

  (4) Update conservation variables:   $\boldsymbol{\mathcal{U}}_e^{n+1} = \frac{1}{2}\left(\boldsymbol{\mathcal{U}}_e + \boldsymbol{\mathcal{U}}_e^{\text{old}}\right)$

  (5) Compute PK1 stresses:   $\boldsymbol{P}_e^{n+1} \leftarrow$ (see Section 2.6)
```

Chapter 7

IMPLEMENTATION IN OPENFOAM

7.1 Preliminaries

OpenFOAM[27], an acronym for Open Source Field Operation and Manipulation, is a Computational Fluid Dynamics (CFD) toolbox widely used across academic and industrial environments. This cell centred Finite Volume Method (FVM) code is licensed under the open source General Public License (GPL) which gives user the flexibility to freely download, install, use and modify this high-end code. Perhaps the biggest advantage of OpenFOAM lies in its C++ implementation and thus the objected oriented nature of programming. As discussed in [123], the object oriented approach has led to the creation of a library of C++ classes which makes it possible to implement complicated mathematical and physical models. The top level syntax of the code closely resembles the standard vector and tensor notation which results in a code that is easier to write, validate and maintain than conventional procedural codes such as Fortran. Object oriented techniques [124, 125] such as abstraction, inheritance, polymorphism and operator overloading have been effectively utilized in creation of the code which is able to solve various types of problems.

Although, OpenFOAM was primarily developed for solving CFD problems, it provides a library of solvers capable of tackling various continuum mechanics problems including solid solvers. However, the solid solvers are only limited to linear elastic cases within the small strain deformation regime. OpenFOAM is a C++ library, used primarily to create executables, known

[27] OpenFOAM is a registered trademark of the ESI group.

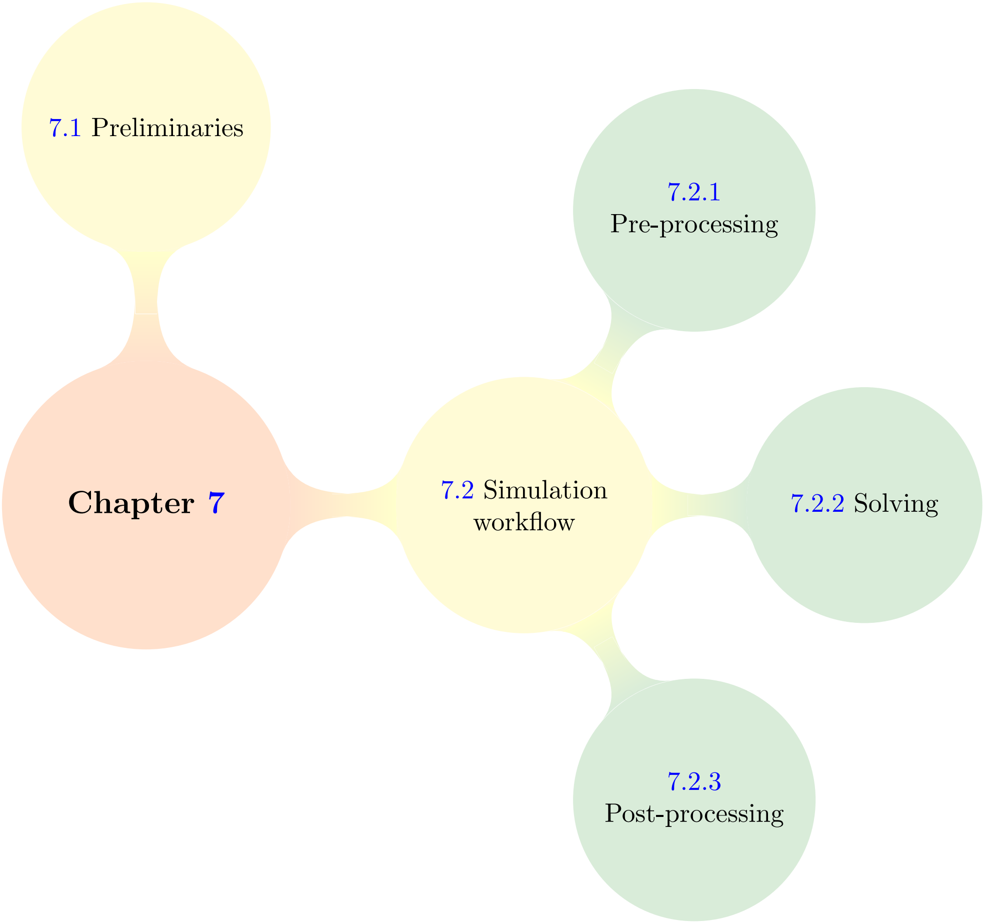
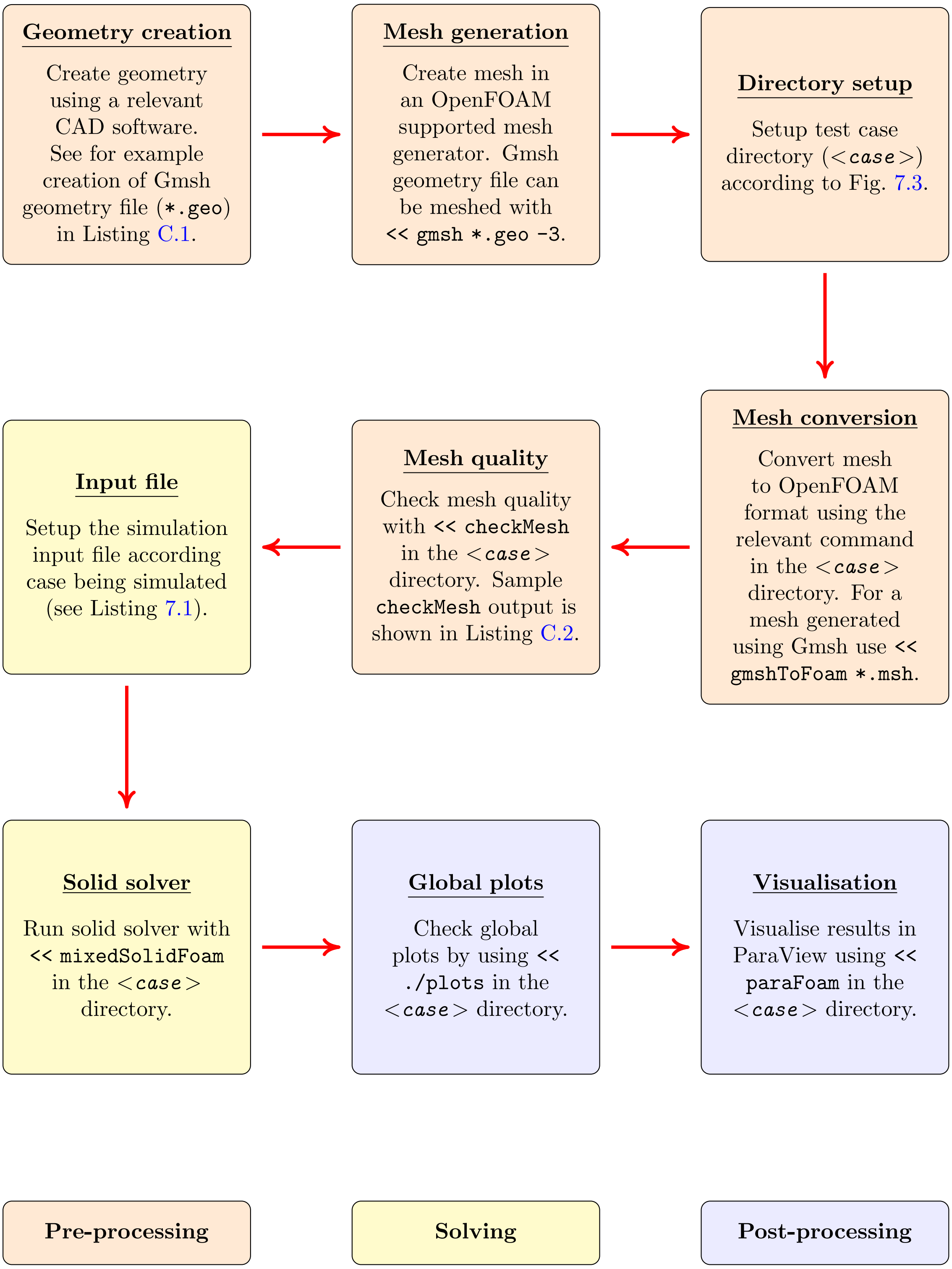


FIGURE 7.1: Structure of Chapter 7

as applications. These applications fall into two categories: solvers, that are each designed to solve a specific problem in continuum mechanics; and utilities, that are designed to perform tasks that involve data manipulation [126]. The structure of this chapter is shown in Fig. 7.1.

7.2 Simulation workflow

It is well known that undertaking any numerical simulation comprises of at least three major steps, namely (a) pre-processing; (b) solving; and (c) post-processing. The simulation workflow is summarised in Fig. 7.2.



Geometry creation

Create geometry using a relevant CAD software. See for example creation of Gmsh geometry file (`*.geo`) in Listing C.1.

Mesh generation

Create mesh in an OpenFOAM supported mesh generator. Gmsh geometry file can be meshed with `<< gmsh *.geo -3`.

Directory setup

Setup test case directory (<*`case`*>) according to Fig. 7.3.

Mesh conversion

Convert mesh to OpenFOAM format using the relevant command in the <*`case`*> directory. For a mesh generated using Gmsh use `<< gmshToFoam *.msh`.

Mesh quality

Check mesh quality with `<< checkMesh` in the <*`case`*> directory. Sample `checkMesh` output is shown in Listing C.2.

Input file

Setup the simulation input file according case being simulated (see Listing 7.1).

Solid solver

Run solid solver with `<< mixedSolidFoam` in the <*`case`*> directory.

Global plots

Check global plots by using `<< ./plots` in the <*`case`*> directory.

Visualisation

Visualise results in ParaView using `<< paraFoam` in the <*`case`*> directory.

FIGURE 7.2: Simulation workflow in OpenFOAM

7.2.1 Pre-processing

The first step when simulating any computational problem is the creation of geometry. Any suitable CAD software can be used to generate the geometry. It is important to point out that OpenFOAM only supports geometries created in a three-dimensional setting. A two-dimensional analysis can be carried out by specifying some special boundary conditions in the third direction [126]. In this study most of the geometries have been generated using Gmsh version 2.16 [127] and the rest with SolidWorks 2017 [128] [28].

Once the geometry is created, it must be meshed using an appropriate mesh generator which is supported by OpenFOAM. This is essential, since later, the generated mesh has to be converted to OpenFOAM format [29]. The Gmsh geometry file (`*.geo`) can be used to generate mesh through the command `<< gmsh *.geo -3`. Here, the `*.geo` file is converted to a mesh file (`*.msh`) where the option `-3` instructs Gmsh to create a 3D mesh. Meshes for the geometries created in SolidWorks have been created using ANSYS Workbench v16.2 and exported in a Fluent mesh format (`*.msh`).

After mesh generation, the next step is to set-up the <*case*> directory [30] as illustrated in

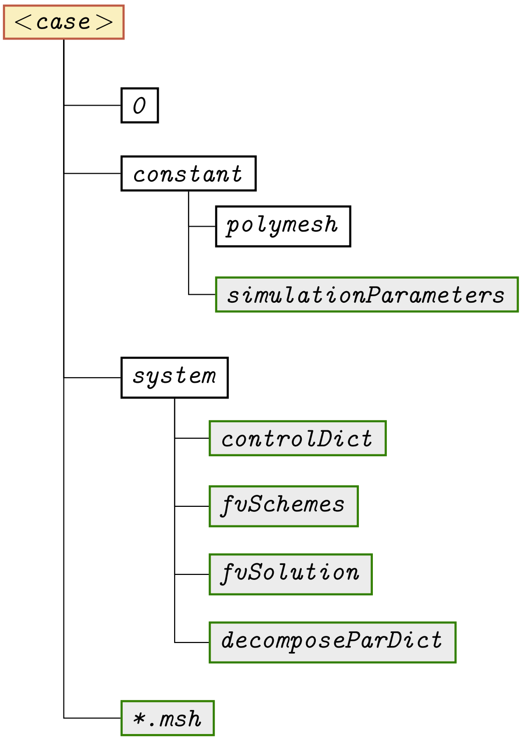


FIGURE 7.3: Case directory setup in OpenFOAM

[28] Note that the geometry and mesh generator supplied with OpenFOAM (`blockMesh`) can also be used. However, due to its limited applications and uneasy usage it has been avoided.

[29] OpenFOAM provides mesh conversion utilities, to convert meshes from popular formats to OpenFOAM compatible format (i.e. gmshToFoam, fluentMeshToFoam, starToFoam, cfxToFoam etc.) [126].

[30] Ideally, the <*case*> directory should be named such that it corresponds to the test case being solved.

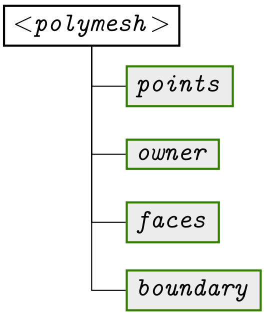


FIGURE 7.4: Polymesh directory structure after mesh conversion

Fig. 7.3. The *0* folder corresponds to zero time directory and it contains all the information pertaining to initial and boundary conditions for the associated variables. The ***constant*** folder contains data which remains constant throughout the simulation. The <*case*>/***constant*** directory contains ***polymesh*** folder which includes complete mesh information. Since the mesh generated hasn't been hasn't been converted to OpenFOAM format, the ***polymesh*** folder is therefore empty at this instance. Furthermore, the ***constant*** folder may contain some configuration files which are known as `dictionaries` (highlighted with a grey background in Fig. 7.3) in OpenFOAM terminology. These `dictionaries` are simply text files which provide the required data structure for an OpenFOAM solver. One of these dictionaries, which contains inputs for the implemented solid solver is the *simulationParameters* configuration file. Details of this input file will be discussed in Section 7.2.2. Furthermore, the ***system*** directory contains other data relevant to numerical methodology of the solver. It contains the *controlDict* file, which includes all inputs related to time control of simulation and reading/writing of solution data [126]. The configuration files *fvSchemes* and *fvSolution* specify the finite volume discretisation schemes and solver controls, respectively [126]. Moreover, for parallel simulations, ***system*** directory should contain the *decomposeParDict* dictionary, which contains inputs for decomposition of the computational domain. Finally, the <*case*> directory also contains the **.msh* file that was created earlier using a suitable mesh generator.

After setting up the <*case*> directory, the next step is to convert the **.msh* file into an OpenFOAM readable format. For the Gmsh mesh file, this is achieved through the command `<< gmshToFoam *.msh`, whereas the Fluent mesh file can be converted using `<< fluentMeshToFoam *.msh`. Once conversion is performed, a list of files are generated inside the ***polymesh*** folder as shown in Fig. 7.4. The *points* file contains a list of vectors corresponding to the nodal coordinates of the mesh generated. The *faces* files includes connectivity of each face with the node labels as a list of lists. In the *owner* file, owner cell labels are specified for all faces[31]. All information related to boundary of the domain is included in the *boundary* file.

Once the mesh is generated, it is crucial to check if the conversion process was successful. This can be achieved by invoking the command `<< checkMesh` inside the <*case*> directory. The resulting output of this `checkMesh` utility, shows important mesh parameters including cell

[31] Note that a face always has two cells attached to it, one owner and the other neighbour, except for the boundary face which is always attached to one cell, it's owner.

aspect ratios, mesh non-orthogonality, mesh skewness etc as shown in Listing C.2. At the end of this output, OpenFOAM gives it's verdict if the mesh is acceptable for use [32].

7.2.2 Solving

In order to obtain solution to the problem, relevant inputs for all dictionaries mentioned in Fig. 7.3 must be given according to the problem being simulated. The inputs for the ***simulationParameters*** file are shown in Listing 7.1. Listing 7.2 provides a list of test cases which have been simulated as part of this thesis and are an input to Fig. 7.3. Once the case has been set-up, solid mechanics solver can be invoked by executing the command `<< mixedSolidFoam` in the <***case***> directory. As the solver runs, it's output is displayed on the terminal window. Moreover, results are also written in the <***case***> directory inside corresponding time directory. The frequency writing data depends on the inputs given in the ***controlDict*** file.

The main source code of the solid mechanics solver, implemented from scratch in OpenFOAM, is shown in Listing 7.3 along with a couple of major header files ***governingEqns.H*** (see Listing 7.5) and ***updateVariables.H*** (see Listing 7.4).

[32] Note that this analysis is based on a loose mesh quality criterion and shouldn't be blindly trusted if the mesh is termed acceptable for usage by OpenFOAM.

```
/*-------------------------------------*- C++ -*----------------------------------------*\
| =========                      |                                                       |
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox                 |
|  \\    /   O peration          | Version:  2.3.0                                       |
|   \\  /    A nd                | Web:      www.OpenFOAM.org                            |
|    \\/     M anipulation       |                                                       |
\*--------------------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      simulationParameters;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

testCase testCaseName;                  // Insert "testCaseName" name from Listing 7.2

problemType nonLinear;                  // Options: {linear}
constitutiveModel neoHookean;           // Options: {hyperElasticPlastic, polyConvexNeoHookean}

Jlaw no;                                // J conservation law
Hlaw no;                                // H conservation law
Elaw no;                                // E conservation law

FVM C-TOUCH;                            // Options: {P-TOUCH, X-GLACE}

if [ $FVM == P-TOUCH ]; then
    xi_F 0.1;                           // Penalisation factor for F equation
    xi_J 0.1;                           // Penalisation factor for J equation
    xi_H 0.1;                           // Penalisation factor for H equation
fi

beta 1.0;                               // Parameter for preconditioned dissipation in RS

enhancedGradient yes;                   // Useful in the presence of fixed boundaries
limiter no;                             // Slope limiter for cell gradients
reconstruction linear;                  // Options: {constant}

riemannWaveSpeeds uniform;              // Options: {nonuniform}
timeIntegration twoStepRK;              // Options: forwardEuler
angularMomentumPreservation yes;        // Options: {no}

rho 1100;                               // Density (kg/m^3)
nu 0.45;                                // Poisson’s ratio
E 17e6;                                 // Young’s modulus of elasticity (Pa)

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Listing 7.1: Input file

```
        Test Case Description                           TestCaseName

    (A) Shock Scenario
        - Elastic cable (Step loading)          --->    elasticCableStep

    (B) Mesh Convergence
        - Elastic cable (Sinusoidal loading)    --->    elasticCableSinusoidal
        - lowDispersionCube                     --->    Low dispersion cube

    (C) Momentum Preservation
        - Spinning plate                        --->    spinningPlate
        - L-shaped block                        --->    lShapedBlock
        - Satellite-like structure              --->    satelliteStructure

    (D) Locking
        - Bending column                        --->    bendingColumn
        - Cook cantilver                        --->    cookCantilver
        - Twisting column                       --->    twistingColumn

    (E) Von Mises Plasticity
        - Taylor impact                         --->    taylorImpact
        - Tensile test                          --->    tensileTest

    (F) Contact Problems
        - Ring impact                           --->    ringImpact
        - Bar rebound                           --->    barRebound
        - Torus impact                          --->    torusImpact

    (G) Algorithm Robustness
        - Complex twisting column               --->    complexTwisting
        - Punch test                            --->    punchTest
        - Stent-like structure                  --->    stent
```

Listing 7.2: Summary of test cases used in this study

```
/*------------------------------------------------------------------------------------*\
  =========                   |
  \\      /   F ield          |   OpenFOAM: The Open Source CFD Toolbox
   \\    /    O peration      |
    \\  /     A nd            |   Copyright (C) 2011 OpenFOAM Foundation
     \\/      M anipulation   |
----------------------------------------------------------------------------------------
License
    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.

Application
    mixedSolidFoam

Description
    A large strain solid mechanics solver based on a linear momentum/strains
    mixed formualtion. An explicit Total Lagrangian formulation utilisiing
    a monolithic Total Variation Diminishing Runge-Kutta time integrator.
    A discrete angular momentum projection algorithm based on two global
    Lagrange Multipliers is added for angular momentum conservation.

\*-----------------------------------------------------------------------------------*/

// INCLUSION OF RELEVANT HEADER FILES
#include "fvCFD.H"
#include "pointFields.H"
#include "gradientSchemes.H"
#include "operators.H"

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

// MAIN PROGRAM BEGINS
int main(int argc, char *argv[])
{
    // HEADER FILES
    #include "setRootCase.H"            // Set path and case directories
    #include "createTime.H"             // Initialise time variable
    #include "createMesh.H"             // Generate mesh for the problem
    #include "simParameters.H"          // Read simulation parameters
    #include "meshData.H"               // Create variables based on mesh
    #include "createFields.H"           // Generate problem variables
    #include "initialConditions.H"      // Specify initial conditions (ICs)
    #include "updateVariables.H"        // Update variables based on ICs
    #include "deltaT.H"                 // Calculate time increment
    #include "output.H"                 // Write results at time zero

    // TIME LOOP
    while (runTime.loop())
    {
        // Calculate time and time step (TS)
        t += deltaT;                    // Time
        tstep++;                        // Time step

        // Solve conservation variables for first Runge-Kutta stage
        #include "governingEqns.H"

        // Update problem variables
        #include "updateVariables.H"

        // Solve conservation variables for second Ringe-Kutta stage
        #include "governingEqns.H"

        // Calculate conservation variables for the time step
        p = 0.5 * (p.oldTime()+p);          // Linear momentum
        F = 0.5 * (F.oldTime()+F);          // Deformation gradient
        H = 0.5 * (H.oldTime()+H);          // Cofactor of deformation
        J = 0.5 * (J.oldTime()+J);          // Jacobian of deformation
        x = 0.5 * (x.oldTime()+x);          // Cell center coordinates
        x_a = 0.5 * (x_a.oldTime()+x_a);    // Nodal coordinates
        x_f = 0.5 * (x_f.oldTime()+x_f);    // Face center coordinates
        E = 0.5 * (E.oldTime()+E);          // Total energy

        #include "updateVariables.H"        // Update variables
        #include "postProcessing.H"         // Calculate results
        #include "output.H"                 // Write results at time step
        #include "deltaT.H"                 // Calculate time increment for next TS
    }

    Info << "\nExecutionTime = " << runTime.elapsedCpuTime() << " s" << "  ClockTime = " << runTime.elapsedClockTime() << " s" << nl << endl;
    Info << "\nEnd\n" << endl;
    return 0;

}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Listing 7.3: Main source file of the solid mechanics solver

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#include "stress.H"                         // Compute PK1 stresses

if ( reconstruction == "linear" )
{
    #include "gradients.H"                  // Least square gradient calculation
    #include "reconstruction.H"             // Linear reconstruction procedure
}

#include "acousticRiemannSolver.H"          // Calculate contact fluxes
#include "nodalLinearMomentum.H"            // Obtain nodal linear momentum pN
#include "strongBoundaryConditions.H"       // Impose strong boundary conditions on pN

if ( finiteVolumeMethod == "C-TOUCH" || finiteVolumeMethod == "P-TOUCH" )
{
    interpolate.pointToSurface(pN, pC_tilde);
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Listing 7.4: updateVariables.H

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
// Update cell linear momentum
p += deltaT * rhs_p;

// Update cell deformation gradient
if ( finiteVolumeMethod == "P-TOUCH" )
{
    F += deltaT * rhs_F + (xi_F * (F_X - F));
}
else if ( finiteVolumeMethod == "C-TOUCH" || finiteVolumeMethod == "X-GLACE" )
{
    F += deltaT * rhs_F;
}

// Update cell cofactor of deformation
if ( Hlaw == "yes" )
{
    if ( finiteVolumeMethod == "P-TOUCH" )
    {
        H += deltaT * rhs_H + (xi_H * (H_F - H));
    }
    else if ( finiteVolumeMethod == "C-TOUCH" || finiteVolumeMethod == "X-GLACE" )
    {
        H += deltaT * rhs_H;
    }
}

// Update cell Jacobian of deformation
if ( Jlaw == "yes" )
{
    if ( finiteVolumeMethod == "P-TOUCH" )
    {
        J += deltaT * rhs_J + (xi_J * (J_F - J));
    }
    else if ( finiteVolumeMethod == "C-TOUCH" || finiteVolumeMethod == "X-GLACE")
    {
        J += deltaT * rhs_J;
    }
}

// Update cell total energy
if ( Elaw == "yes" )
{
    energy += deltaT * rhs_E;
}

x   += deltaT * (p/rho);                    // Update cell center coordinates
x_a += deltaT * (p_a/rho);                  // Update nodal coordinates
x_f += deltaT * (pC_tilde/rho);             // Update face coordinates

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Listing 7.5: governingEqns.H

7.2.3 Post-processing

Post-processing of results in this thesis has been carried out using ParaView[33], an open-source, multi-platform data analysis and visualization application software. OpenFOAM comes equipped with a post-processing utility known as `paraFoam` that reads OpenFOAM results into ParaView[34]. Post-processing can be performed during or after completion of the simulation through the command ***`paraFoam`***, executed in the <***`case`***> directory. Alternatively, user has the option to convert OpenFOAM data to VTK format, particularly useful for the analysis of surface fields, using the `foamToVTK` post-processing utility and then manually reading the results in ParaView.

[33] ParaView is a registered trademark of Kitware [129].

[34] Since version 3.14, Paraview has a built-in native reader for OpenFOAM data.

Chapter 8

BENCHMARK TESTS

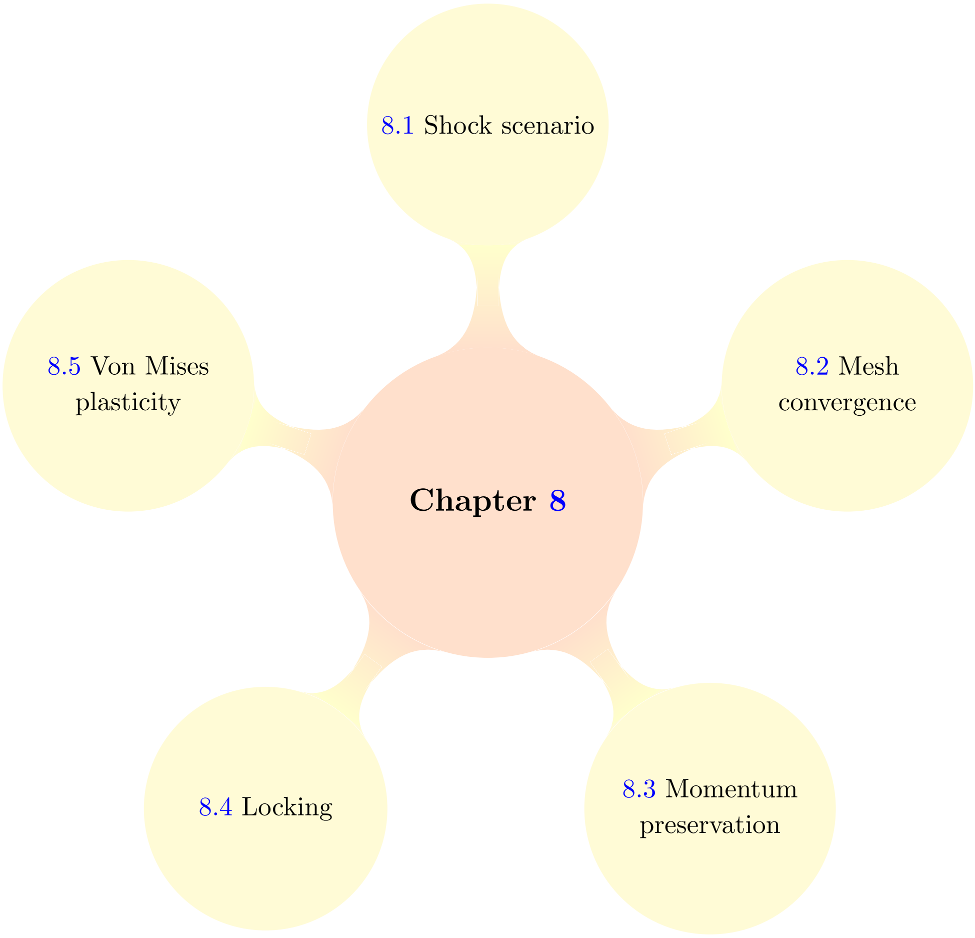


Figure 8.1: Structure of Chapter 8.

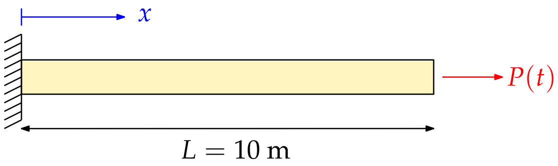


FIGURE 8.2: Elastic cable: Problem setup

In this chapter, a series of benchmark numerical examples is presented to access the convergence characteristics, momentum preservation properties and locking-free nature of the proposed $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ cell centred finite volume schemes, namely; (a) C-TOUCH (see Section 5.2); (b) P-TOUCH (see Section 5.3); and (c) X-GLACE (see Section 4.4). Notice that even though the total energy E of the system is computed as an additional conservation variable (for dissipation monitoring purposes), it is not coupled with the rest of conservation variables. All examples are simulated using the proposed solid mechanics solver (see Chapter 7), implemented from scratch in the open source software package OpenFOAM [130] in a three dimensional space [35]. It must be noted that for simplicity, body forces have been neglected in the conservation of linear momentum for all examples.

Moreover, for comparison purposes, some results are benchmarked against the well-known B-bar method [24] as well as a comprehensive library of alternative mixed numerical methodologies developed at Swansea University. Specifically, the cell centred FVM results are compared with the $\{\boldsymbol{p}, \boldsymbol{F}\}$ Upwind Vertex Centred [11] and the $\{\boldsymbol{p}, \boldsymbol{F}\}$ Jameson-Schmidt-Turkel Vertex Centred [10] finite volume methodologies. In addition, a $\{\boldsymbol{p}, \boldsymbol{F}, J\}$ Petrov-Galerkin Finite Element Method (PG-FEM) will also be used, where the extra conservation variable J is proven to be necessary, in terms of robustness, in nearly incompressible simulations [14]. Finally, a very sophisticated LBB compliant Hu-Washizu $\{\boldsymbol{v}, \boldsymbol{\Sigma_F}, \boldsymbol{\Sigma_H}, \Sigma_J\}$ complementary mixed Finite Element formulation [94] will also be used for comparison purposes.

The chapter is outlined such that each section aims to deal with a specific aspect of numerics. These numerical examples demonstrate shock propagation phenomena (see Section 8.1), spatial convergence analysis (see Section 8.2), linear and angular momentum preservation (see Section 8.3), locking-free nature (see Section 8.4) and plasticity phenomena (see Section 8.5). The layout of this chapter is presented in Fig. 8.1.

8.1 Shock scenario

8.1.1 Elastic cable (Step loading)

In this example, we consider wave propagation in a one dimensional linear elastic cable under the influence of a shock (see Fig. 8.2). The $L = 10$ m long cable is fixed at one end ($x = 0$),

[35] All problems presented in Chapter 8 and Chapter 9 are simulated in a three dimensional space.

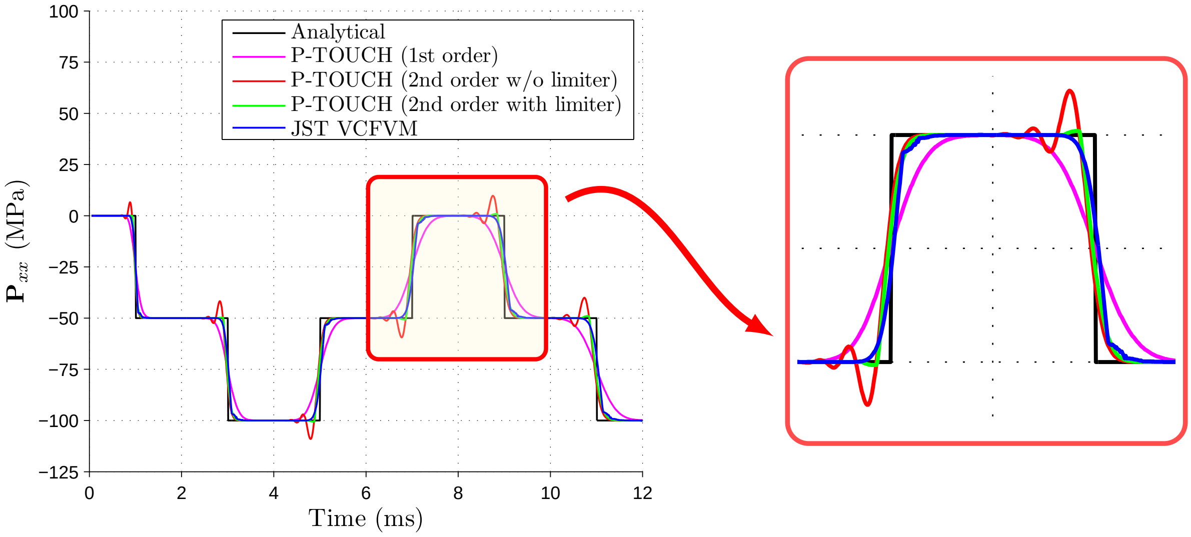


FIGURE 8.3: Elastic cable with step loading: Stress evolution at mid-bar with a shock load $\boldsymbol{t}_b = [-50, 0, 0]^T$ MPa using the standard finite volume update (P-TOUCH with $\xi_{\boldsymbol{F}} = 0$) compared against the analytical solution and JST vertex centred finite volume scheme. Results obtained using a linear elastic constitutive model with $\rho_0 = 8000$ kg/m^3, $E = 200$ GPa, $\nu = 0$, $\alpha_{\text{CFL}} = 0.5$ and $\varDelta t = 1 \times 10^{-5}$ s. Discretisation of $100 \times 1 \times 1$ hexahedral elements.

whilst a forcing function is applied at the other free end ($x = L$). Upon the application of external force, a stress wave propagates along the cable towards fixed end and then gets reflected back. The step traction loading applied, can be mathematically expressed as

$$\boldsymbol{t}_b\left(L, t\right) = \begin{cases} [0,0,0]^T & t < 0 \\ [-50,0,0]^T \;\; \text{MPa} & t \geq 0. \end{cases} \tag{8.1}$$

The material chosen for this problem has density $\rho_0 = 8000$ kg/m^3, Young's modulus $E = 200$ GPa and Poisson's ratio $\nu = 0$.

Fig. 8.3 shows the evolution of stress wave $\boldsymbol{P}_{xx}$ at the middle of the cable ($L = 5$ m). It can be clearly seen that the first order [36] P-TOUCH scheme ($\xi_{\boldsymbol{F}} = 0$) [37] introduces excessive numerical dissipation into the solution. To overcome this, a linear reconstruction procedure together with the use of the two-stage TVD-RK time integrator (see Section 6.2) is employed. Insofar as the scheme is not monotonicity preserving (without the use of a slope limiter), oscillations are observed in the vicinity of shock. These deficiencies can be eliminated to a great extent with the introduction of a slope limiter (see Section 3.4.2). For comparison purposes, the numerical solution obtained using the in-house JST-VCFVM [10] is also displayed.

[36] A piecewise constant reconstruction is used along with a forward Euler time integrator.

[37] For linear elastic material, satisfaction of involutions is guaranteed ab initio, therefore the standard finite volume update is sufficient.

8.2 Mesh convergence

8.2.1 Elastic cable (Sinusoidal loading)

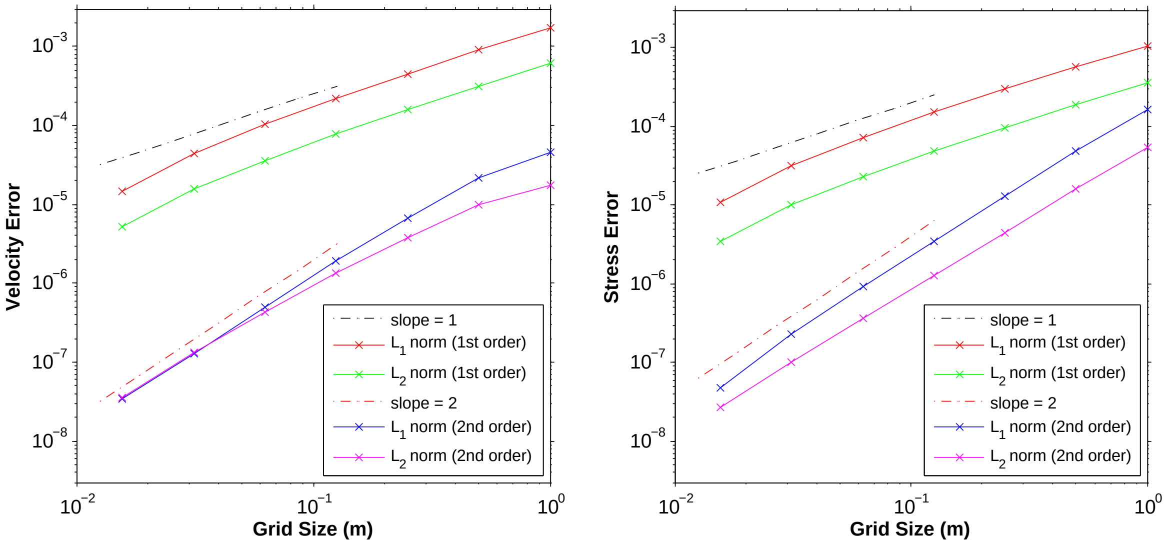
To assess the convergence behaviour of the proposed formulation, the same problem (see Section 8.1.1) is simulated here, but this time with the imposition of a smooth sinusoidal loading defined as

$$P(L,t) = \begin{cases} 0 & \text{t} < 0 \\ 0.001\left[\sin\left(\pi t/20 - \pi/2\right) + 1\right] \ \text{Pa} & \text{t} \geq 0. \end{cases} \tag{8.2}$$

The material parameters used are such that density $\rho_0 = 1$ kg/m^3, Young's modulus $E = 1$ Pa and Poisson's ratio $\nu = 0.3$. The mesh convergence analysis is shown in Fig. 8.4. It is clear that an optimal equal order of convergence for velocities and stresses can be obtained when using either first order or second order methodology. For problems where the accurate evaluation of the stresses is of paramount importance (i.e. onset of plastic yielding), this is certainly one of the greatest advantages of employing this mixed formulation in comparison with a displacement based approach, where stresses converge at a lower order of accuracy.

8.2.2 Low dispersion cube

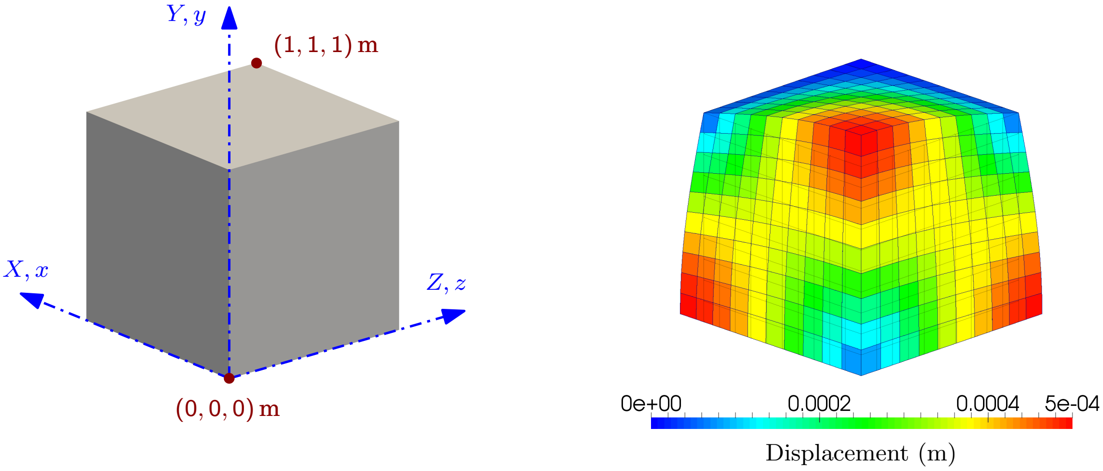
The objective of this example is to assess the spatial convergence behaviour of the proposed cell centred methodologies, namely C-TOUCH, P-TOUCH and X-GLACE. A unit cube is considered (see Fig. 8.5a) with symmetric boundary conditions (roller supports) at faces $X = 0$,



(a) Velocities

(b) Stresses

FIGURE 8.4: Elastic cable with sinusoidal loading: L_1 and L_2 norm convergence of (a) velocities; and (b) stresses at time $t = 34.4757\,$s using the P-TOUCH scheme ($\xi_{\boldsymbol{F}} = 0$). Results obtained using a linear elastic model with $\boldsymbol{t}_b(L,t) = 0.001\left[\sin\left(\pi t/20 - \pi/2\right) + 1, 0, 0\right]^T$ Pa, $\rho_0 = 1$ kg/m^3, $E = 1$ Pa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.5$.



(a) Configuration

(b) Initial deformed state and mesh

FIGURE 8.5: Low dispersion cube: Problem setup

$Y = 0$ and $Z = 0$ and skew-symmetric boundary conditions (restricted tangential displacements) at faces $X = 1\,\mathrm{m}$, $Y = 1\,\mathrm{m}$ and $Z = 1\,\mathrm{m}$. For small deformations, the problem has a closed-form displacement field of the form [6, 8, 10, 11, 14–16, 18, 19, 46, 131**?**]

$$\boldsymbol{u}(\boldsymbol{X}, t) = U_0 \cos\left(\frac{\sqrt{3}}{2} c_d \pi t\right) \begin{bmatrix} A \sin\left(\frac{\pi X_1}{2}\right) \cos\left(\frac{\pi X_2}{2}\right) \cos\left(\frac{\pi X_3}{2}\right) \\ B \cos\left(\frac{\pi X_1}{2}\right) \sin\left(\frac{\pi X_2}{2}\right) \cos\left(\frac{\pi X_3}{2}\right) \\ C \cos\left(\frac{\pi X_1}{2}\right) \cos\left(\frac{\pi X_2}{2}\right) \sin\left(\frac{\pi X_3}{2}\right) \end{bmatrix}; \quad c_d = \sqrt{\frac{\lambda + 2\mu}{\rho_0}}. \tag{8.3}$$

Parameters $\{A, B, C\}$ are user-defined arbitrary constants, chosen such that $A{=}B{=}C$ which ensure the existence of a non-zero pressure field [38]. For values of U_0 below 0.001 m, the solution can be considered to be linear and the closed-form expression (8.3) holds. The problem is initialised with the displacement field $\boldsymbol{u}_0 \equiv \boldsymbol{u}(\boldsymbol{X}, 0)$ according to Eq. (8.3) (see Fig. 8.5b) and subsequently, the initial deformation gradient, its co-factor and its Jacobian can be obtained as $\boldsymbol{F}_0 = \boldsymbol{I} + \boldsymbol{\nabla}_0 \boldsymbol{u}_0$, $\boldsymbol{H}_0 = \frac{1}{2} \boldsymbol{F}_0 \times \boldsymbol{F}_0$ and $J_0 = \frac{1}{6}(\boldsymbol{F}_0 \times \boldsymbol{F}_0) : \boldsymbol{F}_0$. A linear elastic material is chosen with a Poisson's ratio of $\nu = (1 - \mu/\kappa)/2 = 0.3$, Young's modulus $E = 1.7 \times 10^7$ Pa and density $\rho_0 = 1.1 \times 10^3\ \mathrm{kg/m^3}$. The solution parameters are set as $A{=}B{=}C{=}1$ and $U_0 = 5 \times 10^{-4}$ m.

Fig. 8.6 shows the time evolution of deformation of the cube (scaled 300 times) along with pressure distribution. An investigation into the convergence pattern of a $\{\boldsymbol{p}, \boldsymbol{F}\}$ formulation, when using a first order spatial and temporal discretisation, is carried out in Fig. 8.7. The L^1 and L^2 norm errors for linear momentum $\boldsymbol{p}$ and first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ clearly show that the C-TOUCH and P-TOUCH schemes are slightly more accurate than X-GLACE scheme. In Fig. 8.8 shows the expected second order convergence pattern when using a second order spatial and temporal discretisation as compared to the analytical solution obtained from Eq. (8.3).

[38] Relationship $A{+}B{+}C = 0$ and parameter $c_d = \sqrt{\frac{\mu}{\rho_0}}$ have been used in previous publications [10, 11, 14, 16, 17] leading to a non-volumetric deformation field.

Fig. 8.9 shows the expected second order convergence pattern (e.g. L^1 and L^2 norm errors) of the linear momentum $\boldsymbol{p}$ and the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ as compared to the closed-form solution described in Eq. (8.3) [39]. It can be clearly seen that the C-TOUCH and P-TOUCH schemes produce practically identical convergence patterns for both velocities and stresses. Crucially, their solutions are slightly more accurate than the results obtained from the X-GLACE scheme.

[39] Given the fact that A=B=C, all the three components of velocities and stresses are of the same magnitude. For instance, $\boldsymbol{v}_x$=$\boldsymbol{v}_y$=$\boldsymbol{v}_z$ and $\boldsymbol{P}_{xx}$=$\boldsymbol{P}_{yy}$=$\boldsymbol{P}_{zz}$.

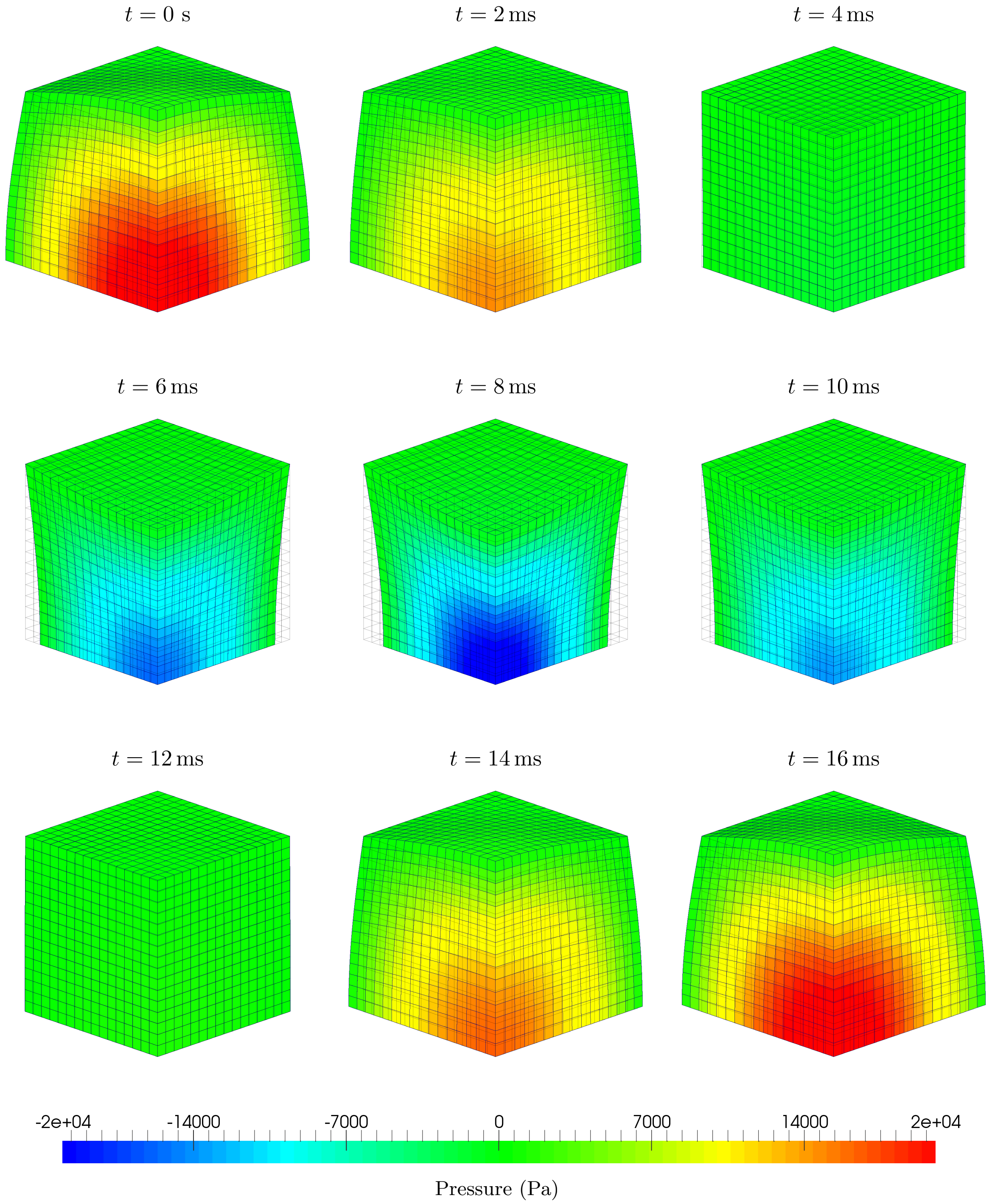
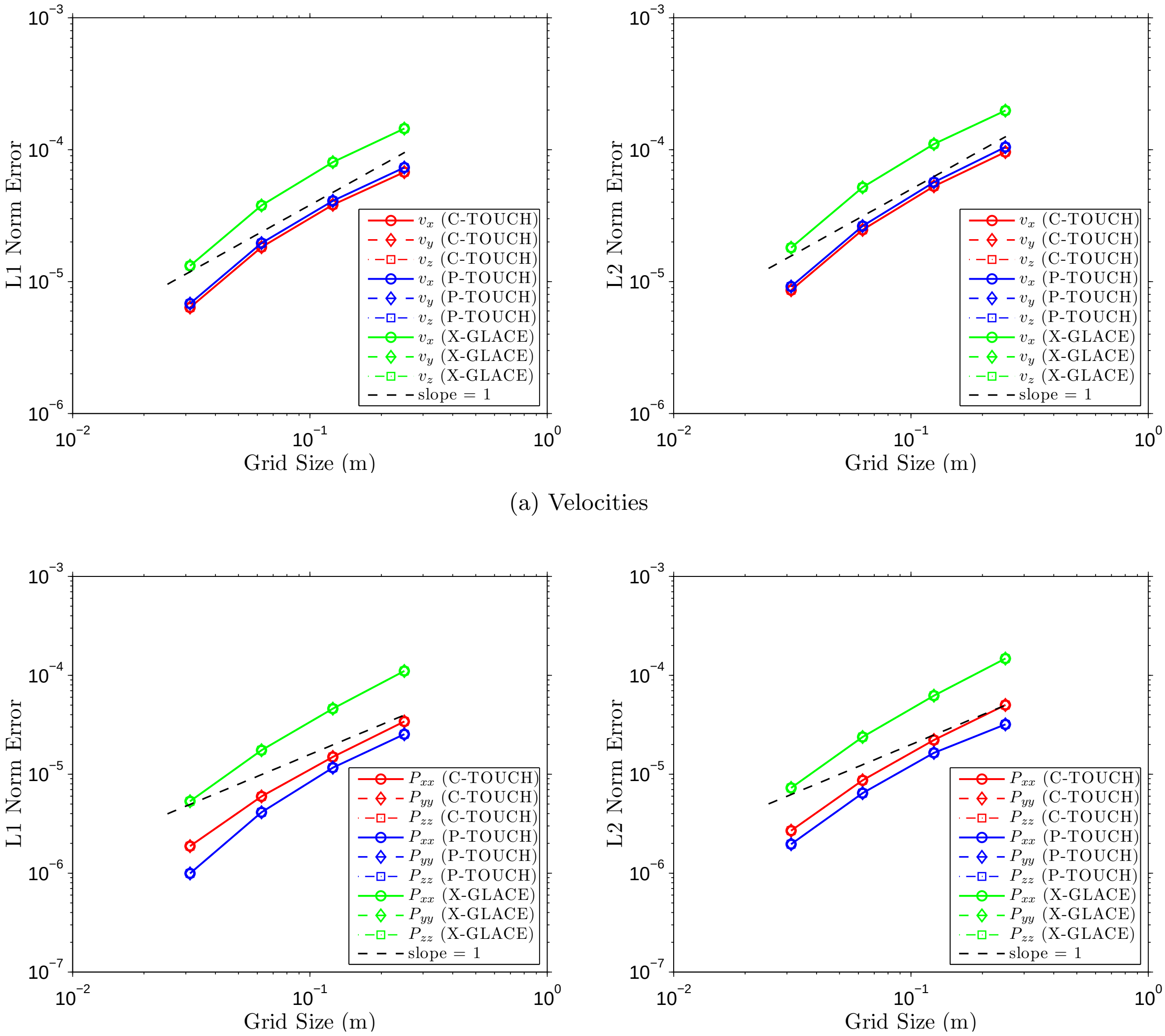


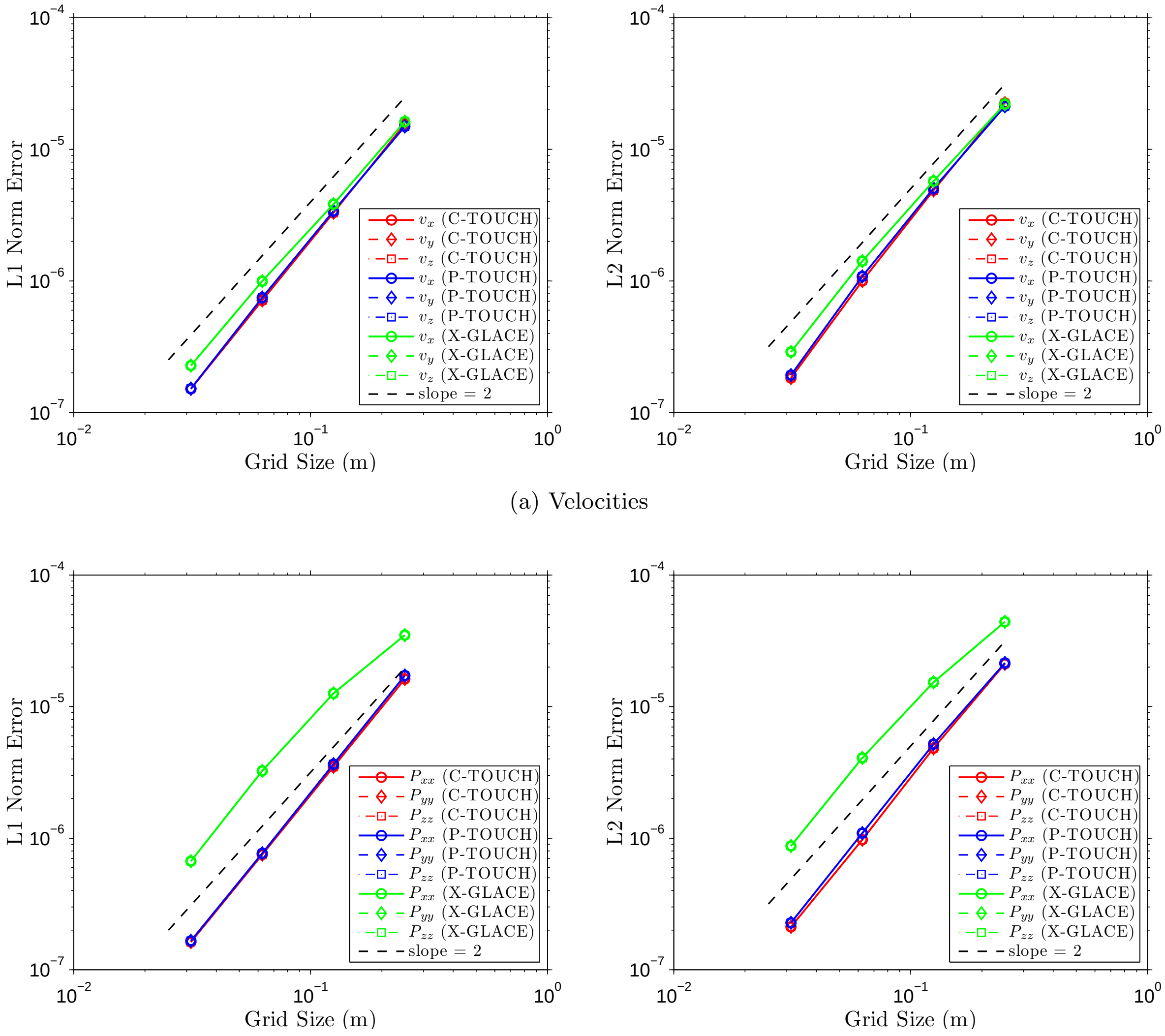
FIGURE 8.6: Low dispersion cube: Time evolution of deformation (scaled 300 times) plotted with pressure distribution. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with $A = B = C = 1$, $U_0 = 5 \times 10^{-4}$ and a discretisation of $16 \times 16 \times 16$ cells per edge. A linear elastic material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$.



(a) Velocities

(b) Stresses

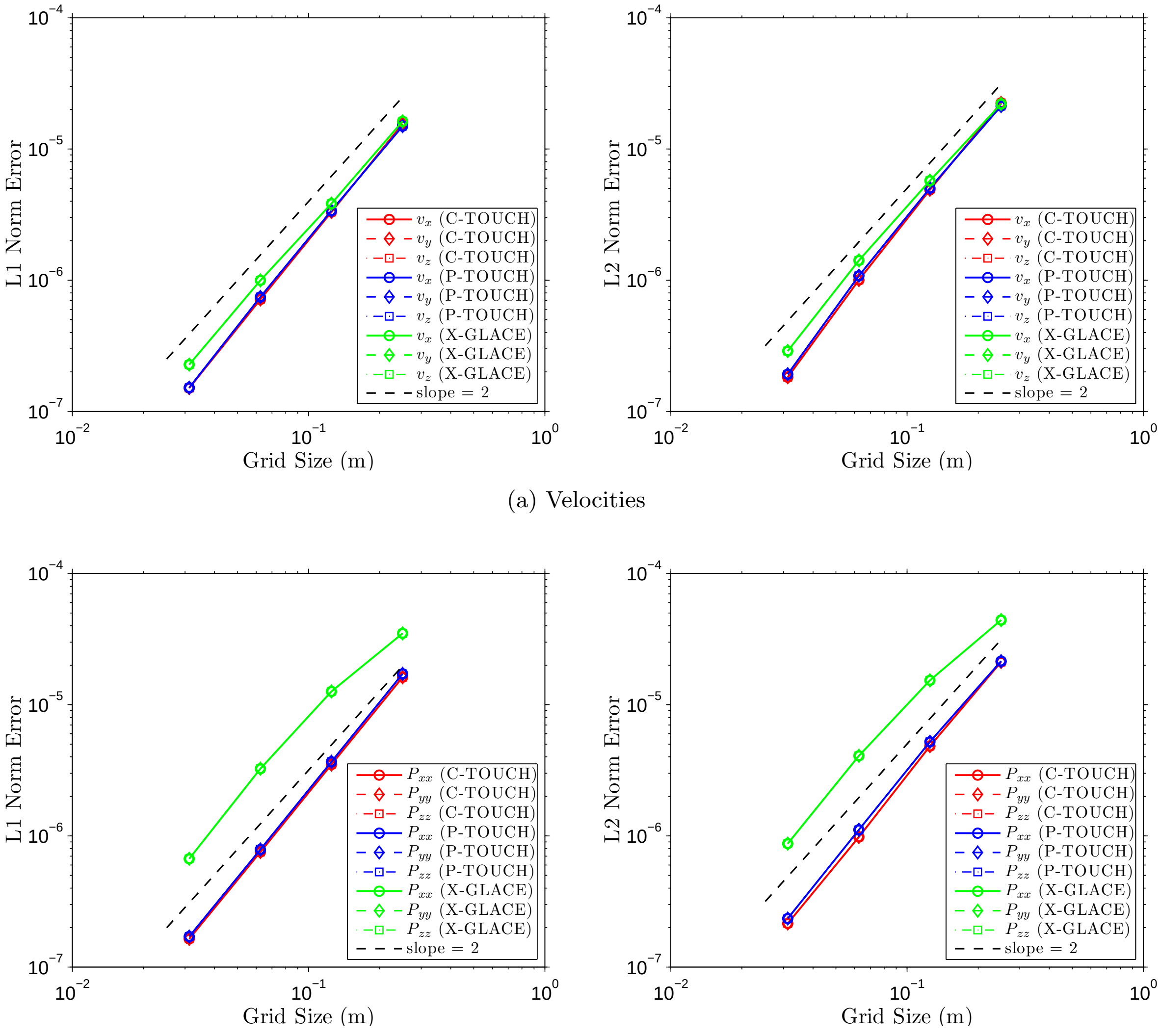
FIGURE 8.7: Low dispersion cube: L_1 and L_2 norm convergence of components of (a) velocities; and (b) stresses using first order $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH, P-TOUCH ($\xi_{\boldsymbol{F}} = 0$) and X-GLACE schemes in both space and time. Results with $A = B = C = 1$ and $U_0 = 5 \times 10^{-4}$ m at time $t = 4$ ms. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$.



(a) Velocities

(b) Stresses

FIGURE 8.8: Low dispersion cube: L_1 and L_2 norm convergence of components of (a) velocities; and (b) stresses using second order $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH, P-TOUCH ($\xi_{\boldsymbol{F}} = 0$) and X-GLACE schemes in both space and time. Results obtained with $A = B = C = 1$ and $U_0 = 5 \times 10^{-4}$ m at time $t = 4$ ms. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$.



(a) Velocities

(b) Stresses

FIGURE 8.9: Low dispersion cube: L_1 and L_2 norm convergence of components of (a) velocities; and (b) stresses using second order $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH, P-TOUCH ($\xi_{\{\boldsymbol{F},\boldsymbol{H},J\}} = 0$) and X-GLACE schemes in both space and time. Results obtained with $A = B = C = 1$ and $U_0 = 5 \times 10^{-4}$ m at time $t = 4$ ms. A polyconvex neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$.

8.3 Momentum preservation

8.3.1 Spinning plate

In this section, we consider a simple example of a spinning plate which is free on all sides [6, 13]. The unit square plate is initialised with a constant angular velocity of $\boldsymbol{\omega}_0 = [0, 0, 105]^T$ rad/s relative to the origin and is then left rotating in space [40] (see Fig. 8.10). A nearly incompressible neo-Hookean material is utilised with material properties such that density $\rho_0 = 1100$ kg/m^3, Youngs's modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.45$.

Fig. 8.11 shows a comparison of deformed shapes with pressure distribution at time $t = 0.15$ s for structured and unstructured meshes when using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. It can be clearly observed that the structured mesh with 400 elements (see Fig. 8.11a) compares very well the unstructured mesh comprising of 484 elements (see Fig. 8.11b) in terms of both pressure and deformation. Furthermore, Fig. 8.12 compares the time evolution of horizontal x and vertical y displacements of the corner point $\boldsymbol{X} = [0.5, 0.5, 0]^T$ obtained from meshes used in Fig. 8.11. The excellent agreement of displacements proves that the proposed C-TOUCH methodology can reliably be applied to structured as well as unstructured meshes. Moreover, it is also interesting to monitor the momentum preserving characteristics of the C-TOUCH scheme when applied to both categories of meshes. Fig. 8.13a shows the evolution of components of global angular momentum in the case of structured and unstructured meshes. As expected only the out-of-plane z component of angular momentum is non-zero. Most importantly, the values of the components A_x, A_y and A_z remain constant as time evolves. In Fig. 8.13b the components of global linear momentum are also plotted to show that they are within machine accuracy, since no translational movement is expected for this problem.

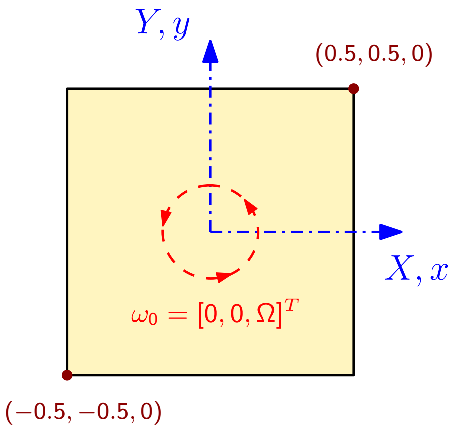
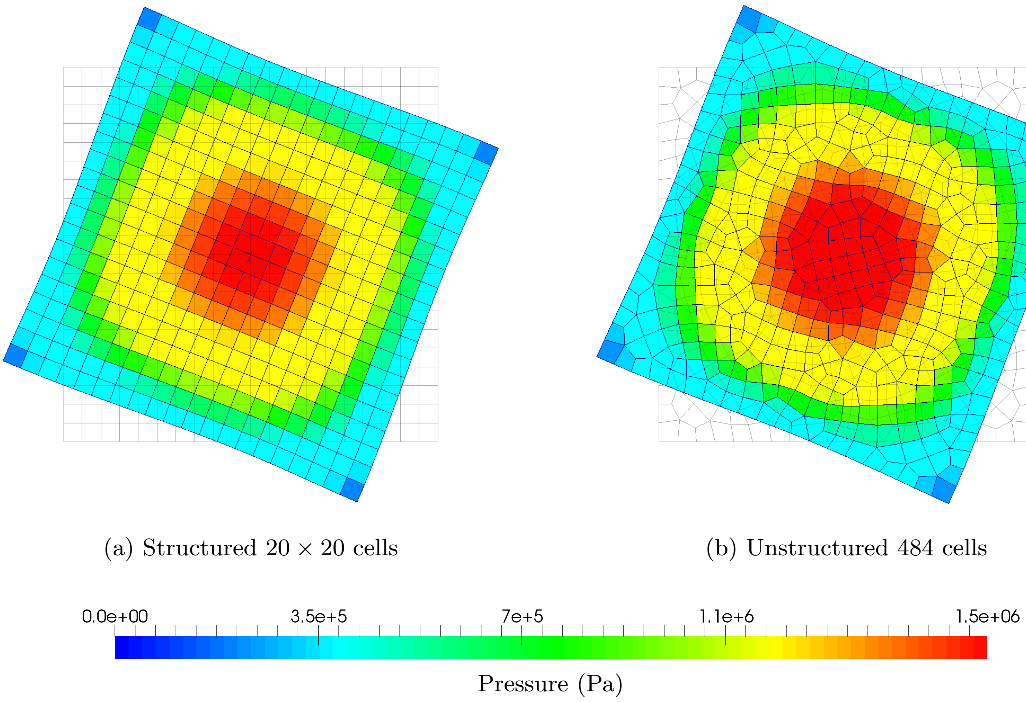


FIGURE 8.10: Spinning plate: Problem setup

[40] Note that there is no steady state solution to this problem.


(a) Structured 20 × 20 cells

(b) Unstructured 484 cells

FIGURE 8.11: Spinning plate: Comparison of deformed shapes along with pressure distribution at $t = 0.15$ s using (a) structured mesh (20 × 20 cells); and (b) unstructured mesh (484 cells). Results are obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with an angular velocity $\boldsymbol{\omega}_0 = [0, 0, 105]^T$ rad/s. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

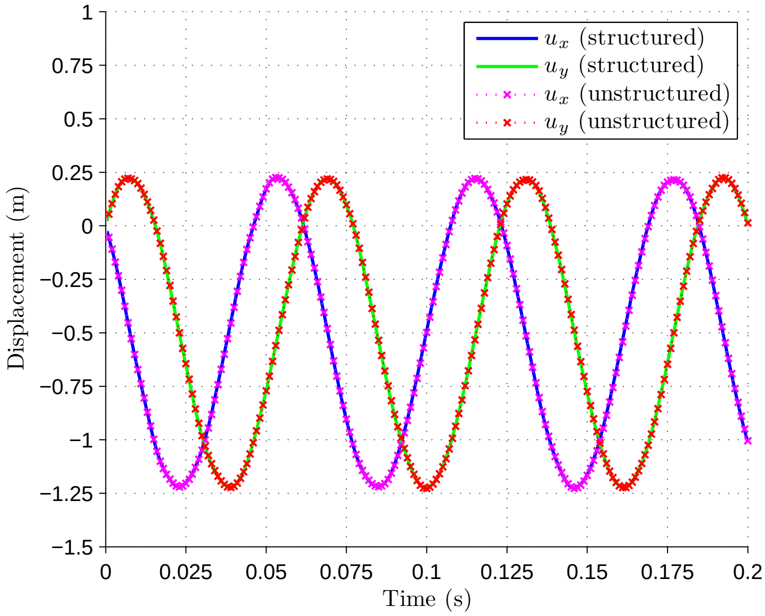
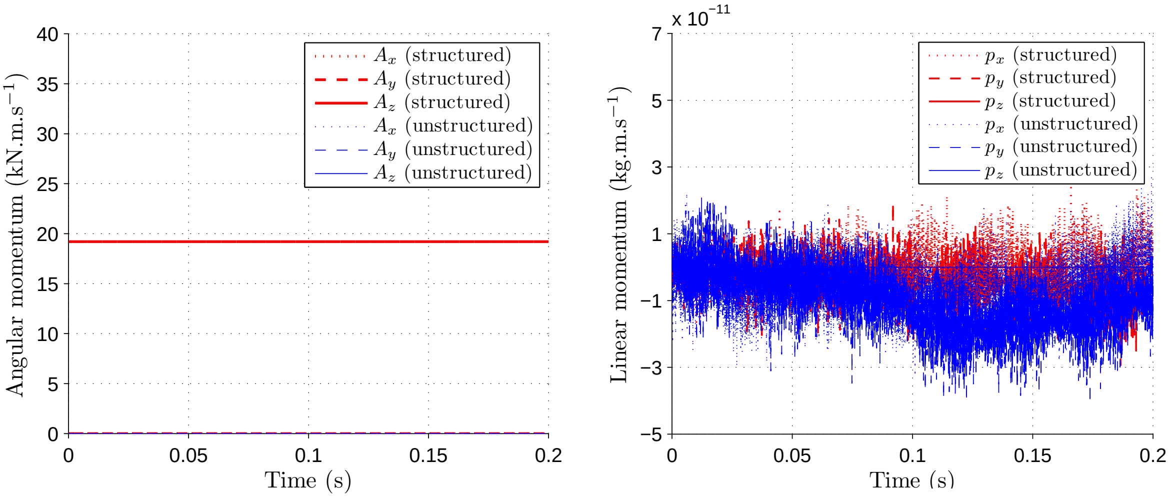


FIGURE 8.12: Spinning plate: Time evolution of horizontal u_x and vertical u_y displacements of the material point $\boldsymbol{X} = [0.5, 0.5, 0]^T$ m using structured (20 × 20 cells) and unstructured (484 cells) meshes. Results are obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with an angular velocity $\boldsymbol{\omega}_0 = [0, 0, 105]^T$ rad/s. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.



(a) Angular momentum

(b) Linear momentum

FIGURE 8.13: Spinning plate: Time evolution of the components of (a) global angular momentum; and (b) global linear momentum with structured (20×20 cells) and unstructured (484 cells) meshes. Results are obtained with an angular velocity $\boldsymbol{\omega}_0 = [0, 0, 105]^T$ rad/s using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

8.3.2 L-shaped block

This example presents an L-shaped block (see Fig. 8.14), first introduced by Simo and Tarnow [132] and then later explored in [10, 11, 16, 133–135]. The block is subjected to time varying forces on two of its sides according to

$$\boldsymbol{F}_1(t) = -\boldsymbol{F}_2(t) = [150, 300, 450]^T \begin{cases} t & 0 \le t < 2.5 \\ 5 - t & 2.5 \le t < 5 \\ 0 & t \ge 5. \end{cases}$$

After the removal of external forces, the block is left tumbling in space (free boundaries) suffering from finite deformations but with large overall rotations. A neo-Hookean material is considered with density $\rho_0 = 1000$ kg/m^3, Young's modulus $E = 50.05$ kPa and Poisson's ratio $\nu = 0.3$.

Fig. 8.15 depicts the sequence of the deformation process showing the domination of rotational forces along with pressure distribution when using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. In Fig. 8.16, a mesh refinement study is carried out at time $t = 7.5$ s by refining the mesh in all (x, y and z directions). As the mesh is refined very similar deformation is obtained, however, the pressure is captured more accurately. Furthermore, in Fig. 8.17 the mesh refinement is only carried out in the z direction leading elements with high aspect ratios. It can be clearly seen that the use of a high aspect ratio $r = 4$ (see Fig. 8.17c) leads to a very similar deformation and pressure distribution. Fig. 8.18a shows the time evolution of the components of global angular momentum with and without the consideration of the angular momentum projection algorithm

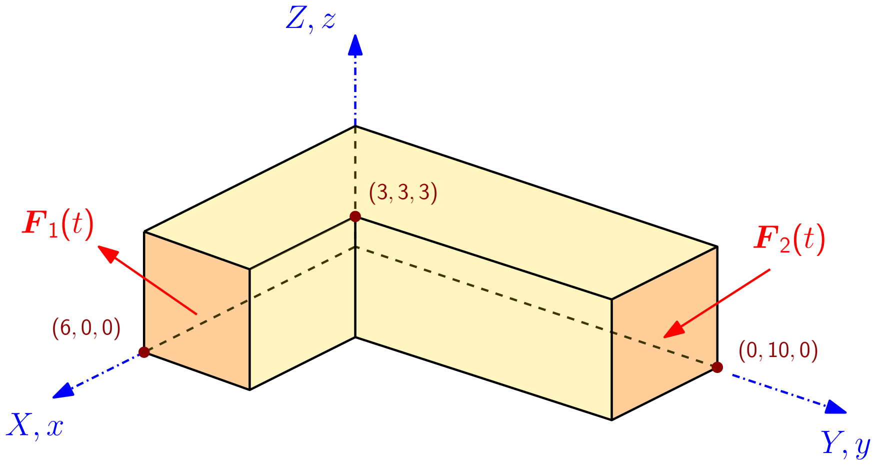


FIGURE 8.14: L-shaped block: Problem setup.

(see Section 6.3). As can be observed, without the use of the projection algorithm[41], the block rapidly decelerates until it reaches a standstill. This is in clear contrast to the perfectly captured behaviour of the block which correctly exhibits no change in angular momentum (after the external loading ceases) when the projection algorithm is employed. Moreover, Fig. 8.18b illustrates the conservation of components of global linear momentum, which is zero to machine accuracy. It is worthwhile to highlight that the amount of numerical dissipation introduced by the proposed algorithm can be accurately measured by considering the total energy conservation equation (see Section 2.4.5). As expected and shown in Fig. 8.19, the amount of numerical dissipation decreases with the reduction in mesh size.

8.3.3 Satellite-like structure

In this section, motion of a satellite-like structure, similar to the one presented in [132], is studied in order to demonstrate the momentum conservation characteristics of the proposed formulation. The structure comprises of (a) cylindrical central section of radius 1.5 m and height 3 m; and (b) four arms of cross-section 1 × 1 m^2 that extend 6.5 m from the center of the structure (see Fig. 8.20a). The problem is initialised with an initial angular velocity field $\boldsymbol{\omega}_0 = [0,0,1]^T$ rad/s resulting in an initial velocity distribution shown in Fig. 8.20b. A polyconvex constitutive model is chosen with the material parameters such that $\rho_0 = 1000$ kg/m^3, Young's modulus $E = 50.05$ kPa and Poisson's ratio $\nu = 0.3$.

The time evolution of deformation along with a smooth pressure distribution is shown in Fig. 8.21 when using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. A mesh refinement study is also carried out in Fig. 8.22 at time $t = 5.5\,$s using meshes comprising of 4032, 16960 and 51336 hexahedral elements. The use of fine mesh produces very similar deformation in comparison with a coarse mesh, but clearly with better pressure representation near the bending region of the four arms.

[41] The results obtained without employing the angular momentum projection algorithm have been simulated using a first order spatial discretisation in order to amplify the difference.

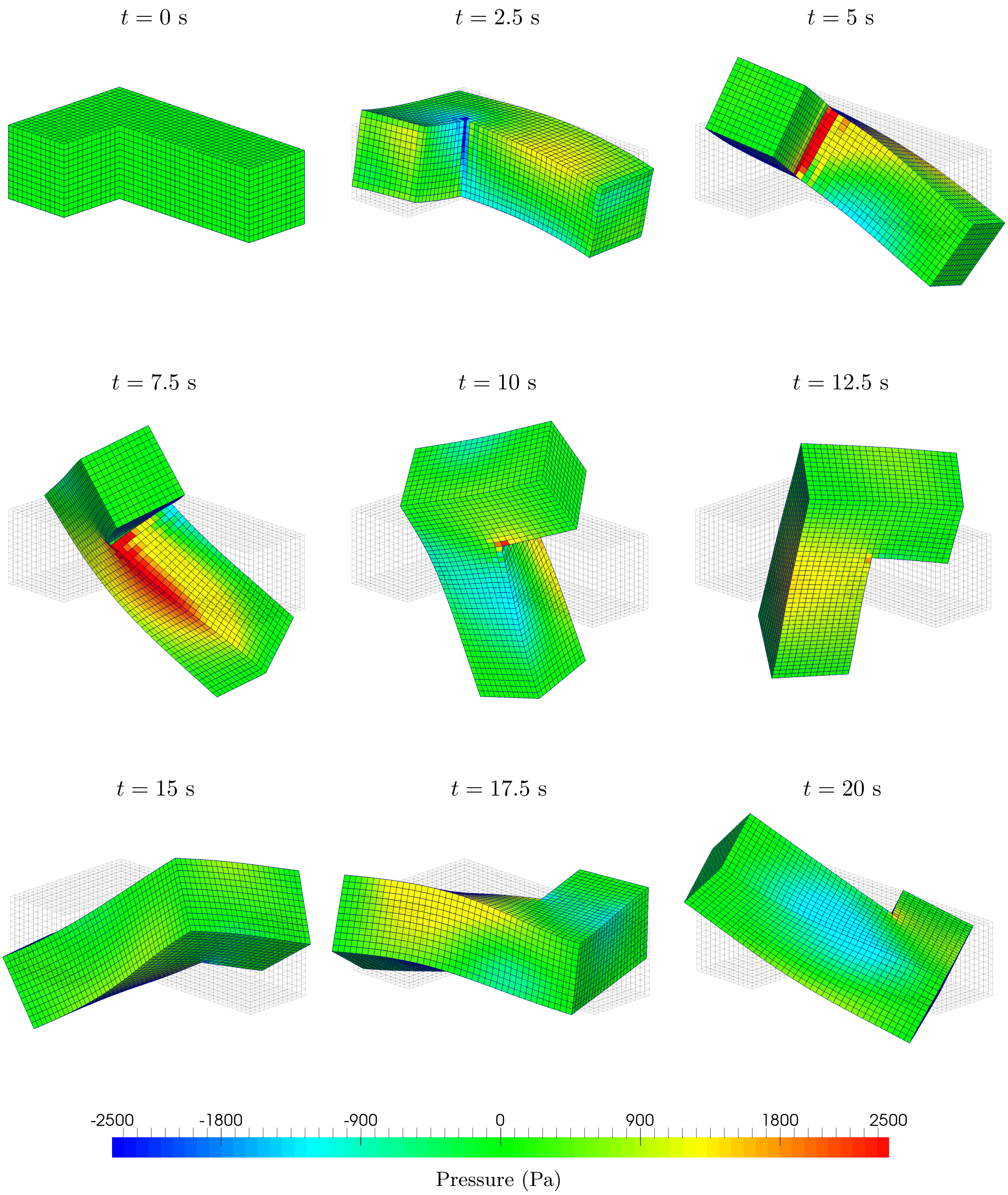
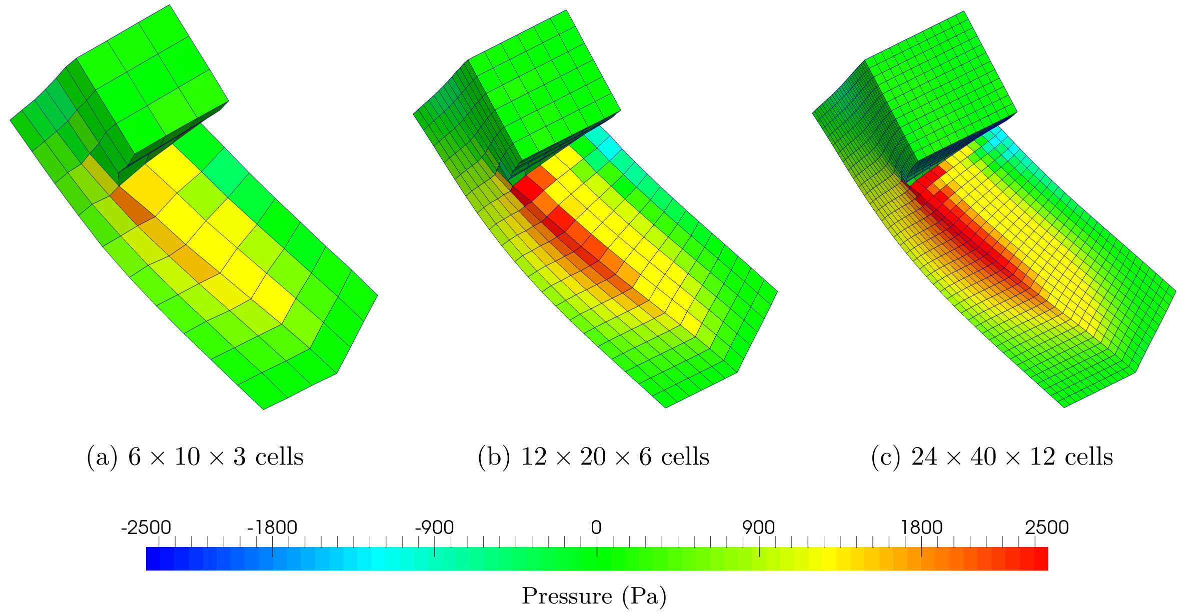


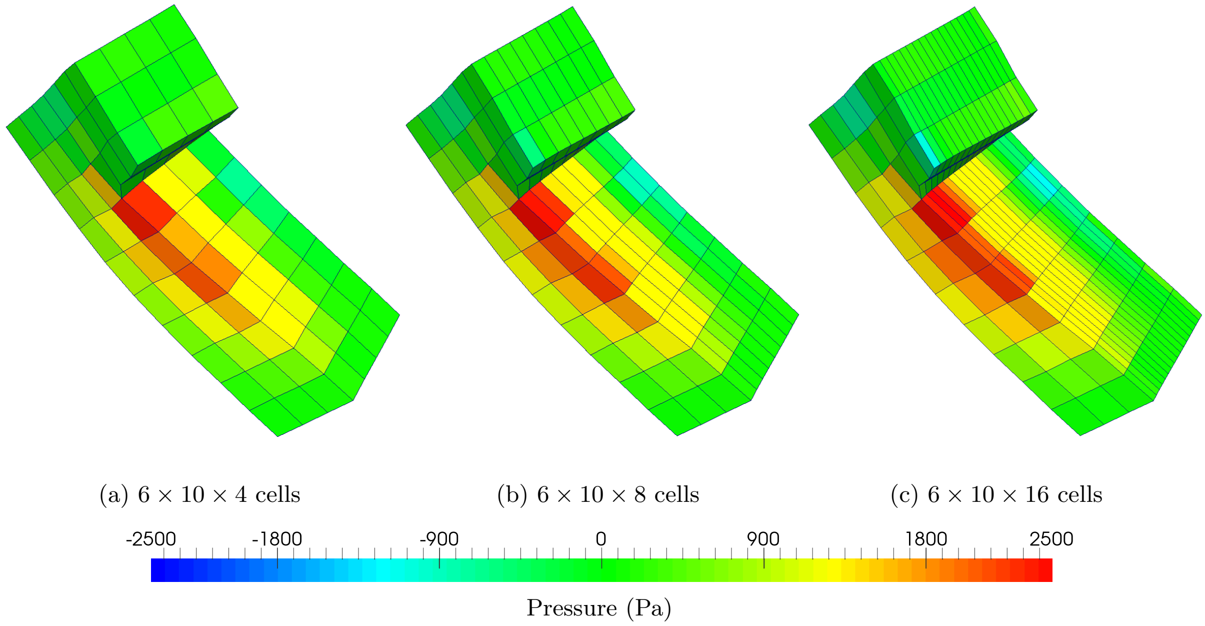
FIGURE 8.15: L-shaped block: Time evolution of the deformation plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. Results obtained with impulse boundary conditions on two of the sides and a discretisation of 24 × 40 × 12 hexahedral elements. A neo-Hookean material is used with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$, $\alpha_{\text{CFL}} = 0.3$ and $\Delta t \approx 0.008$ s.


(a) 6 × 10 × 3 cells

(b) 12 × 20 × 6 cells

(c) 24 × 40 × 12 cells

FIGURE 8.16: L-shaped block: Mesh refinement of deformed shapes plotted along with pressure distribution for structured cube elements at $t = 7.5$ s for three different mesh sizes: (a) $h = 1$ m; (b) $h = 1/2$ m; and (c) $h = 1/4$ m. Results are obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH algorithm with impulse boundary conditions applied on two of its sides. A neo-Hookean material is used with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$.


(a) 6 × 10 × 4 cells

(b) 6 × 10 × 8 cells

(c) 6 × 10 × 16 cells

FIGURE 8.17: L-shaped block: Mesh refinement along the z axis plotted with pressure distribution for structured cuboid elements with various aspect ratios r: (a) $r = 4/3$; (b) $r = 8/3$; and (c) $r = 4$. Results are obtained at $t = 7.5$ s using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with impulse boundary conditions applied on two of its sides. A neo-Hookean material is used with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$.

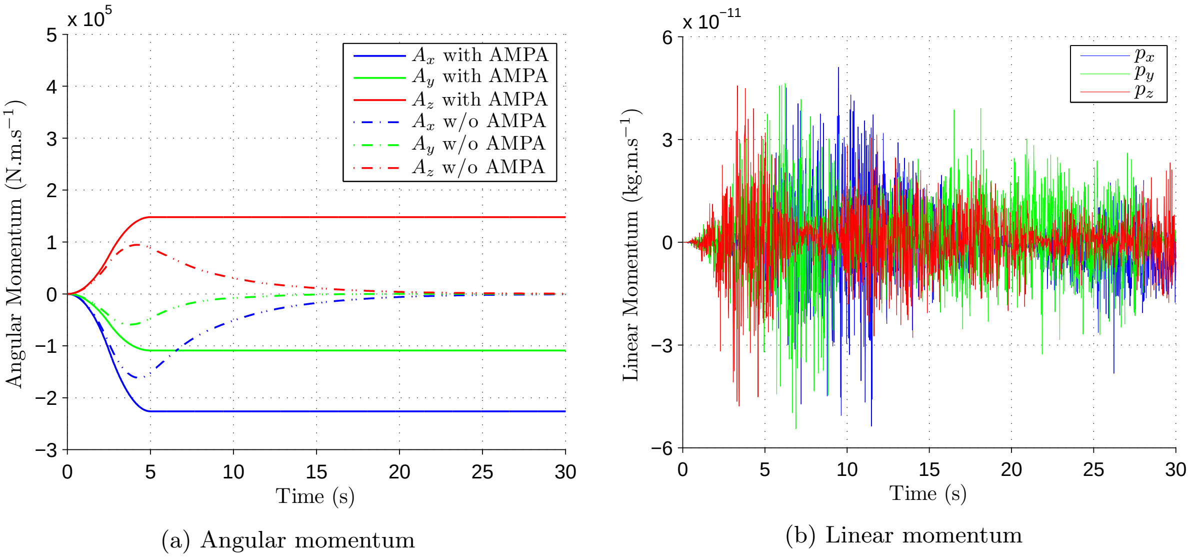


FIGURE 8.18: L-shaped block: Time evolution of the components of (a) global angular momentum with and without the consideration of discrete angular momentum projection algorithm; and (b) global linear momentum. Results are obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with a discretisation of $12 \times 20 \times 6$ hexahedral elements by imposing impulse boundary conditions on two of its sides. A hyperelastic neo-Hookean constitutive model is utilised with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$

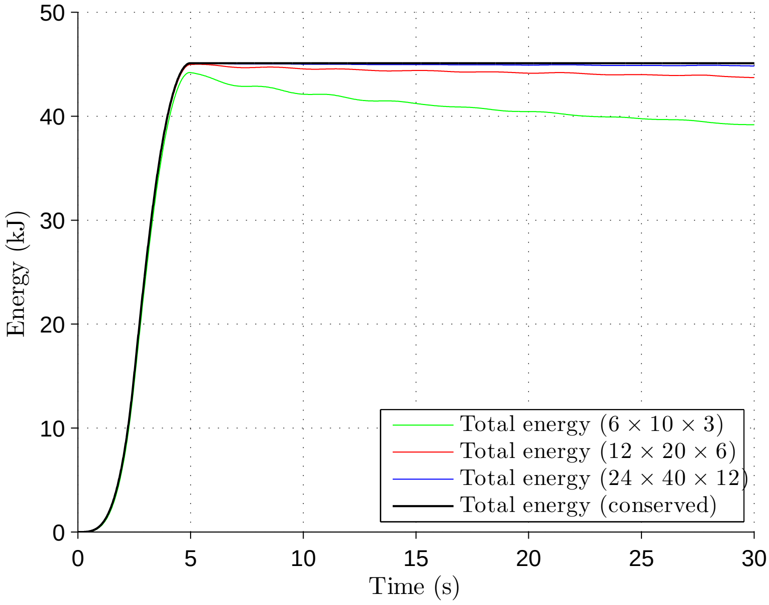
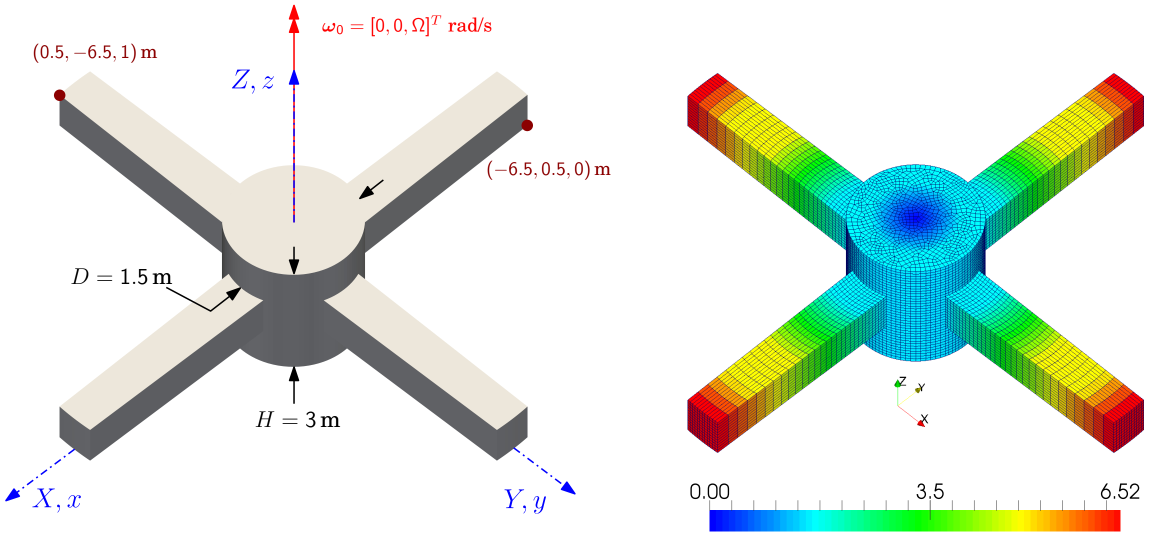


FIGURE 8.19: L-shaped block: Numerical dissipation of the proposed $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH algorithm discretised using three different mesh sizes. Results are obtained with the impulse boundary conditions applied on two of its sides. A hyperelastic neo-Hookean constitutive model is used with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$ and $\alpha_{\text{CFL}} = 0.3$.



(a) Initial configuration

(b) Initial velocity in X-Y plane (m/s)

Figure 8.20: Satellite-like structure: Problem setup

Figs. 8.23 and 8.24 show the time history of the components of global linear and angular momentum within the system, simulated using C-TOUCH and X-GLACE respectively. With the use of angular momentum projection algorithm (see Section 6.3), as expected, the global linear momentum fluctuates around zero machine accuracy, whereas the global angular momentum is constantly conserved after long term response. Otherwise, significant reduction in momenta can be observed as denoted by the dashed lines in Figs. 8.23a and 8.24a [42].

[42] The results obtained without employing the angular momentum projection algorithm have been simulated using a first order spatial discretisation in order to amplify the difference.

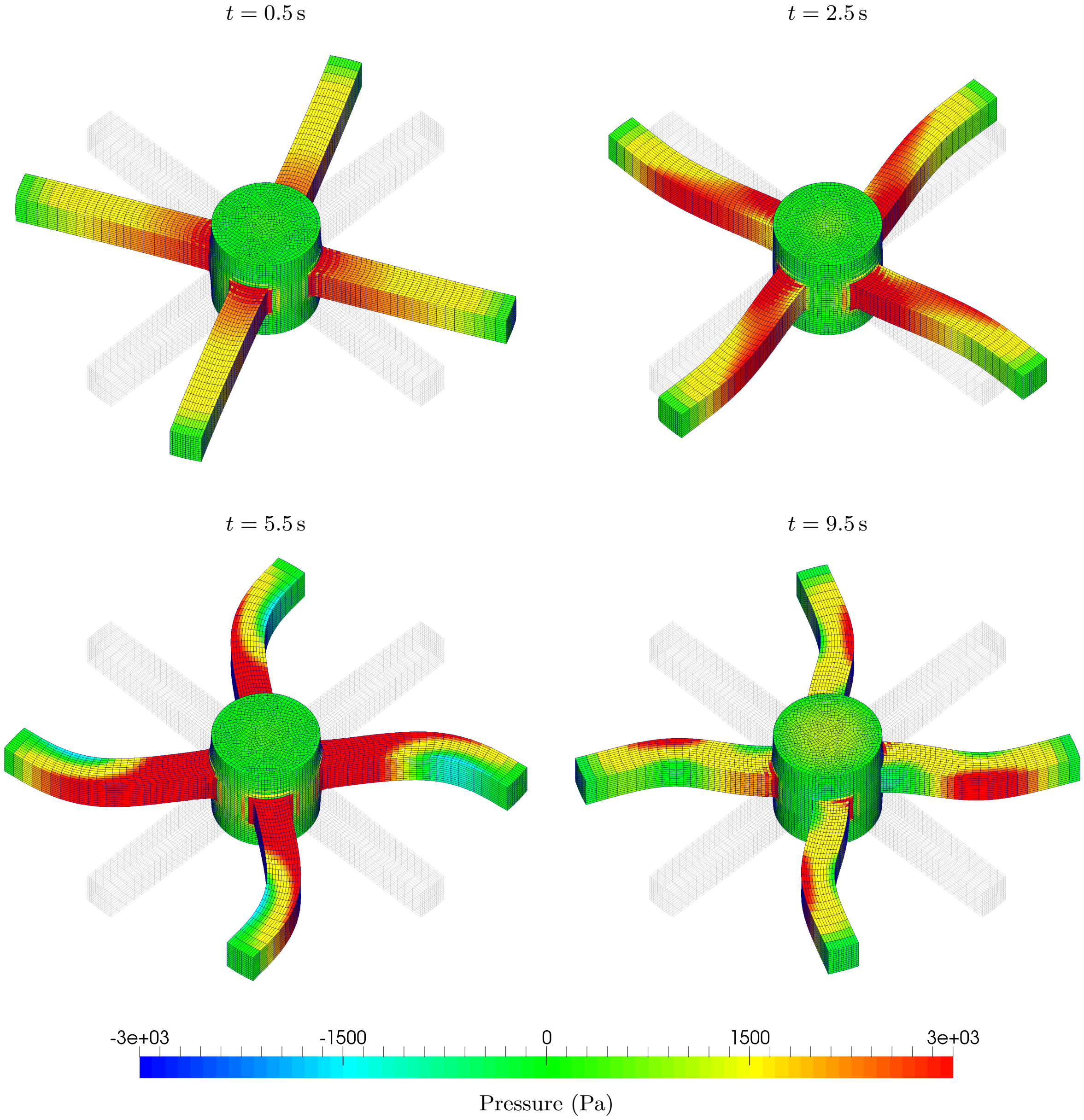
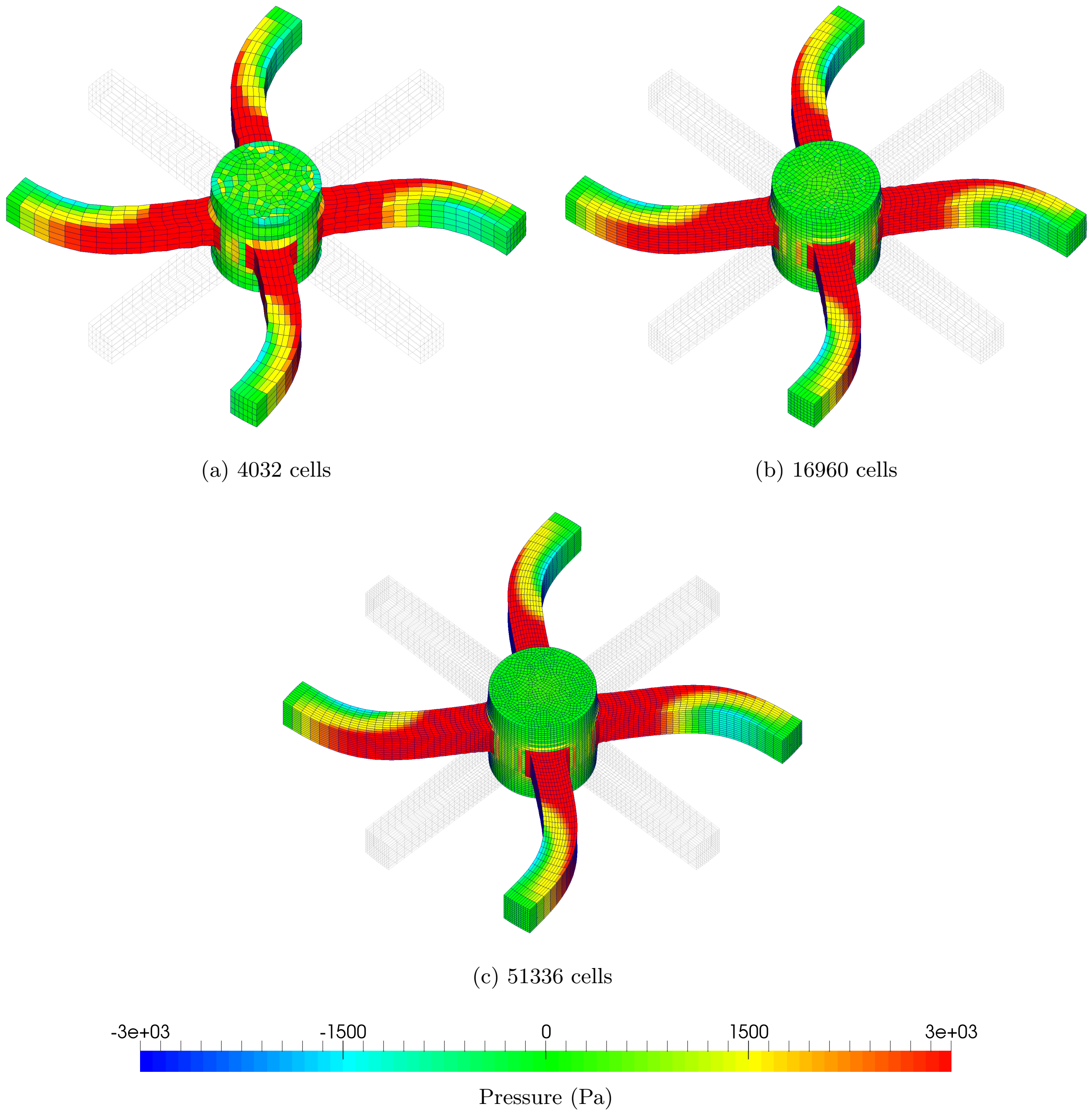


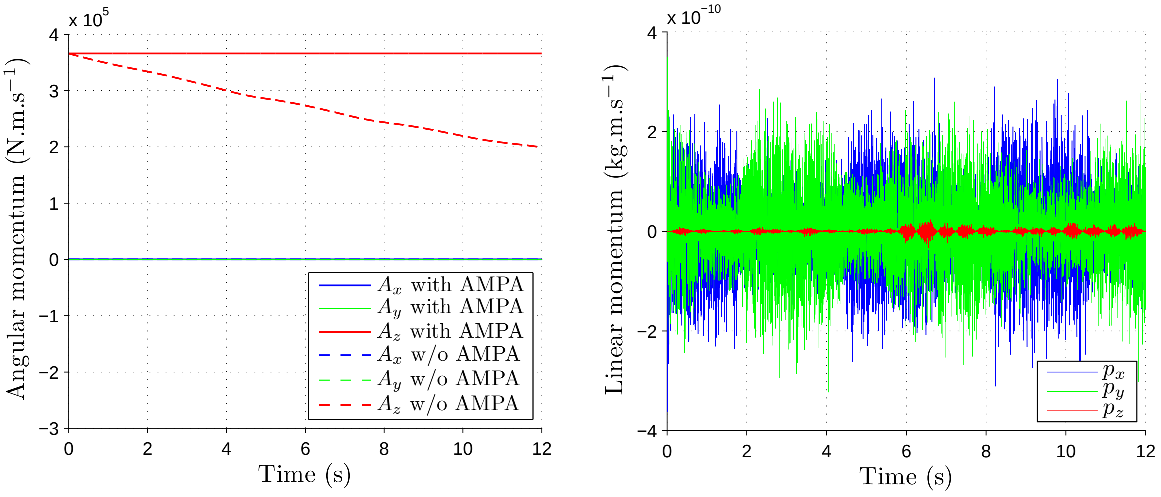
FIGURE 8.21: Satellite-like structure: Time evolution of the deformation along with the pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. Results obtained using a discretisation of 51336 hexahedral elements with an angular velocity $\boldsymbol{\omega}_0 = [0, 0, 1]^T$ rad/s. A polyconvex neo-Hookean material is used with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$, $\alpha_{\mathrm{CFL}} = 0.3$ and $\Delta t \approx 1.3 \times 10^{-3}$ s.


(a) 4032 cells

(b) 16960 cells

(c) 51336 cells

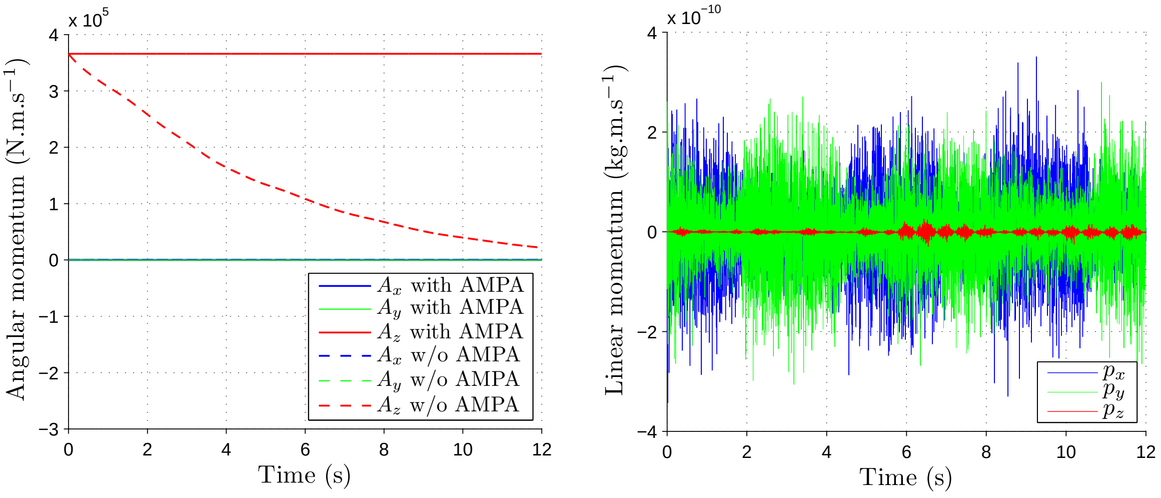
FIGURE 8.22: Satellite structure: Mesh refinement of deformed shapes along with pressure distribution at time $t = 5.5\,\mathrm{s}$ using various mesh sizes: (a) 4032; (b) 16960; and (c) 51336 hexahedral elements. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with an angular velocity $\boldsymbol{\omega}_0 = [0, 0, 1]^T$ rad/s. A polyconvex material is used with $\rho_0 = 1000\ \mathrm{kg/m}^3$, $E = 50.05$ kPa, $\nu = 0.3$ and $\alpha_{\mathrm{CFL}} = 0.3$.



(a) Global angular momentum

(b) Global linear momentum

FIGURE 8.23: Satellite-like structure: Time evolution of the components of (a) global angular momentum with and without the consideration of discrete angular momentum projection algorithm (AMPA); and (b) global linear momentum using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. Results obtained with an angular velocity $\boldsymbol{\omega}_0 = [0, 0, 1]^T$ rad/s with a discretisation of 51336 hexahedral elements. A polyconvex constitutive model is utilised with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$ and $\alpha_{\mathrm{CFL}} = 0.3$.



(a) Global angular momentum

(b) Global linear momentum

FIGURE 8.24: Satellite-like structure: Time evolution of the components of (a) global angular momentum with and without the consideration of discrete angular momentum projection algorithm (AMPA); and (b) global linear momentum using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ X-GLACE scheme. Results obtained with an angular velocity $\boldsymbol{\omega}_0 = [0, 0, 1]^T$ rad/s with a discretisation of 51336 hexahedral elements. A polyconvex constitutive model is utilised with $\rho_0 = 1000$ kg/m^3, $E = 50.05$ kPa, $\nu = 0.3$ and $\alpha_{\mathrm{CFL}} = 0.3$.

8.4 Locking

8.4.1 Bending column

The main objective of this example is to demonstrate the performance of the proposed schemes in nearly incompressible bending dominated scenarios. Following [6, 10, 11, 14–16], a 1 m squared cross section column clamped at the bottom and free on all other sides is presented (see Fig. 8.25). The column is subjected to bending by the application of an initial linearly varying velocity profile in the X-Y plane given by $\boldsymbol{v}_0 = V[(Y/H), 0, 0]^T$ m/s, where $V = 10$ m/s and $H = 6$ m is the height of column. A nearly incompressible neo-Hookean constitutive law is used where the material parameters are density $\rho = 1100$ kg/m^3, Young's modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.45$. For comparison purposes, an ample spectrum of alternative numerical strategies will be employed, namely $\{\boldsymbol{p}, \boldsymbol{F}\}$ X-GLACE, JST-VCFVM [10], Upwind-VCFVM [11], $\{\boldsymbol{p}, \boldsymbol{F}, J\}$ PG-FEM [14–16], the classical B-bar hexahedral element [24] and Hu-Washizu type variational principle [94].

Fig. 8.26 depicts a sequence of locking-free deformed shapes for the column without the appearance of any spurious pressure instabilities by using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. It is remarkable to point out the excellent locking-free behaviour and pressure distribution observed despite employing only four elements (control volumes) across the thickness of column. Fig. 8.27 demonstrates the importance of performing an accurate reconstruction procedure near the fixed boundary (clamped bottom side). As can observed in Fig. 8.27a, the use of a standard reconstruction procedure (i.e. standard least square gradient scheme used in OpenFOAM [62]) leads to the appearance of hourglassing-type instabilities, which can lead to the breakdown

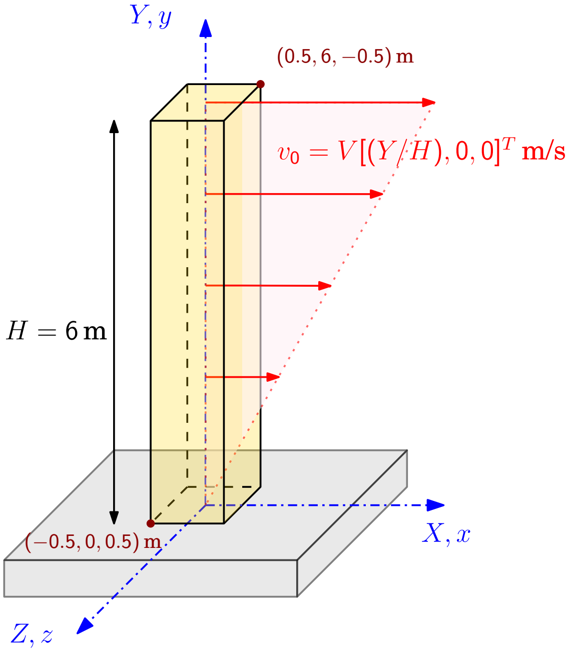


FIGURE 8.25: Bending column: Problem setup

of the numerical simulation over time specially when dealing with nearly incompressible materials. This is in contrast to the smooth profile displayed in Fig. 8.27b, where an enhanced reconstruction procedure is used by taking into account the fixed boundary when evaluating the local gradient (see Fig. 3.6b). A mesh refinement study is shown in Fig. 8.28 which shows very similar deformation for the coarse and fine meshes. Moreover, a time evolution of the horizontal component of the displacement measured at the tip of column $\boldsymbol{X} = [0.5, 6, 0.5]^T$ m is monitored in Fig. 8.29. Results are presented for the proposed C-TOUCH and X-GLACE methodologies for three different mesh sizes. As can be clearly observed, both computational methodologies converge to the same result when the mesh is refined (results obtained with the mesh of $16 \times 96 \times 16$ hexahedral elements).

For completeness, Fig. 8.30 compares the deformed shape and the pressure contour for eight different methodologies, namely (a) C-TOUCH; (b) P-TOUCH ($\xi_{\boldsymbol{F}} = 0.1$); (c) X-GLACE; (d) Classical B-bar method [24]; (e) Upwind-VCFVM [11]; (f) JST-VCFVM [10]; (g) PG-FEM [15]; and (h) Hu-Washizu type variational principle [94]. Results displayed for the cell centred approaches (see Figs. 8.30a to 8.30c) have been obtained with a slightly finer discretisation to that of the other techniques due to the higher numerical dissipation of the cell centred schemes. As can be observed, the results of the schemes proposed match very well those of the other in-house methodologies, where the $\{\boldsymbol{p}, \boldsymbol{F}, J\}$ PG-FEM scheme displays a slight over-diffusion of the pressure field near the clamped boundary. The latter can be attributed to the higher number of stabilisation parameters involved in the formulation [14] which would require an in-depth selection procedure.

8.4.2 Twisting column

In this section, another benchmark problem of a twisting column is presented [8, 10, 11, 14, 16, 94]. The 1 m squared cross section column problem already presented in Section 8.4.1 is considered. The problem is initialised with a sinusoidal angular velocity field (see Fig. 8.31a) relative to the origin given by $\boldsymbol{\omega}_0 = [0, \Omega \sin(\pi Y/2L), 0]^T$ rad/s, where Ω is the initial angular velocity and $H = 6$ m is the height of the column (see Fig. 8.31b). A nearly incompressible neo-Hookean material is used with material properties density $\rho_0 = 1100$ kg/m^3, Youngs's modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.45$ unless otherwise stated.

The time evolution of deformation is shown in Fig. 8.32 along with the pressure distribution when using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. It is remarkable that the complex non-linear locking-free behaviour is captured despite employing only four elements across the thickness of column. The discontinuous cell-wise pressure contours displayed along the longitudinal direction of the domain, for instance at time $t = 0.125$ in Fig. 8.32, do not correspond to any spurious pressure modes. This is clear from Fig. 8.33 where a mesh refinement (see Figs. 8.33c to 8.33e) eliminates this pressure fluctuation in Fig. 8.33a. Notice that spurious pressure checkerboard modes can never be alleviated through mesh refinement due to dissatisfaction of the LBB condition. It is important to emphasise that a nodal averaging (smoothing) process could have been used to display the results (see Fig. 8.33b). However, it is known that this can lead to the removal of any possible spurious oscillations, which is the reason why it hasn't been explicitly carried out. A mesh refinement study is also shown in Fig. 8.34 at time $t = 0.1$ s, where smooth pressure

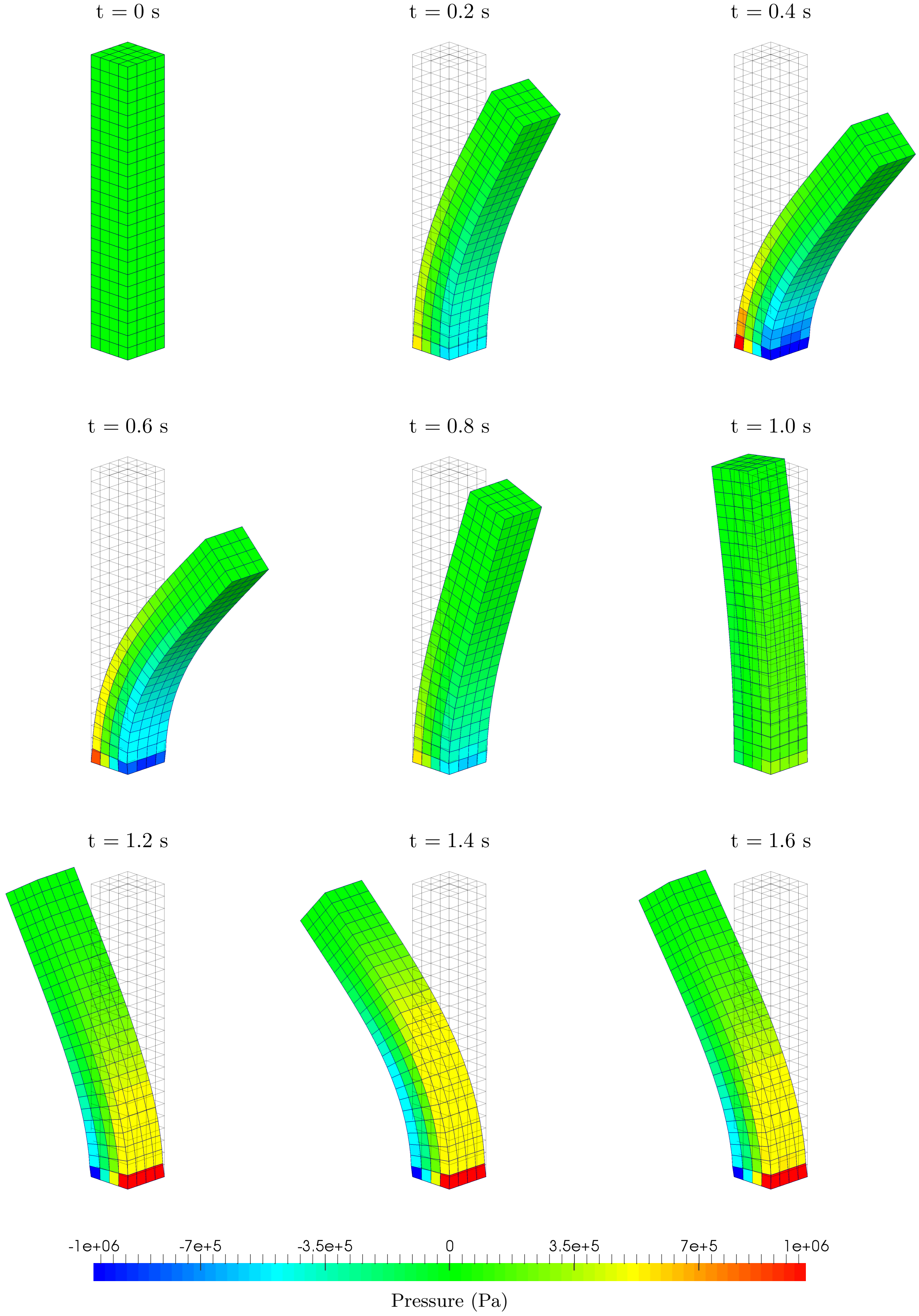
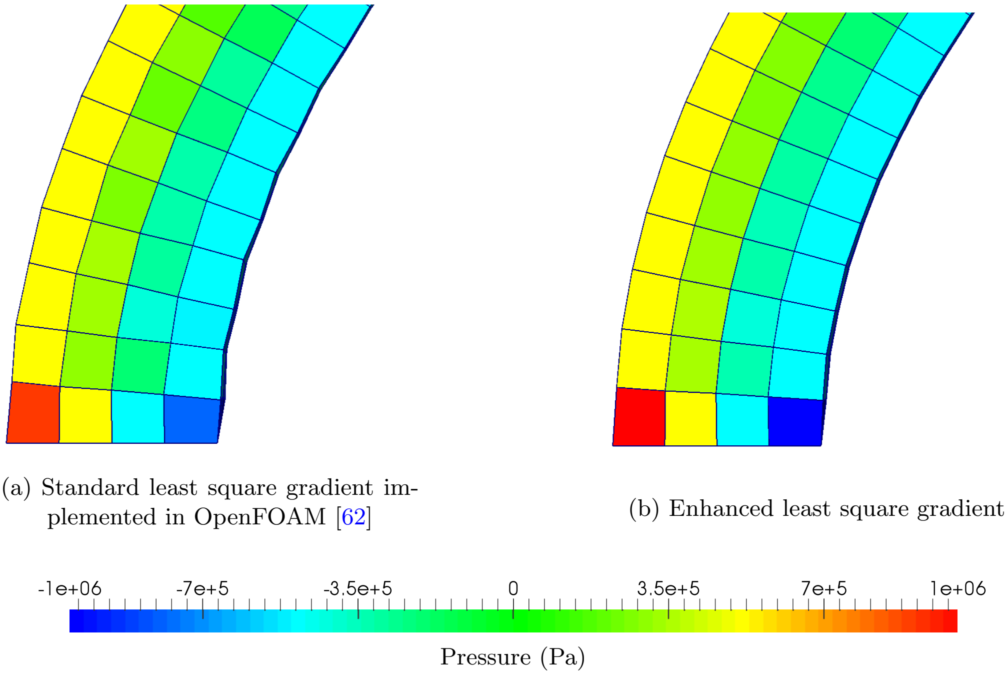
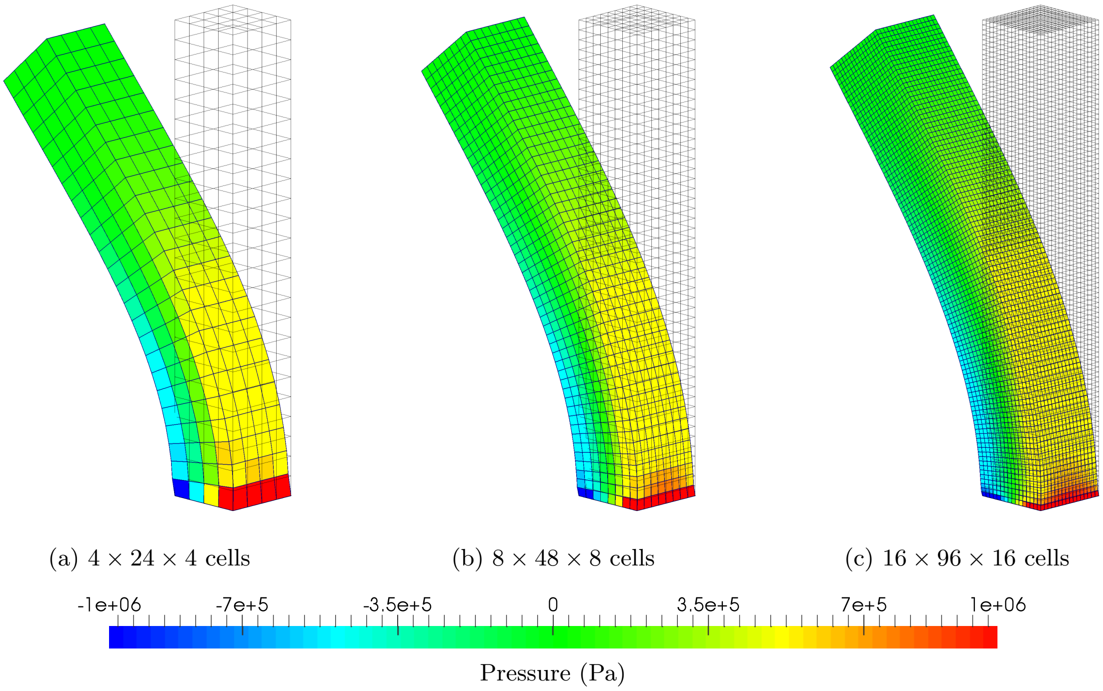


FIGURE 8.26: Bending column: Time evolution of deformation along with pressure distribution using $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. Results obtained with velocity $\boldsymbol{v}_0 = V[(Y/H), 0, 0]^T$ m/s where $V = 10$ m/s using a discretisation of $4 \times 24 \times 4$ hexahedral elements. A neo-Hookean material is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$, $\alpha_{\text{CFL}} = 0.3$ and $\Delta t \approx 2.8 \times 10^{-4}$ s.


(a) Standard least square gradient implemented in OpenFOAM [62]

(b) Enhanced least square gradient

FIGURE 8.27: Bending column: Comparison of deformation and pressure distribution at time $t = 0.5$ s using: (a) standard least square gradient already implemented in OpenFOAM [62]; and (b) enhanced least square gradient. Results are obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with velocity $\boldsymbol{v}_0 = V[(Y/H), 0, 0]^T$ m/s where $V = 10$ m/s. A neo-Hookean model is used where $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$. Discretisation of $4 \times 24 \times 4$ hexahedral elements.


(a) $4 \times 24 \times 4$ cells

(b) $8 \times 48 \times 8$ cells

(c) $16 \times 96 \times 16$ cells

FIGURE 8.28: Bending column: Mesh refinement of deformed shapes plotted with pressure distribution at $t = 1.5$ s using mesh sizes: (a) $h = 1/4$ m; (b) $h = 1/8$ m; and (c) $h = 1/16$ m. Results obtained using $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with velocity $\boldsymbol{v}_0 = V[(Y/H), 0, 0]^T$ m/s where $V = 10$ m/s. A neo-Hookean material is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

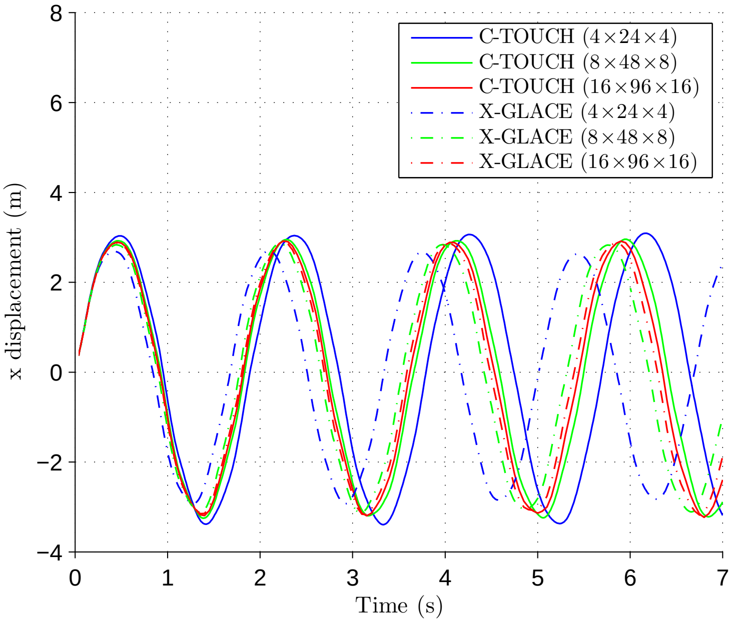


FIGURE 8.29: Bending column: Time evolution of horizontal displacement u_x at material point $\boldsymbol{X} = [0.5, 6, 0.5]^T$ m using $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH and X-GLACE schemes. Results are obtained with velocity $\boldsymbol{v}_0 = V[(Y/H), 0, 0]^T$ m/s, where $V = 10$ m/s, using meshes comprising of $4 \times 24 \times 4$, $8 \times 48 \times 8$ and $16 \times 96 \times 16$ hexahedral elements. A neo-Hookean material is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

results can be observed. Moreover, in Fig. 8.35 the non-dimensionalised height of column is monitored at time $t = 0.1$ s for various mesh sizes. It can be clearly observed that as the mesh is refined, convergence for column height is achieved. Fig. 8.36 has been included to clearly show that the use of classical finite volume update for deformation gradient tensor $\boldsymbol{F}$ introduces curl-errors which accumulate over time leading to non-physical results and eventual breakdown of the numerical scheme. It emphasises that the standard CCFVM cannot be used in the context of large strain solid dynamics and the consideration of curl-free algorithms is of paramount importance (see Chapter 5).

For benchmarking purposes, the problem is simulated and comparisons are made using other available methodologies in Fig. 8.37, namely PG-FEM [14–16], JST-VCFVM [10], Upwind-VCFVM [11], the classical B-bar, the LBB compliant Q2-Q1 hexahedral element [24] and Hu-Washizu type mixed formulation [94]. Results displayed using C-TOUCH, P-TOUCH and X-GLACE (Figs. 8.37a to 8.37c) have been obtained with a slightly finer discretisation to that of the other techniques (due to the higher numerical dissipation of the cell centred schemes). It can be observed that the deformation patterns predicted by the family of numerical mixed methodologies are practically identical, apart from a slight out-of-plane deformation introduced by the JST-VCFVM (Fig. 8.37f). Pressure distribution shown using the B-bar method (8.37d) are shown per cell and not nodally interpolated[43]. Fig. 8.38 includes a similar comparative

[43] Pressure is sometimes nodally interpolated in order to smooth out any possible pressure oscillations [24].

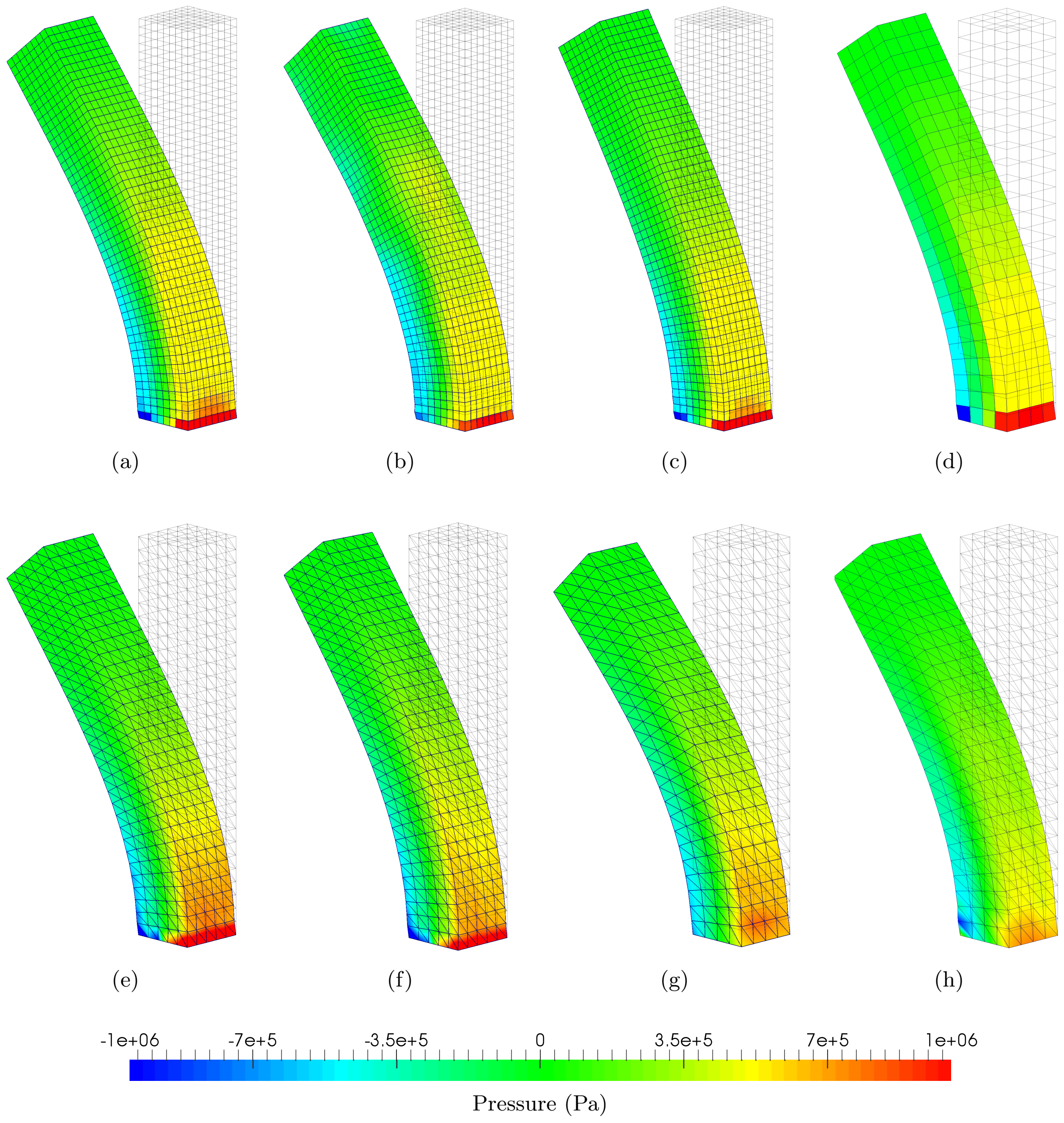
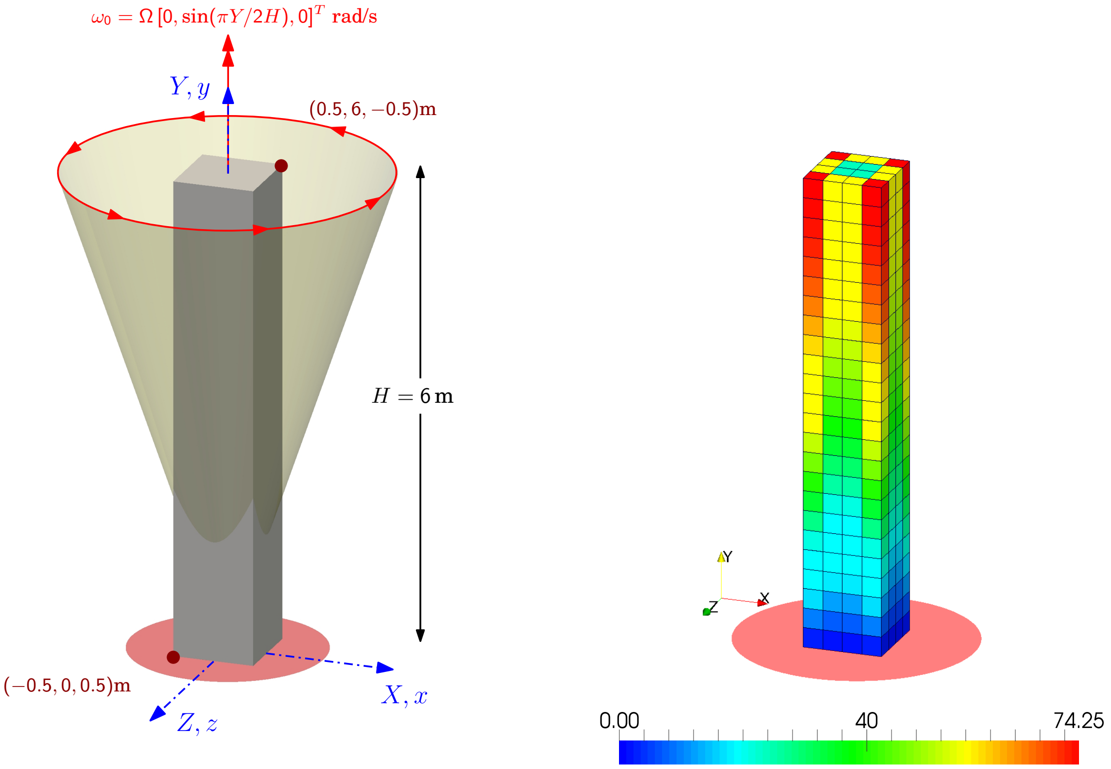


FIGURE 8.30: Bending column: Comparison of deformed shapes plotted with pressures at time $t = 1.5$ s using various numerical schemes: (a) $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH; (b) $\{\boldsymbol{p}, \boldsymbol{F}\}$ P-TOUCH ($\xi_{\boldsymbol{F}} = 0.1$); (c) $\{\boldsymbol{p}, \boldsymbol{F}\}$ X-GLACE [5]; (d) B-bar hexahedral method; (e) Upwind-VCFVM [11]; (f) JST-VCFVM [10]; (g) PG-FEM [15]; and (h) Hu-Washizu type variational principle [94]. Results are obtained with velocity $\boldsymbol{v}_0 = V[(Y/H), 0, 0]^T$ m/s where $V = 10$ m/s. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa and $\nu = 0.45$.



(a) Initial configuration

(b) Initial velocity (m/s) for $\Omega = 105$ rad/s and mesh

Figure 8.31: Twisting column: Problem setup.

study, with a reduced number of methodologies and with a higher Poisson's ratio $\nu = 0.495$. Crucially, all computational mixed methodologies presented, produce very similar deformation patterns with smooth pressure distribution and absence of locking. Here again, the results obtained with the C-TOUCH and X-GLACE schemes (Figs. 8.38a and 8.38b) have been obtained with a slightly finer mesh. The results have also been benchmarked using mixed Smooth Particle Hydrodynamics (SPH) methodology [18, 19], in Figs. 8.38g and 8.38h. In this case of a high Poisson's ratio $\nu = 0.495$, the simulation could not be run with JST-VCFVM, Upwind-VCFVM and hyperelastic-GLACE schemes due to lack of robustness.

In order to simulate highly incompressible materials with Poisson's ratio $\nu > 0.495$, we observe that the Riemann solver presented in Section 4.3 is not robust enough. In Fig. 8.39a, the twisting scenario is simulated using $\nu = 0.4999$ where the resulting pressure checker-boarding can be clearly observed. One way to overcome this, is to utilise preconditioned wave speeds ($\tilde{c}_p$, $\tilde{c}_s$) in the Riemann solver as presented in Section 4.6. The resulting deformation pattern with smooth pressure contours can be observed in Fig. 8.39b. In Fig. 8.40, two different system of conservation equations, namely $\{\boldsymbol{p}, \boldsymbol{F}\}$ and $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$, at time $t = 0.1\,$s are presented using the C-TOUCH scheme. It is interesting to notice that the deformation and pressure distribution obtained are practically identical for both formulations. Interestingly, the discontinuous cell pressure distribution along the longitudinal direction of the column does not correspond to any pressure instability since it gets eliminated after mesh refinement (Fig. 8.40c). Alternatively, a nodal averaging (smoothing) process could have been used to display the results (see Fig. 8.40d).

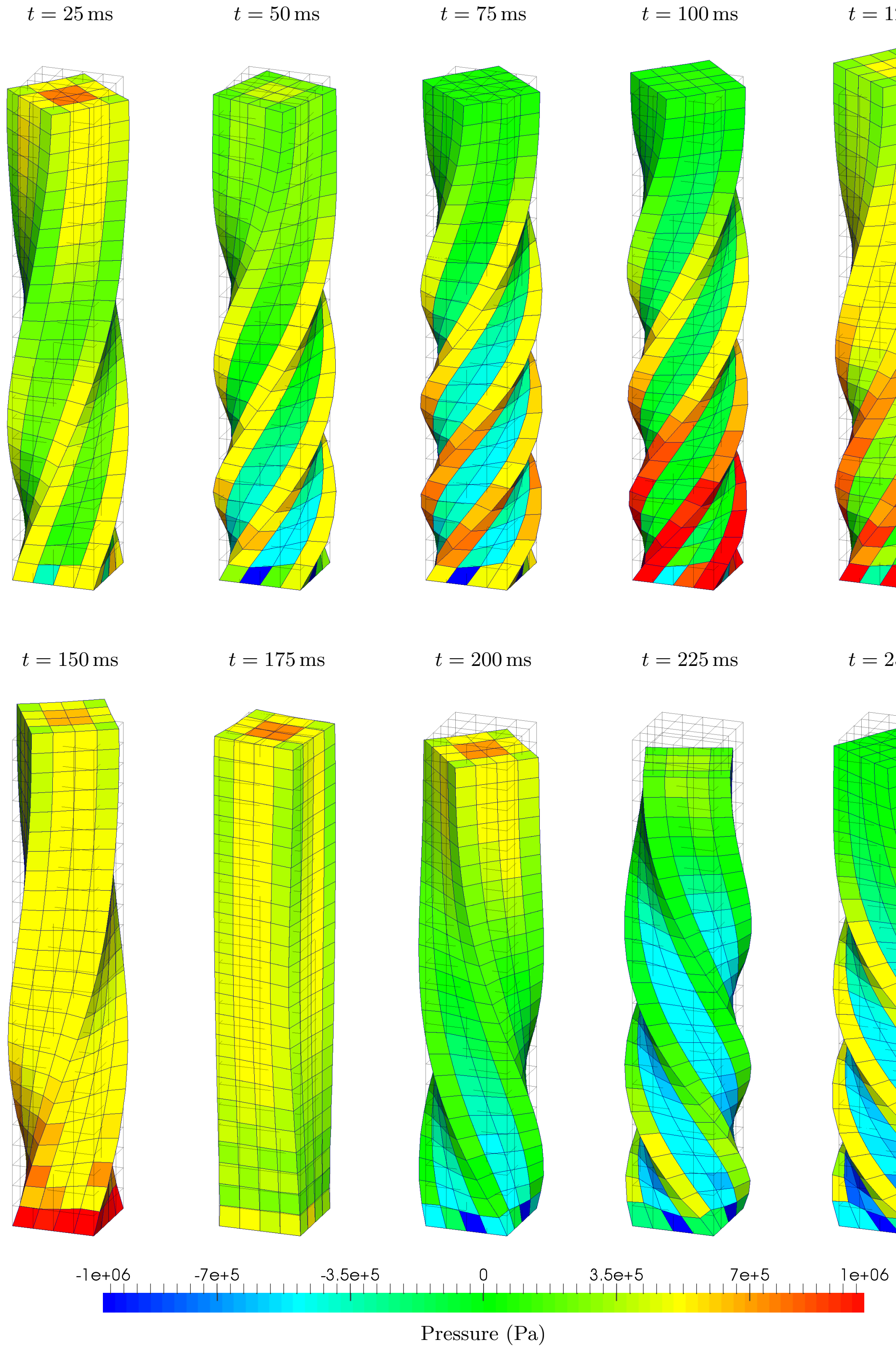


FIGURE 8.32: Twisting column: Time evolution of deformation along with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. Results obtained with an angular velocity $\boldsymbol{\omega}_0 = \varOmega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\varOmega = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$, $\alpha_{\text{CFL}} = 0.3$ and $\varDelta t \approx 2.5 \times 10^{-3}$ s. Discretisation of $4 \times 24 \times 4$ hexahedral elements.

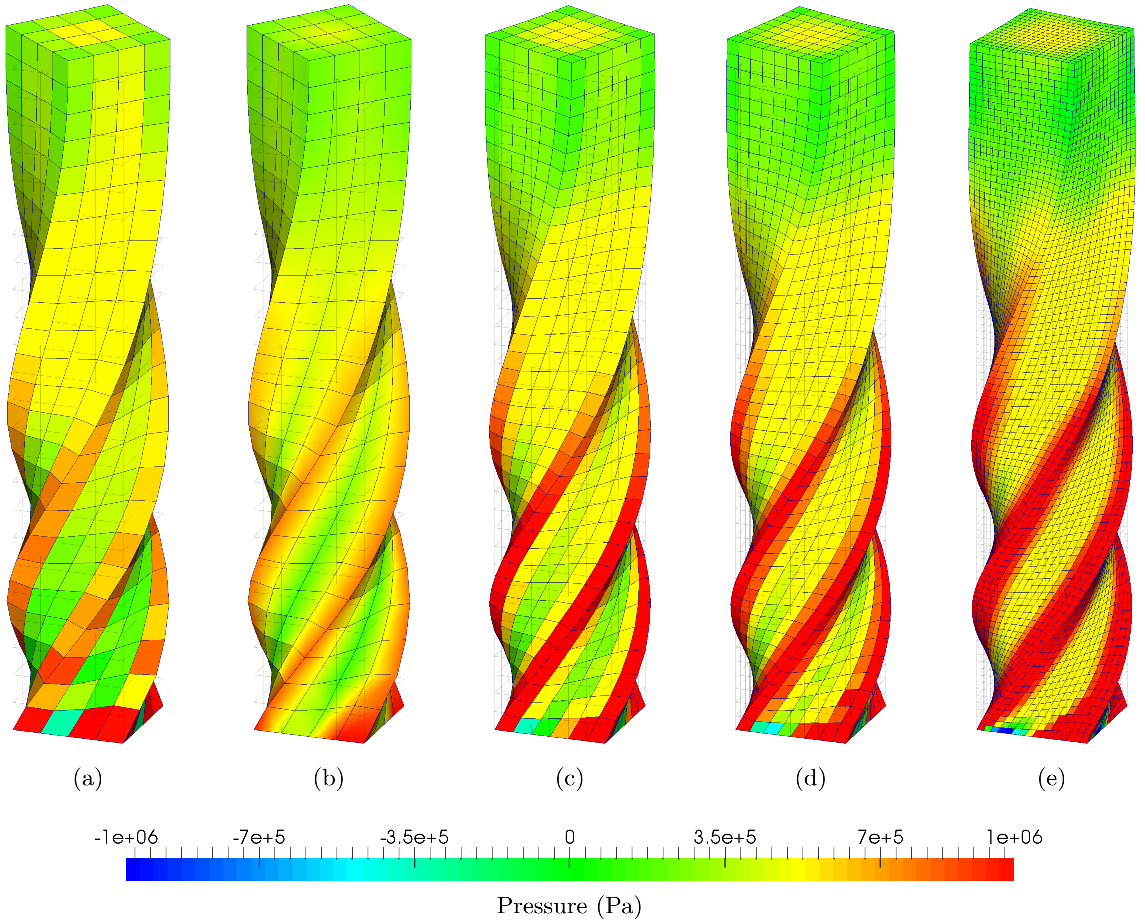
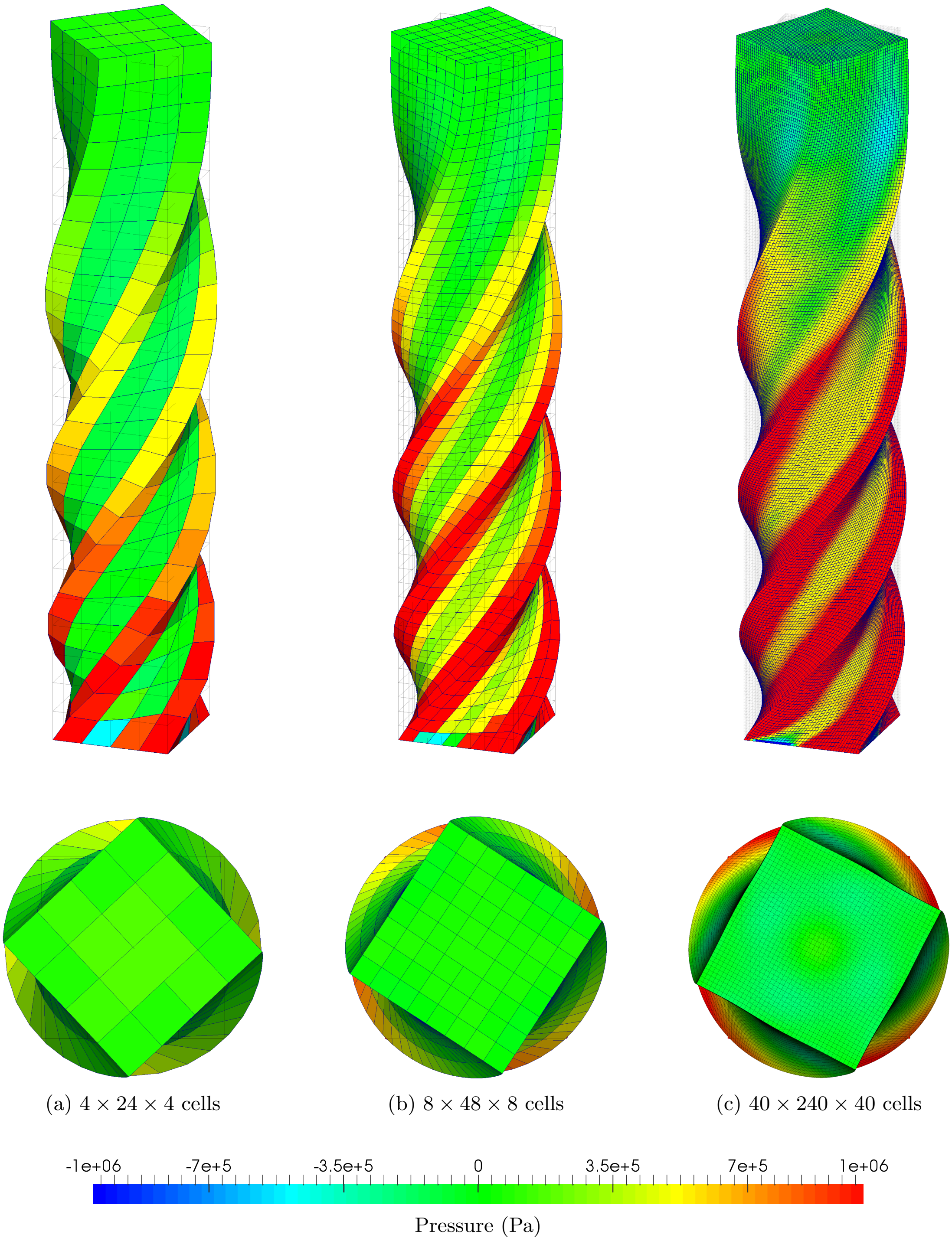


FIGURE 8.33: Twisting column: Comparison of the deformation along with the pressure distribution at time $t = 125\,\text{ms}$ using meshes with (a) 4×24×4 (cell values); (b) 4×24×4 (node values); (c) 6×36×6 (cell values); (d) 8×48×8 (cell values); and (e) 16×96×16 (cell values) hexahedral elements. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

However, this can lead to the removal of possible pressure fluctuations which is the reason why it has not been carried out.

Insofar as a symmetric hexahedral mesh is employed, the column is expected to prevent out-of-axis deformation. This can be easily shown by monitoring displacement of a point located at the top surface of the column. Fig. 8.41 shows that the evolution of horizontal displacement components (e.g. u_x and u_z) at point $\boldsymbol{X} = [0, 6, 0]^T$ m is within zero machine accuracy.

More crucially, the problem becomes significantly challenging by increasing the initial angular velocity now to $\Omega = 200\,\text{rad/s}$ with a Poisson's ratio of $\nu = 0.499$. A mesh refinement study is shown in Fig. 8.43. In particular, the number of twists shown in the column is captured extremely well even with the use of a coarse mesh. Aiming to show mesh convergence, Fig. 8.44a illustrates the time evolution of (non-dimensionalised) height of the column using successive meshes of $4 \times 24 \times 4$, $8 \times 48 \times 8$ and $16 \times 96 \times 16$ hexahedral elements. Moreover, the overall

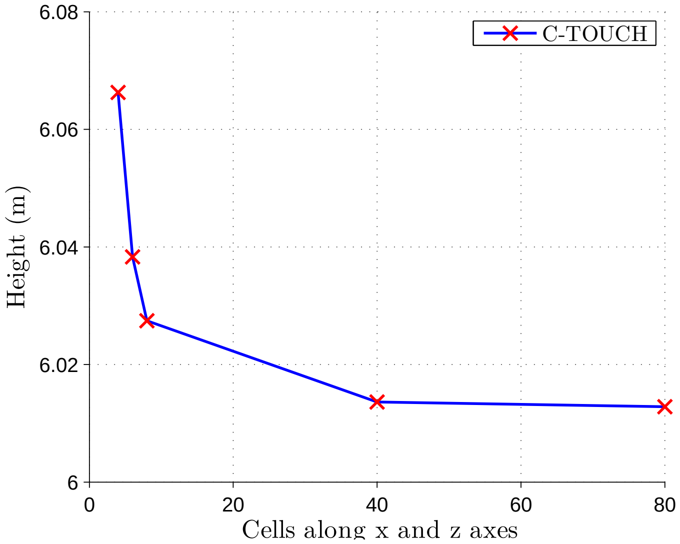


(a) 4 × 24 × 4 cells

(b) 8 × 48 × 8 cells

(c) 40 × 240 × 40 cells

FIGURE 8.34: Twisting column: Mesh refinement of deformed shapes with pressure distribution at time $t = 100\,\text{ms}$ using meshes with (a) $4 \times 24 \times 4$; (b) $8 \times 48 \times 8$; and (c) $40 \times 240 \times 40$ cells. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with an angular velocity $\boldsymbol{\omega}_0 = \mathit{\Omega}\,[0, \sin(\pi Y/2H), 0]^T$ rad/s, where $\mathit{\Omega} = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.



Mesh size	Height (m)
$4 \times 24 \times 4$	6.0663
$6 \times 36 \times 6$	6.0383
$8 \times 48 \times 8$	6.0275
$40 \times 240 \times 40$	6.0136
$80 \times 480 \times 80$	6.0128

FIGURE 8.35: Twisting column: Grid independence of column height at material position $\boldsymbol{X} = [0, 6, 0]^T$ m using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. Results obtained at $t = 100\,\text{ms}$ using angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 100$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$

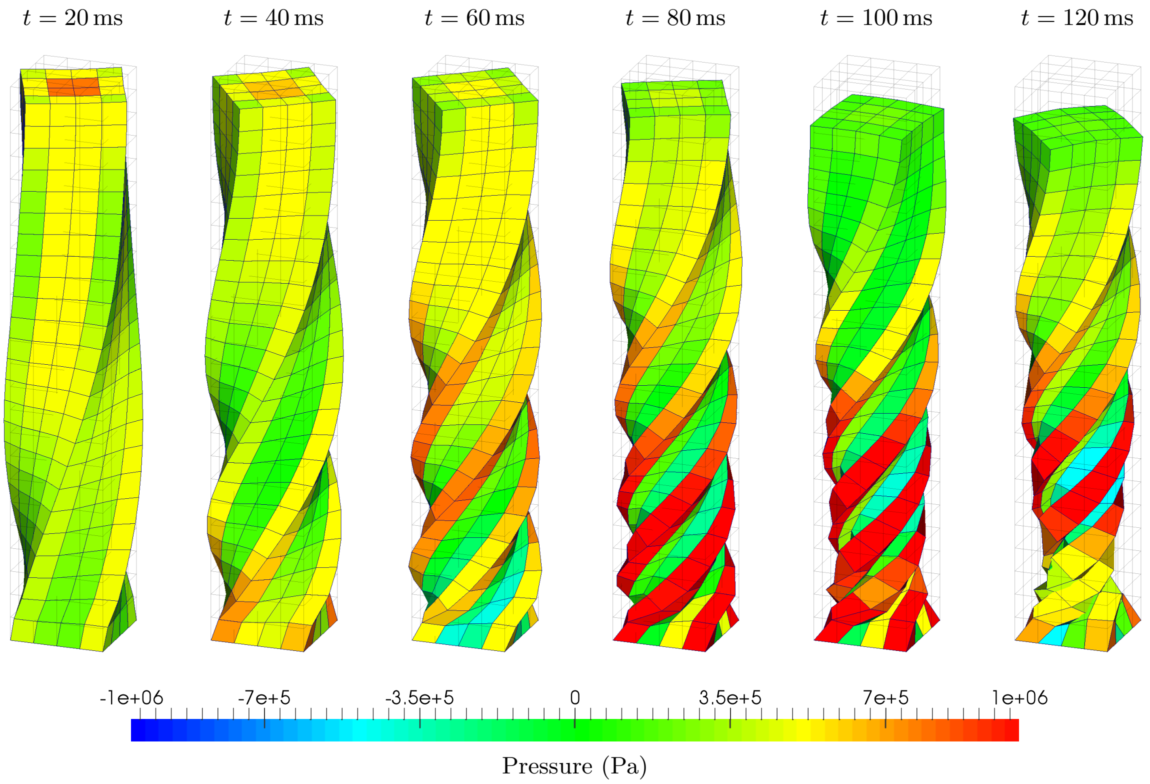


FIGURE 8.36: Twisting column: Time evolution of deformation plotted with pressure distribution. Results are obtained using the standard FVM update (P-TOUCH with $\xi_F = 0$) with angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s, where $\Omega = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$. Discretisation of $4 \times 24 \times 4$ hexahedral elements.

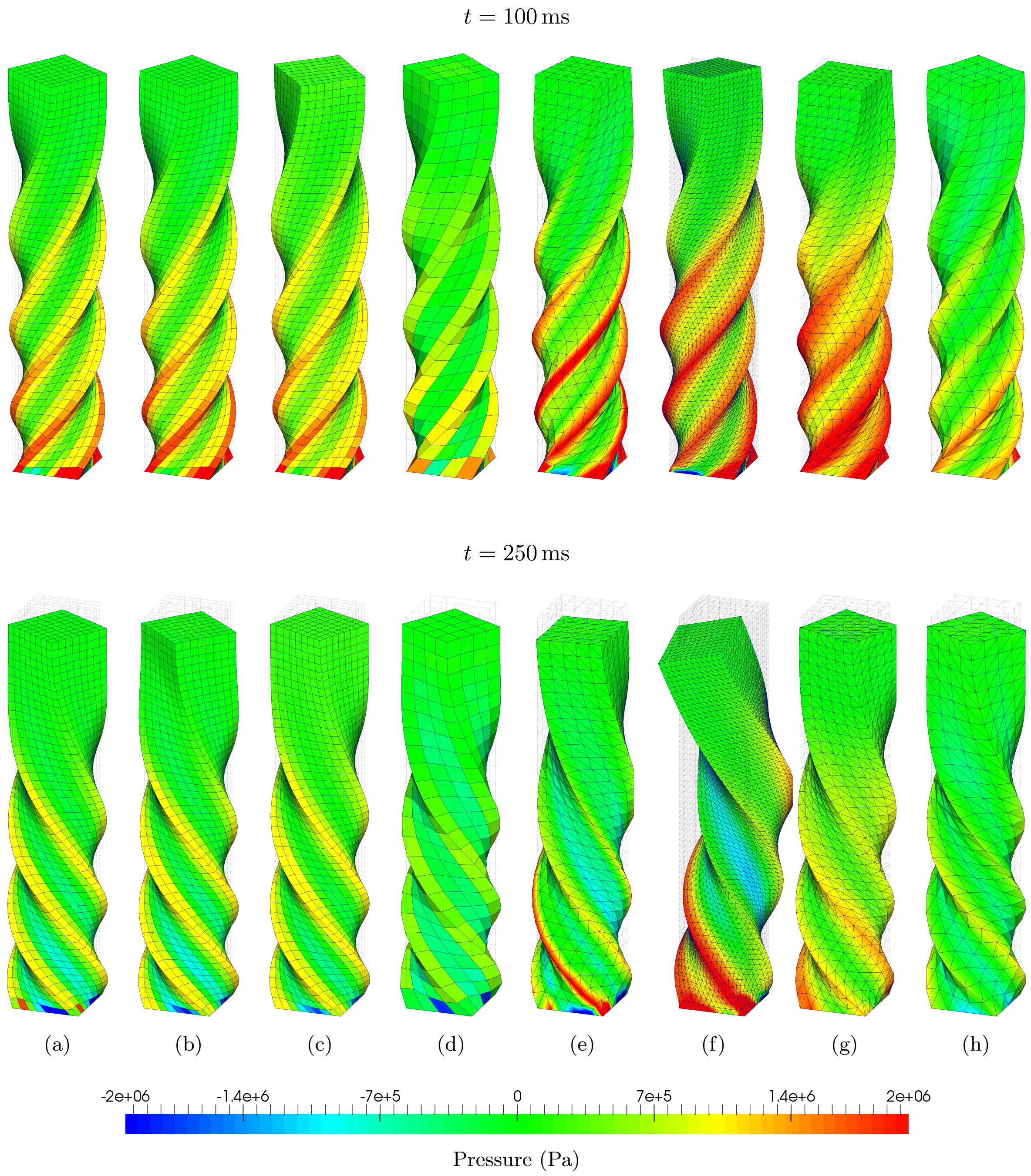
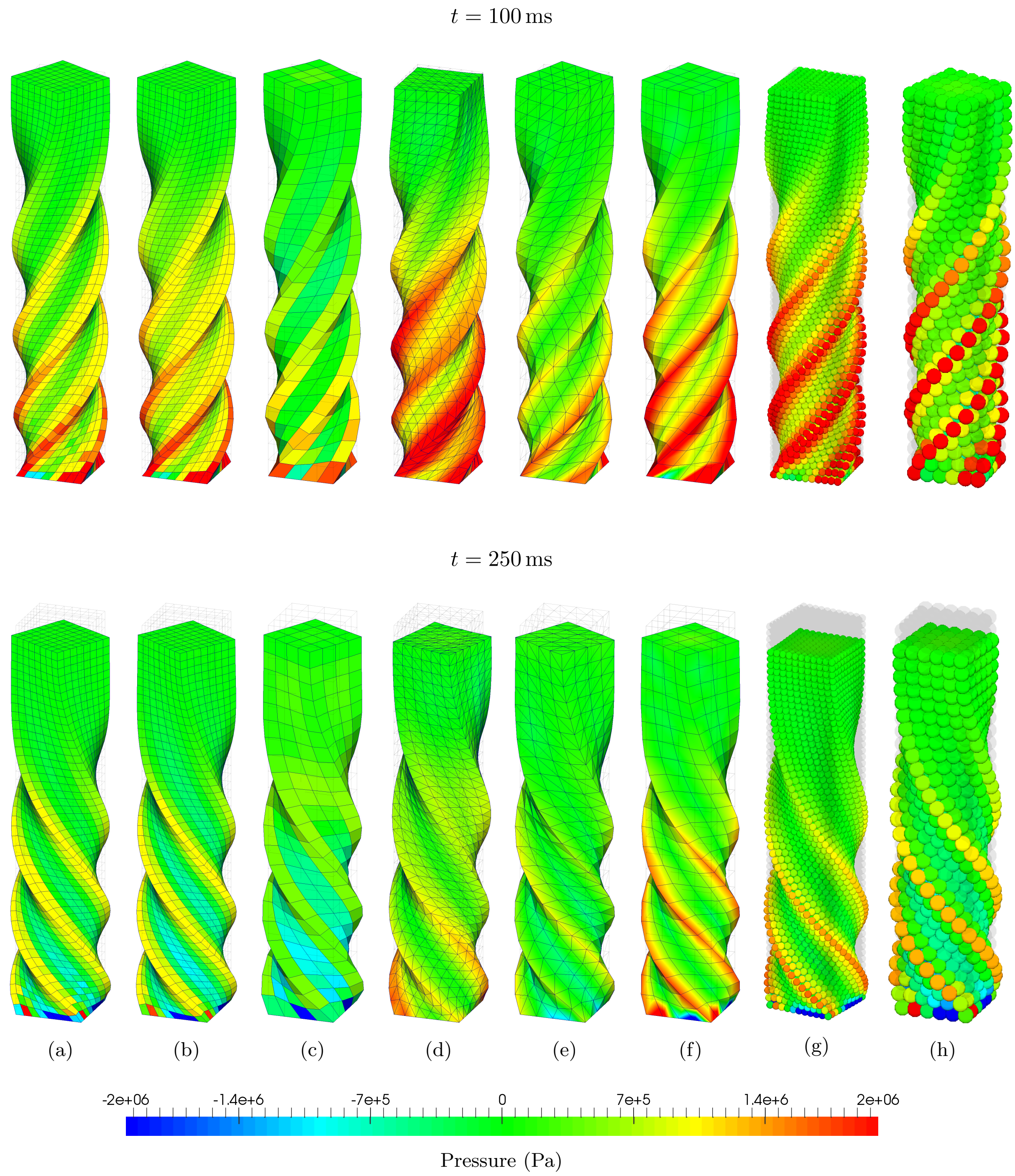


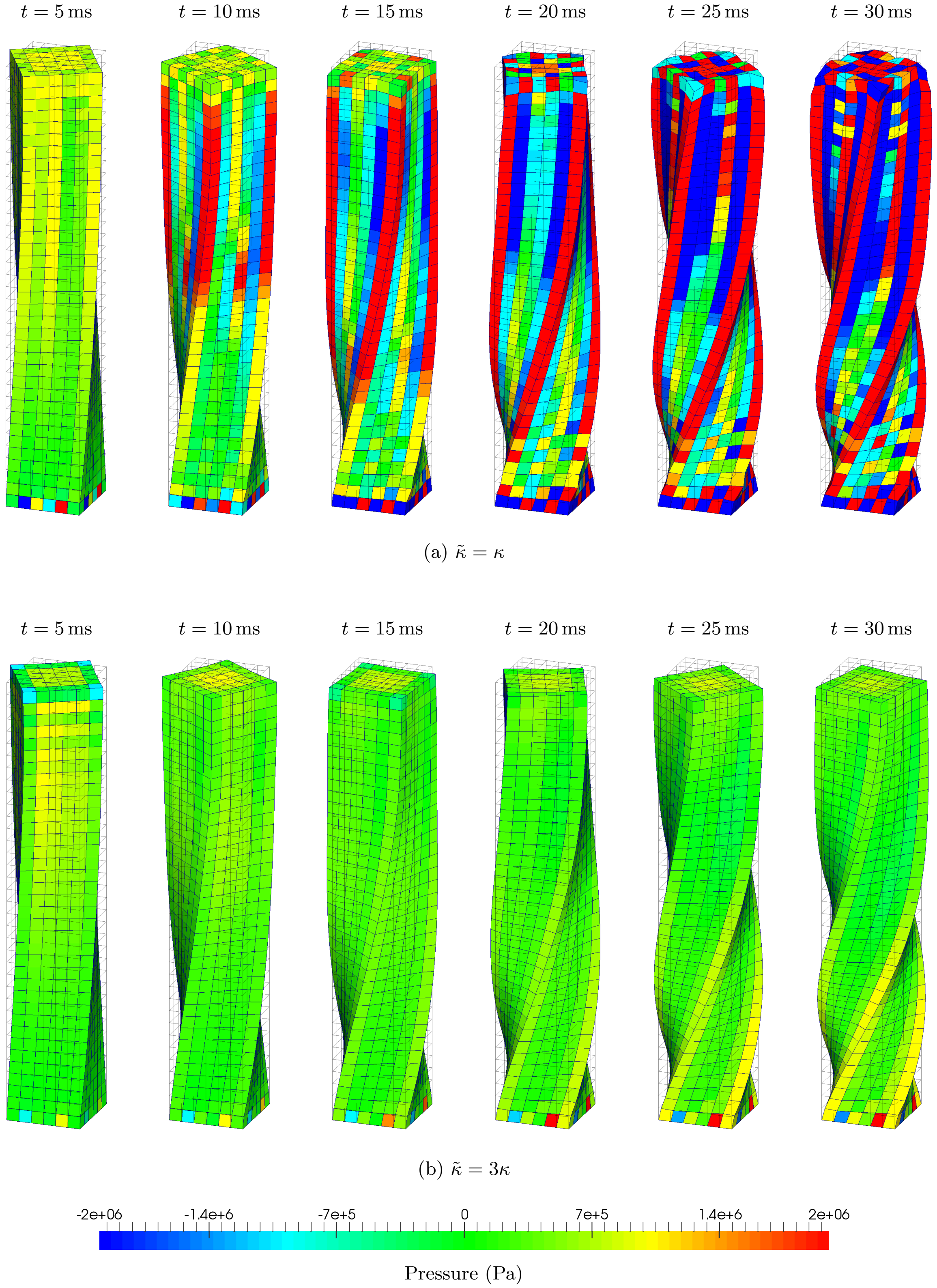
FIGURE 8.37: Twisting column: Comparison of deformed shapes plotted with pressures at time $t = 100\,\text{ms}$ and $t = 250\,\text{ms}$ using various numerical schemes: (a) C-TOUCH; (b) P-TOUCH ($\xi_{\boldsymbol{F}} = 0.1$); (c) X-GLACE [5]; (d) B-bar hexahedral method; (e) Upwind-VCFVM [11]; (f) JST-VCFVM [10]; (g) PG-FEM [15]; and (h) Hu-Washizu type variational principle [94]. Results obtained with angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa and $\nu = 0.45$.


$t = 100\,\mathrm{ms}$

$t = 250\,\mathrm{ms}$

(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 8.38: Twisting column: Comparison of deformed shapes plotted with pressures at time $t = 100\,\mathrm{ms}$ and $t = 250\,\mathrm{ms}$ using various numerical schemes: (a) C-TOUCH; (b) P-TOUCH ($\xi_{\boldsymbol{F}} = 0.05$); (c) B-bar hexahedral method; (d) PG-FEM [15]; (e) Hu-Washizu type variational principle [94]; (f) Q2-Q1 hexahedral FEM; (g) JST-SPH [18]; and (h) SUPG-SPH [19]. Results obtained with an angular velocity $\boldsymbol{\omega}_0 = [0, \mathit{\Omega} \sin(\pi Y/2H), 0]^T$ rad/s where $\mathit{\Omega} = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with density $\rho_0 = 1100$ kg/m^3, Young's modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.495$.



(a) $\tilde{\kappa} = \kappa$

(b) $\tilde{\kappa} = 3\kappa$

FIGURE 8.39: Twisting column: Time evolution of deformation plotted with pressure distribution highlighting the importance of preconditioned dissipation for nearly incompressible scenarios. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with an angular velocity $\boldsymbol{\omega}_0 = \varOmega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s, where $\varOmega = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.4999$ and $\alpha_{\text{CFL}} = 0.3$. Discretisation of $6 \times 36 \times 6$ hexahedral elements.

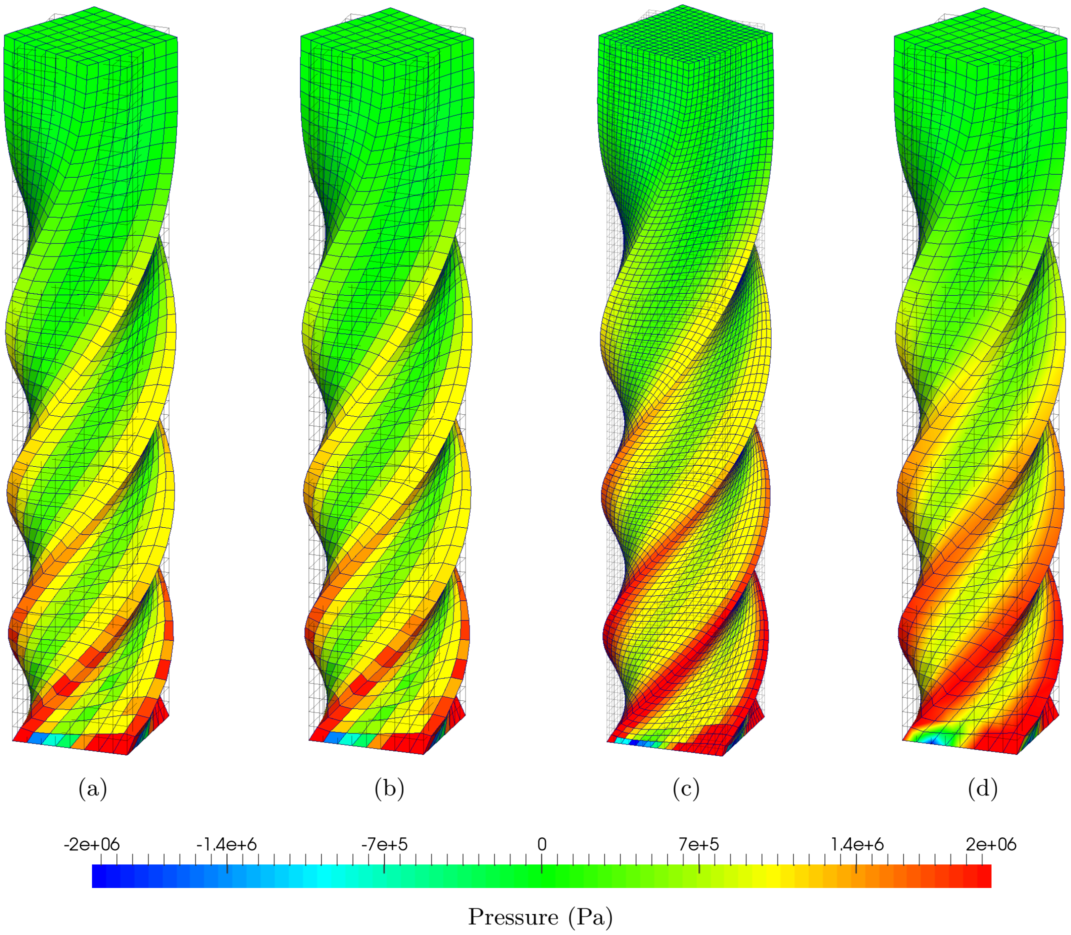


FIGURE 8.40: Twisting column: Comparison of deformed shapes along with the pressure distribution using (a) $\{\boldsymbol{p}, \boldsymbol{F}\}$ (cell values); (b) $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ (cell values); (c) $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ (cell values); and (d) $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ (node values) C-TOUCH schemes with $\tilde{\kappa} = 3\kappa$. Results obtained at time $t = 0.1$ s with an angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 105$ rad/s and $H = 6$ m. A neo-Hookean constitutive model is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.4999$, $\alpha_{\text{CFL}} = 0.3$ and $\Delta t \approx 1.3 \times 10^{-5}$ s. Discretisation of $8 \times 48 \times 8$ and $16 \times 96 \times 16$ hexahedral elements.

numerical dissipation introduced within the C-TOUCH scheme is plotted in Fig. 8.44b. As expected, reduced numerical dissipation can be obtained by increasing the mesh density.

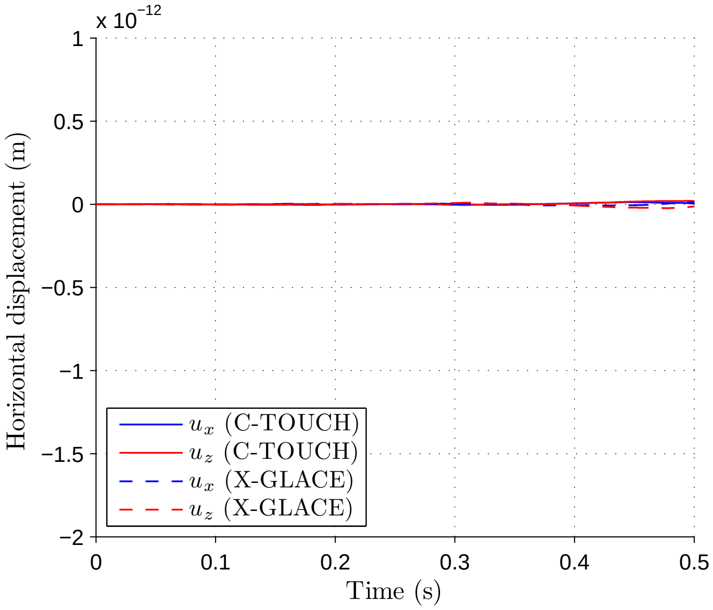


FIGURE 8.41: Twisting column: Comparison of time evolution of horizontal displacements $\boldsymbol{u}_x$ and $\boldsymbol{u}_z$ of the point at top of column along the central Y axis $\boldsymbol{X} = [0, 6, 0]^T$ m using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH and X-GLACE schemes with $\tilde{\kappa} = 3\kappa$. Results obtained using a discretisation of $8 \times 48 \times 8$ hexahedral elements with an angular velocity $\boldsymbol{\omega}_0 = \mathit{\Omega}\,[0, \sin(\pi Y/2H), 0]^T$ rad/s, where $\mathit{\Omega} = 105$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.495$ and $\alpha_{\text{CFL}} = 0.3$.

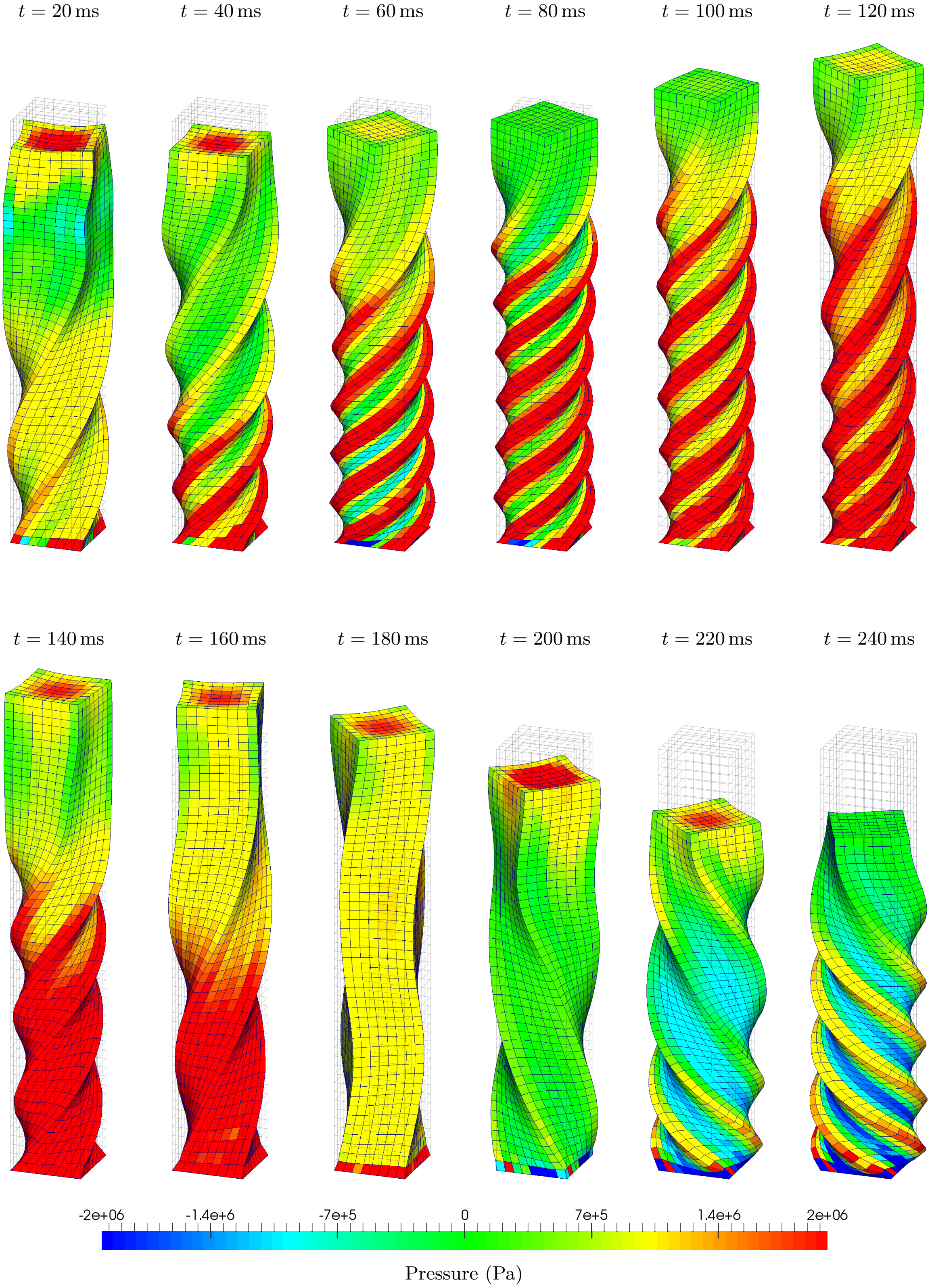
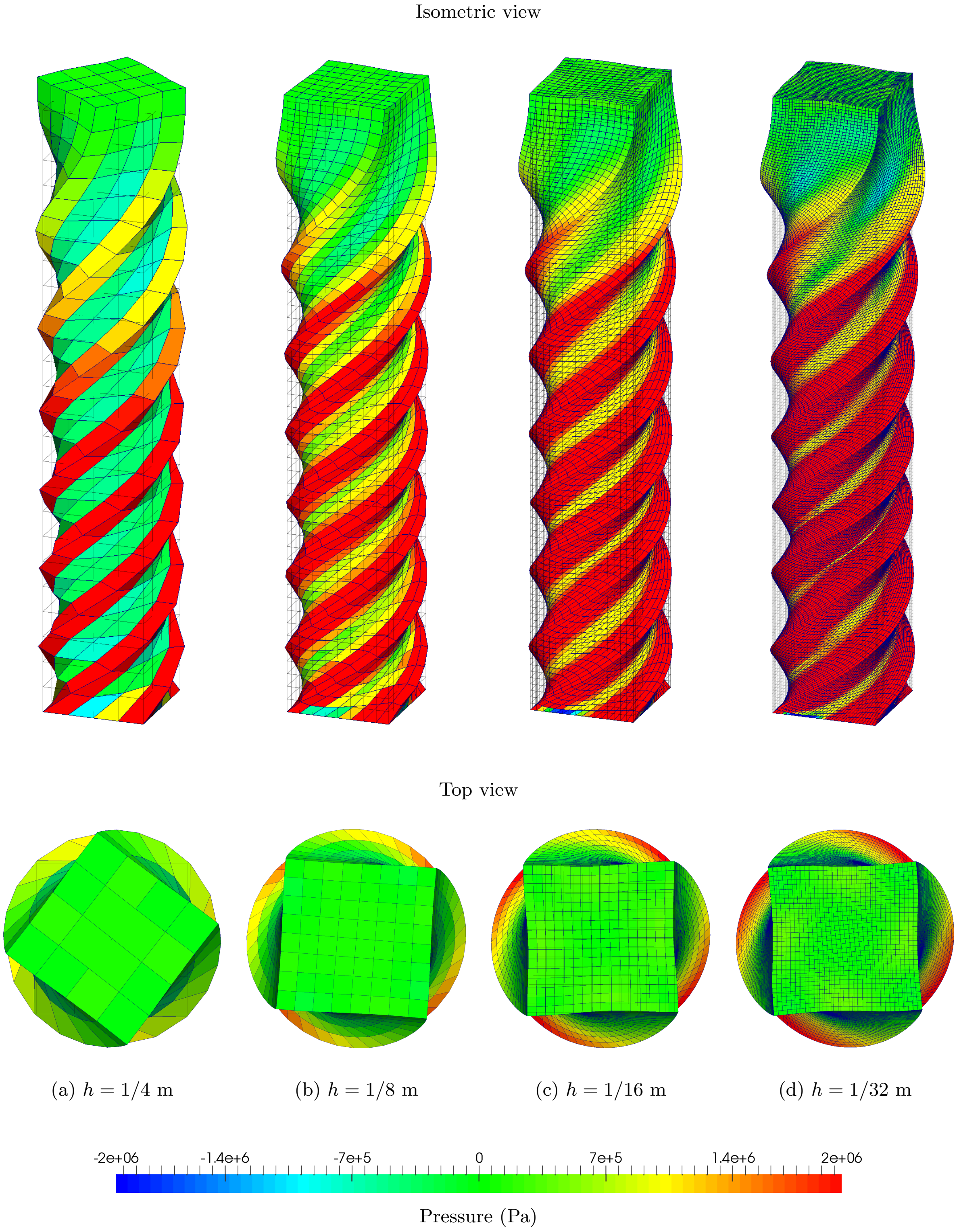
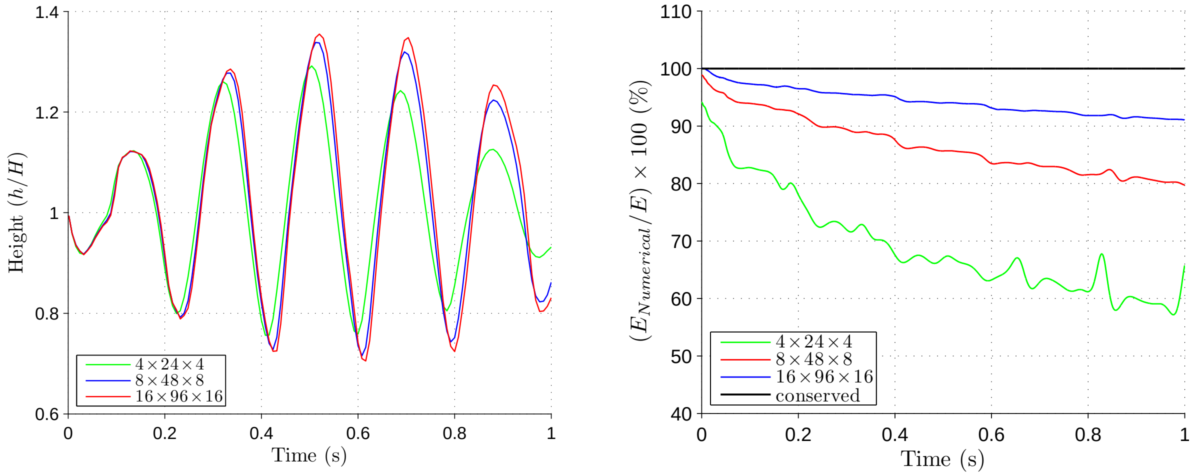


FIGURE 8.42: Twisting column: Time evolution of deformation plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. Results obtained using a discretisation of $10 \times 60 \times 10$ cells and angular velocity $\boldsymbol{\omega}_0 = \mathit{\Omega}\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\mathit{\Omega} = 200$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$, $\alpha_{\mathrm{CFL}} = 0.3$ and $\mathit{\Delta} t \approx 6 \times 10^{-5}$ s.



(a) $h = 1/4$ m (b) $h = 1/8$ m (c) $h = 1/16$ m (d) $h = 1/32$ m

FIGURE 8.43: Twisting column: Mesh refinement of deformed shapes with pressure distribution obtained using an increased angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 200$ rad/s and $H = 6$ m. Results obtained at $t = 90$ ms using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with $\tilde{\kappa} = 3\kappa$ using meshes with (a) $4 \times 24 \times 4$; (b) $8 \times 48 \times 8$; (c) $16 \times 96 \times 16$; and (d) $32 \times 192 \times 32$ hexahedral elements. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\text{CFL}} = 0.3$.



(a) Column height

(b) Numerical dissipation

FIGURE 8.44: Twisting column: Time evolution of (a) non-dimensionalised height of column measured at the material point $\boldsymbol{X} = [0, 6, 0]^T$ m; and (b) numerical dissipation using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with $\tilde{\kappa} = 3\kappa$. Results obtained using a discretisation of $4 \times 24 \times 4$, $8 \times 48 \times 8$ and $16 \times 96 \times 16$ hexahedral elements with an increased angular velocity $\boldsymbol{\omega}_0 = \mathit{\Omega}\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\mathit{\Omega} = 200$ rad/s and $H = 6$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\text{CFL}} = 0.3$.

8.5 Von Mises plasticity

8.5.1 Taylor impact

In 1948, Taylor [136] investigated the impact of a cylindrical bar on a rigid surface with a very high velocity to determine the dynamic yield stress of materials. Since then, this classical benchmark example has been numerically investigated on several occasions in [15, 40, 133, 134, 137–142] and within the context of Finite Volume Method in [8, 10, 66, 80]. We simulate the plastic deformation of a circular copper bar with an initial radius $r_0 = 3.2$ mm and an initial length of 32.4 mm, which impacts against a rigid frictionless wall at time $t = 0$ s with an initial velocity $\boldsymbol{v}_0 = [0, -227, 0]^T$ m/s (see Fig. 8.45). A von Mises hyperelastic-plastic material with isotropic hardening (see Algorithm 2.1) is chosen for simulation until the end time $t = 80\,\mu$s, where a steady state solution is achieved (nearly all of the kinetic energy has dissipated into internal energy). The material parameters are such that density $\rho_0 = 8.930 \times 10^3$ kg/m^3, Young's modulus $E = 117$ GPa, Poisson's ratio $\nu = 0.35$, yield stress, $\bar{\tau}_y^0 = 0.4$ GPa and hardening/plastic modulus $H = 0.1$ GPa. In order to simulate this problem, the nodes in contact with the wall are constrained in such a way that they remain attached to the wall at all times (symmetric boundary condition). This is not true in real-life since a bounce-off motion will occur. Furthermore, due to the existence of two planes of symmetry in the problem, only a quarter of the domain is discretised with appropriate free and symmetric boundary conditions.

Since the cylindrical bar comes in contact with the rigid surface at time $t = 0$, stress near the contact region surpasses the elastic limit and the bar undergoes plastic deformation. At the

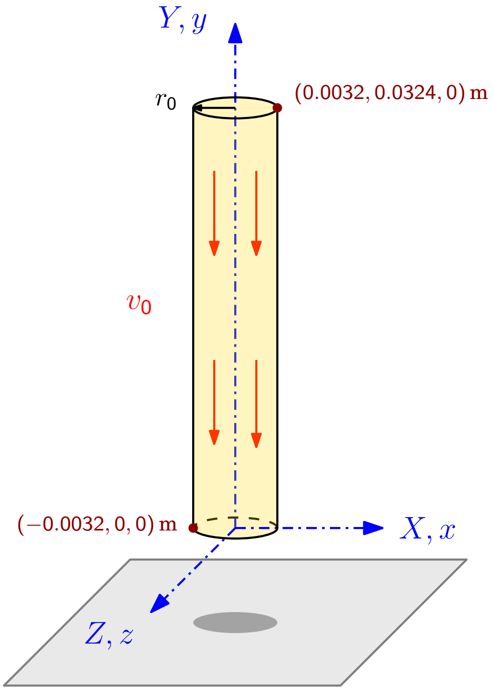


Figure 8.45: Taylor impact: Problem setup

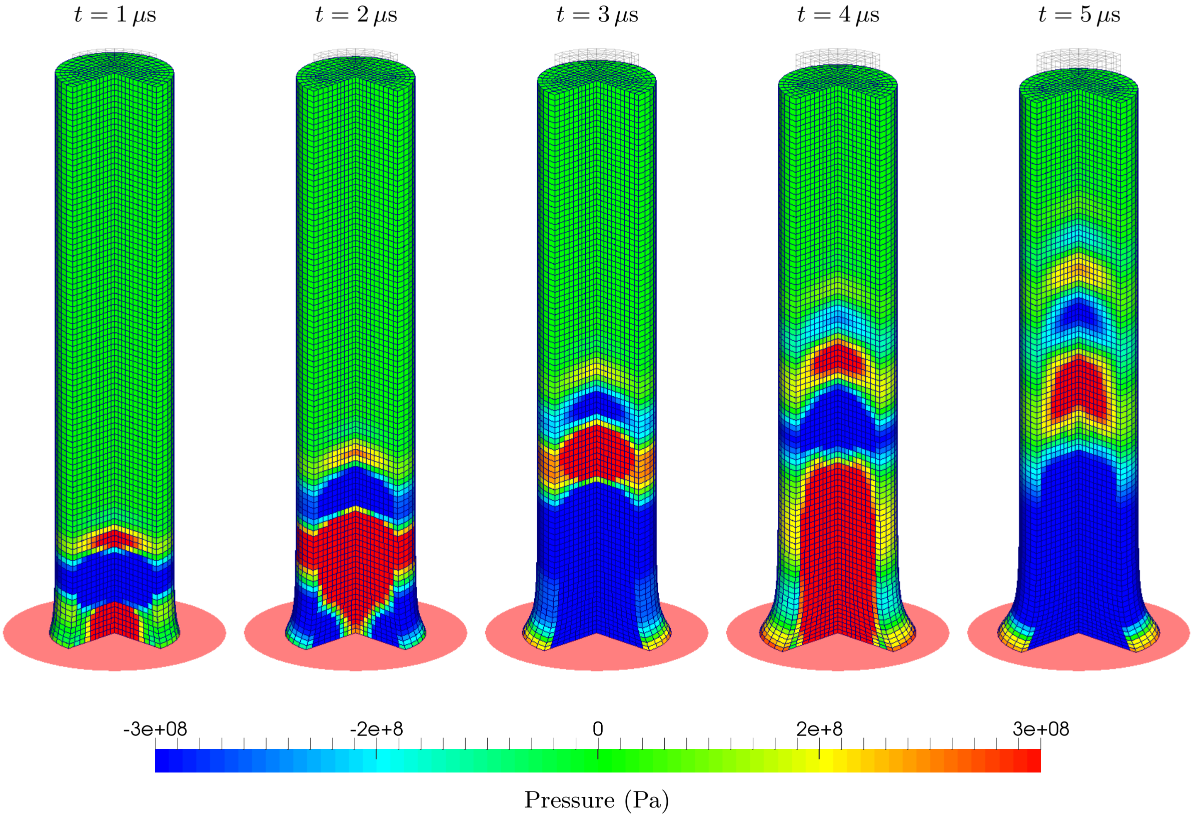
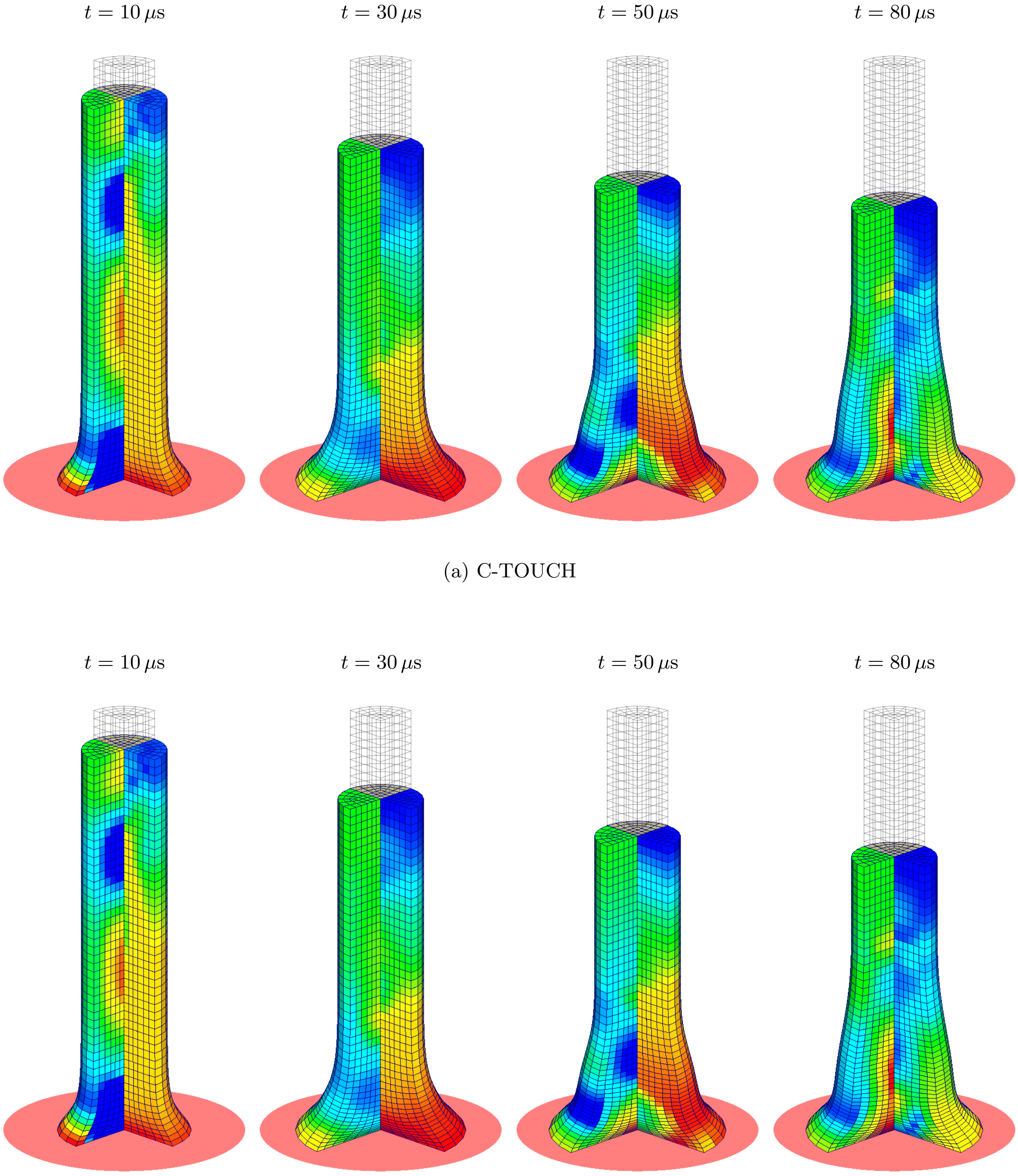


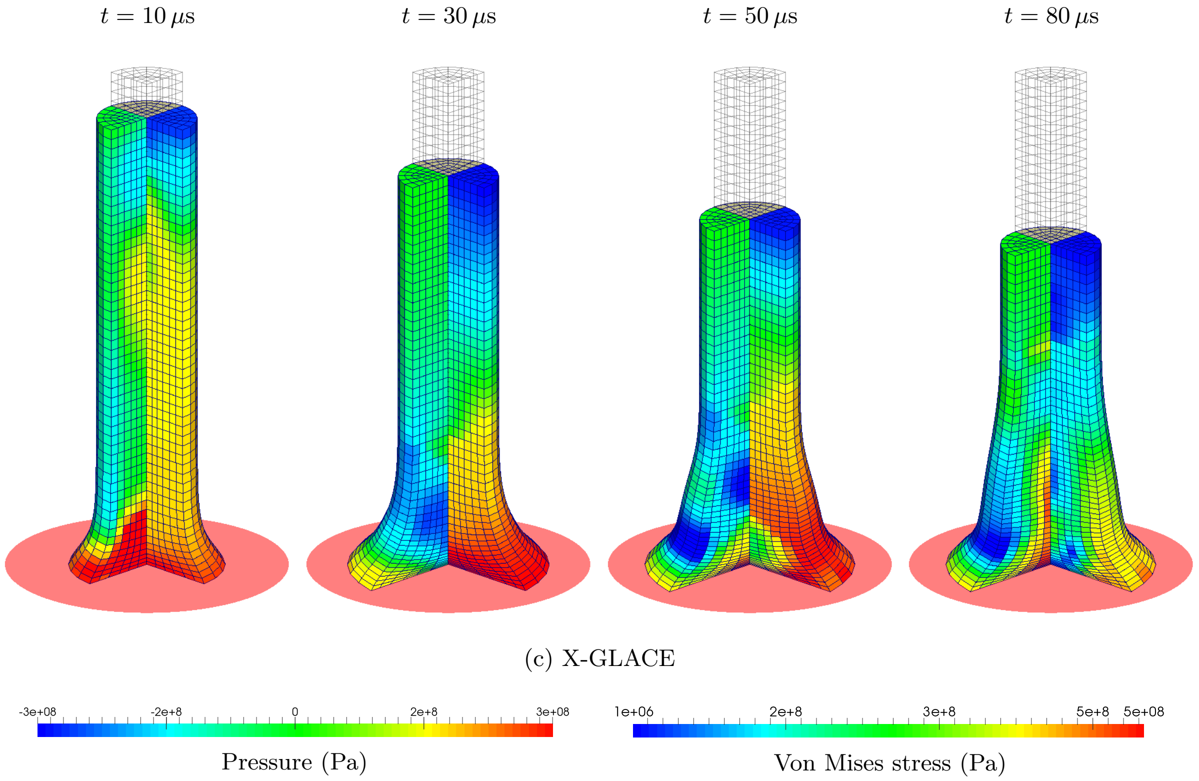
FIGURE 8.46: Taylor impact: Time evolution of pressure wave along with deformation in half domain during the initial stages of impact. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with a velocity $\boldsymbol{v}_0 = [0, -227, 0]^T$ m/s and a discretisation of 7500 structured hexehedral elements. A von Mises hyperelastic-plastic material is used with $\rho_0 = 8930$ kg/m^3, $E = 117$ GPa, $\nu = 0.35$, yield stress $\bar{\tau}_y^0 = 0.4$ GPa, hardening modulus $H = 0.1$ GPa and $\alpha_{\text{CFL}} = 0.3$.

same time an elastic pressure (compression) wave starts travelling to the far end (see Fig. 8.46). When this pressure wave reaches the far end of the bar, it gets reflected back as an expansion wave which gets superimposed on the compression wave. The pressure distribution obtained is clearly very smooth without the appearance of any oscillations. In Fig. 8.47 the pressure and von Mises stress distribution can be seen using C-TOUCH, P-TOUCH and X-GLACE schemes. It can be seen that the results obtained are very similar. Moreover, in Fig. 8.48 a mesh refinement study has been done using 480, 1350 and 7500 hexahedral cells for the C-TOUCH scheme. In Fig. 8.49 a comparison of the time evolution of bar radius at the location $\boldsymbol{X} = [0.0032, 0, 0]^T$ m is plotted for the three cell centred $\{\boldsymbol{p}, \boldsymbol{F}\}$ finite volume schemes. It can be seen that for a coarse mesh of 480 elements (dashed lines), the C-TOUCH and P-TOUCH schemes give very similar results where as the X-GLACE scheme under predicts the radius. However, when a finer mesh of 1350 elements is utilised, all three schemes converge to the same result. It is a common practice when simulating this problem to monitor the final radius of the bar. For comparison purposes Table 8.1 summarises the final radii available in literature using various numerical methodologies.



(a) C-TOUCH

(b) P-TOUCH ($\xi_{\boldsymbol{F}} = 0.1$)

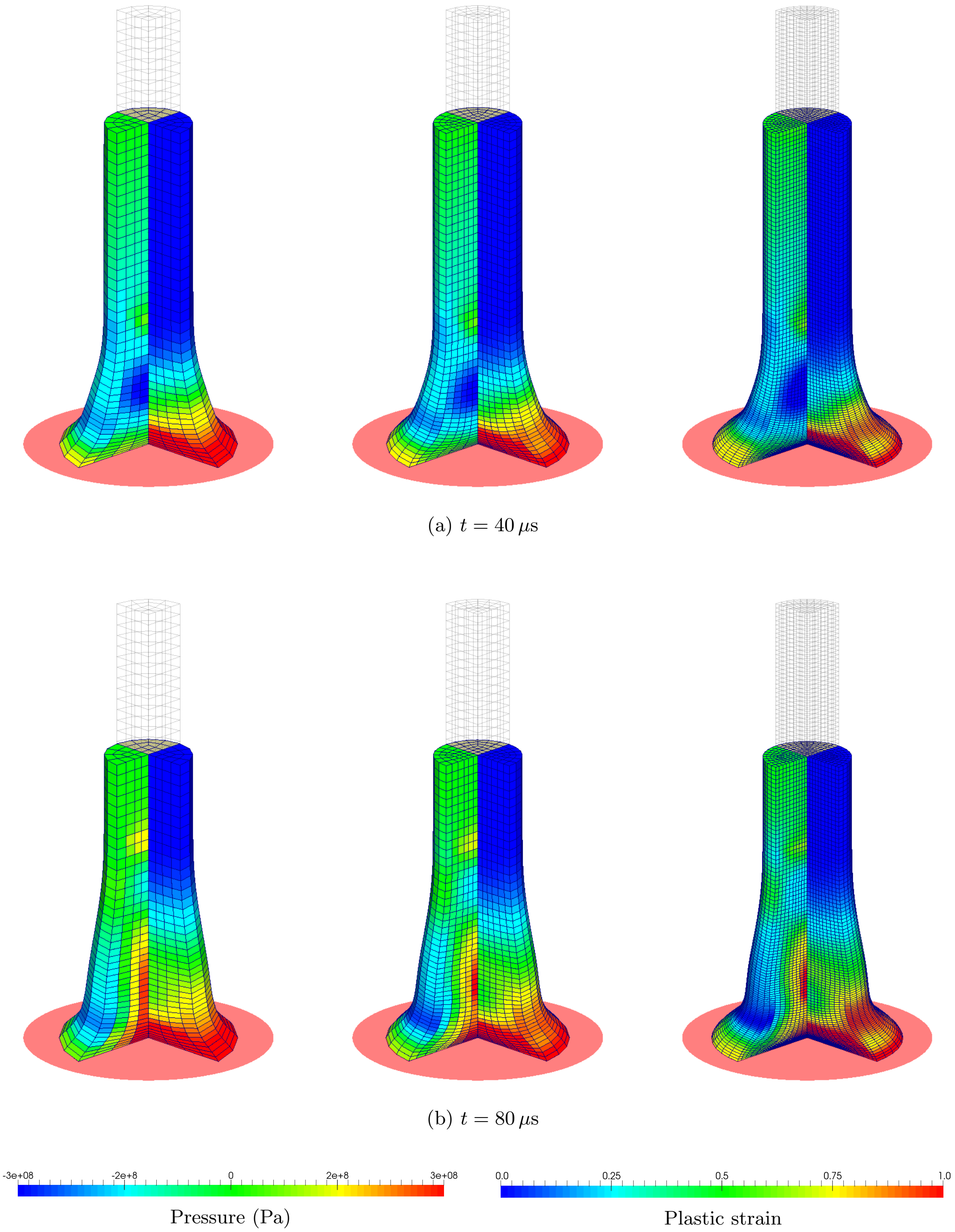


(c) X-GLACE

FIGURE 8.47: Taylor impact: Time evolution of pressure distribution in the left quarter and von Mises stresses in the right quarter of the domain along with the deformation. Results obtained using (a) $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH; (b) $\{\boldsymbol{p}, \boldsymbol{F}\}$ P-TOUCH ($\xi_{\boldsymbol{F}} = 0.1$); and (c) $\{\boldsymbol{p}, \boldsymbol{F}\}$ X-GLACE schemes with velocity $\boldsymbol{v}_0 = [0, -227, 0]^T$ m/s and a discretisation of 1350 structured hexehedral elements in quarter domain. A von Mises hyperelastic-plastic material is used with $\rho_0 = 8930$ kg/m^3, $E = 117$ GPa, $\nu = 0.35$, yield stress $\bar{\tau}_y^0 = 0.4$ GPa, hardening modulus $H = 0.1$ GPa and $\alpha_{\text{CFL}} = 0.3$.

Methodology	**Elements**	**Final radius (mm)**
Standard FEM [40]	4-Node tet	5.55
Standard FEM [40]	8-Node hex	6.95
Newmark FEM [137]	hex (972)	≈ 7.00
Average nodal pressure FEM [40]	4-Node tet	6.99
Split FEM [140]	tet	7.07-7.33
CC-FVM [66]	—	7.14
Mixed FEM [140]	hex	7.11
Mixed JST VC-FVM [10]	4-Node tet	6.98
Mixed PG-FEM [15]	4-Node tet	7.00
Mixed CC-FVM [44] [8]	hex (480)	6.88-7.11

TABLE 8.1: Taylor impact: Comparison of final radii at $t = 80\,\mu$s obtained from various numerical methodologies.



(a) $t = 40\,\mu$s

(b) $t = 80\,\mu$s

FIGURE 8.48: Taylor impact: Mesh refinement of deformed shapes with pressure distribution in the left quarter and plastic strain distribution in the right quarter of the domain. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with velocity $\boldsymbol{v}_0 = [0, -227, 0]^T$ m/s and a discretisation of 480, 1350 and 7500 hexehedral elements in quarter domain. A von Mises hyperelastic-plastic material is used with $\rho_0 = 8930$ kg/m^3, $E = 117$ GPa, $\nu = 0.35$, yield stress $\bar{\tau}_y^0 = 0.4$ GPa, hardening modulus $H = 0.1$ GPa and $\alpha_{\text{CFL}} = 0.3$.

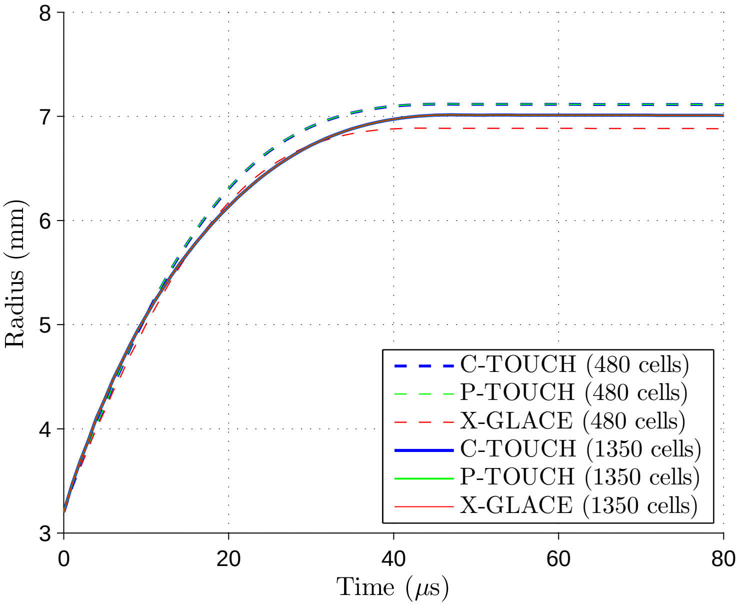


FIGURE 8.49: Taylor impact: Evolution of radius at the material point $\boldsymbol{X} = [3.2, 0, 0]^T$ mm using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ cell centred methodologies, namely; (a) C-TOUCH; (b) P-TOUCH ($\xi_{\boldsymbol{F}} = 0.1$); and (c) X-GLACE schemes. Results obtained with velocity $\boldsymbol{v}_0 = [0, -227, 0]^T$ m/s and a discretisation of 480 and 1350 hexehedral elements in quarter domain. A von Mises hyperelastic-plastic material is used with $\rho_0 = 8930$ kg/m^3, $E = 117$ GPa, $\nu = 0.35$, yield stress $\bar{\tau}_y^0 = 0.4$ GPa, hardening modulus $H = 0.1$ GPa and $\alpha_{\text{CFL}} = 0.3$.

[44] This result includes the following cell centred $\{\boldsymbol{p}, \boldsymbol{F}\}$ finite volume methodologies; namely (a) C-TOUCH; (b) P-TOUCH ($\xi_{\boldsymbol{F}} = 0.1$); and (c) X-GLACE scheme of [5].

Chapter 9

COMPLEX PROBLEMS

In this chapter, more challenging problems are presented both in terms of the physics involved and complexity of the computational domain. In Section 9.1 contact problems are presented where it is assumed that one body comes into contact with another rigid planar body. Moreover, in Section 9.2, robustness of the proposed cell centred finite volume methodologies is shown on complex geometries. The roadmap to this chapter is presented in Fig. 8.1.

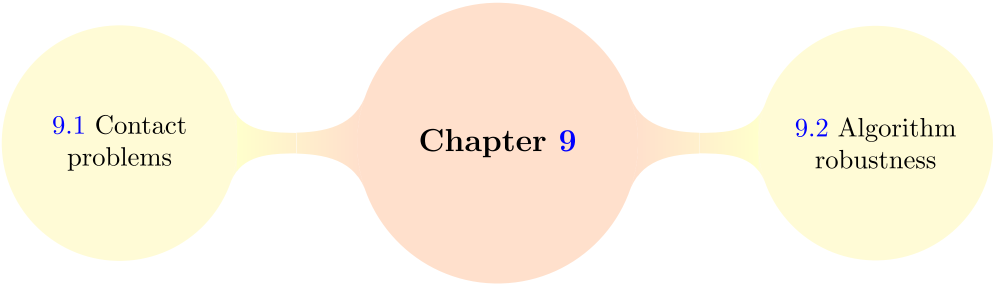


FIGURE 9.1: Structure of Chapter 9.

9.1 Contact problems

9.1.1 Ring impact

This is a classical benchmark example in SPH used to study the effects of tensile instability. We investigate the collision of a rubber ring with an inner radius of 30 mm and an outer radius of 40 mm against a rigid wall (see Fig. 9.2). Swegle [143] originally proposed this problem demonstrating the possible fracturing that can occur in an SPH code. Later this problem was also investigated in [69, 74, 139, 144]. The ring has an initial velocity $v_0 = [0, -0.59, 0]^T$ m/s and is placed 0.004 m away from the wall. Upon impacting the wall at $t = 0.004/0.59 = 0.00678$ s, the outer part of the ring suffers from compression while the inner part experiences tensile forces. A nearly incompressible neo-Hookean constitutive model is used with material properties density $\rho_0 = 1000$ kg/m^3, Youngs's modulus $E = 1$ MPa and Poisson's ratio $\nu = 0.4$.

Fig. 9.3 shows the sequence of deformed states along with a smooth pressure distribution as no instabilities are encountered during and after the impact. The evolution of global angular momentum is plotted in Fig. 9.4a where, as expected, the y and z components of angular momentum are zero. Moreover, in Fig. 9.4b global linear momentum is also plotted against time and it is observed that y component of linear momentum is non zero. A mesh refinement study has also been carried out in Fig. 9.4 which shows that very similar results are obtained for a coarse (420 cells) and a fine (25760 cells) mesh. Finally, Fig. 9.5 displays the deformed state of the ring at time $t = 0.18$ s utilising meshes of 420 and 6400 structured hexahedral elements.

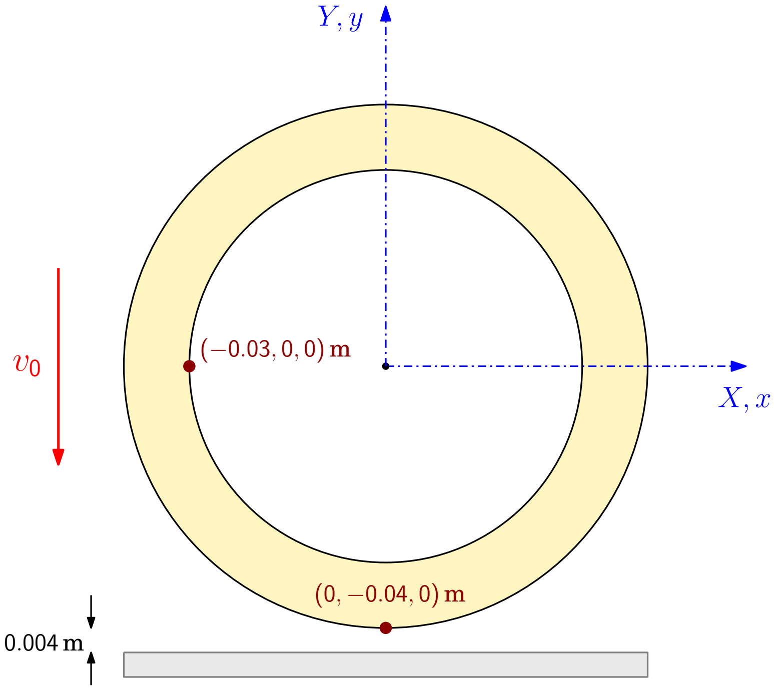


FIGURE 9.2: Ring impact: Problem setup.

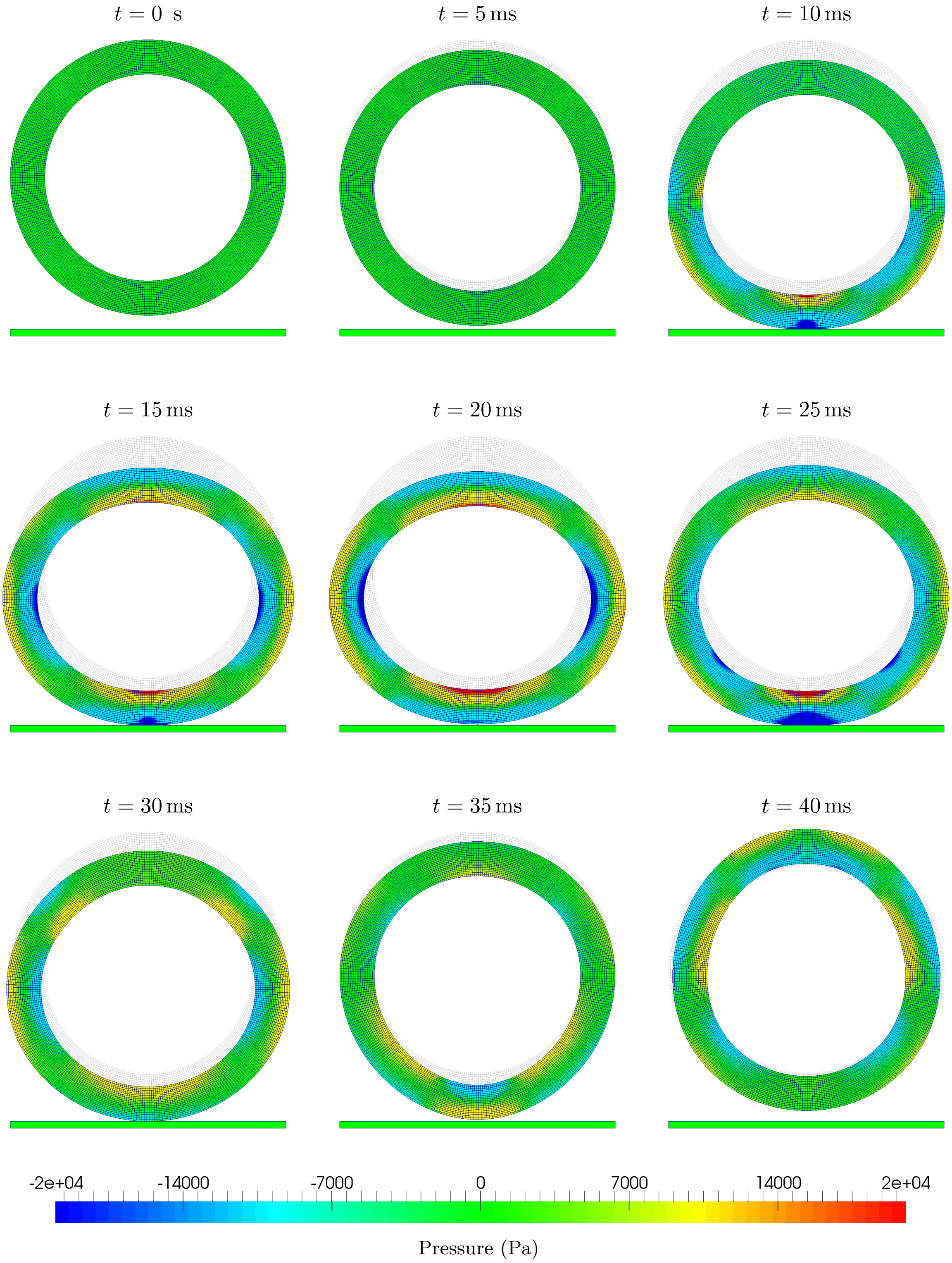
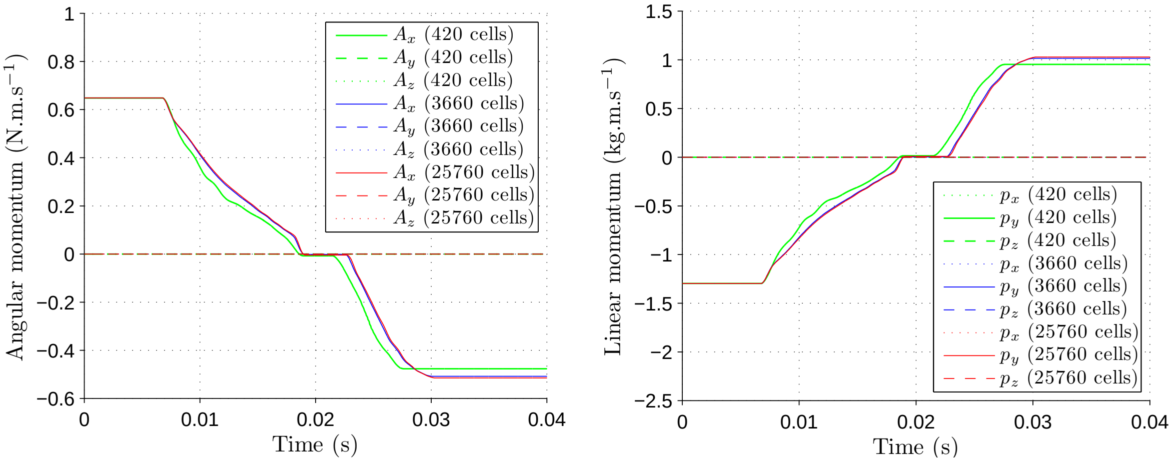


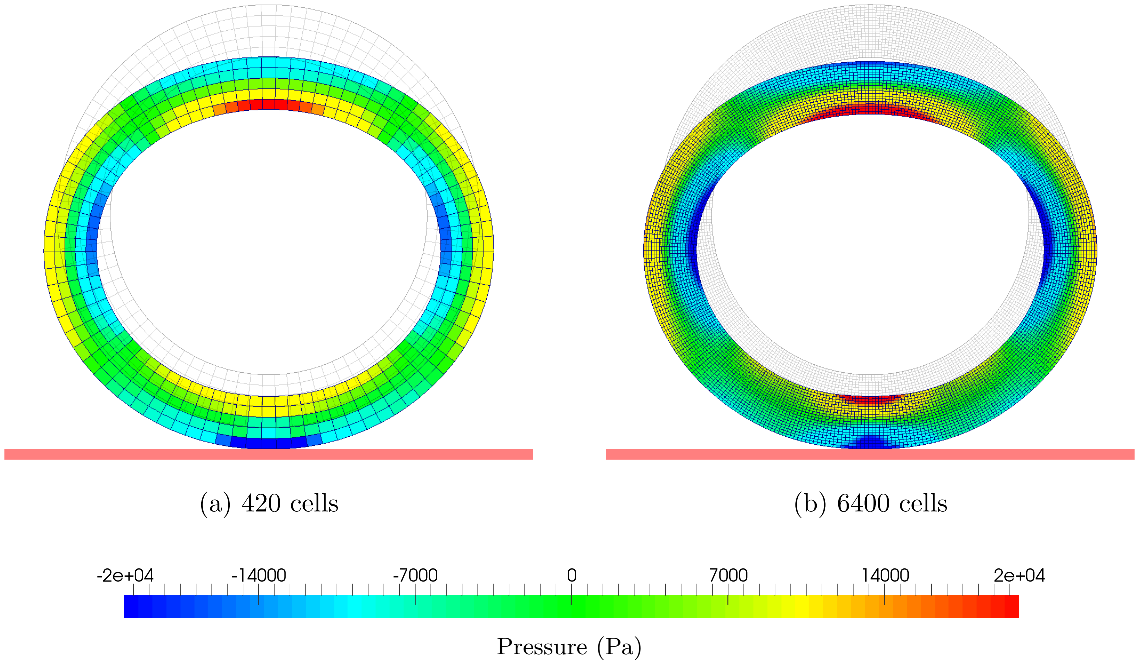
FIGURE 9.3: Ring impact: Time evolution of deformation plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. Results obtained with velocity $\boldsymbol{v}_0 = [0, -0.59, 0]^T$ m/s and a discretisation of 6480 hexahedral elements. A neo-Hookean material is used with $\rho_0 = 1000$ kg/m^3, $E = 1$ MPa, $\nu = 0.4$, $\alpha_{\text{CFL}} = 0.3$ and $\varDelta t \approx 3 \times 10^{-6}$ s.



(a) Angular momentum

(b) Linear momentum

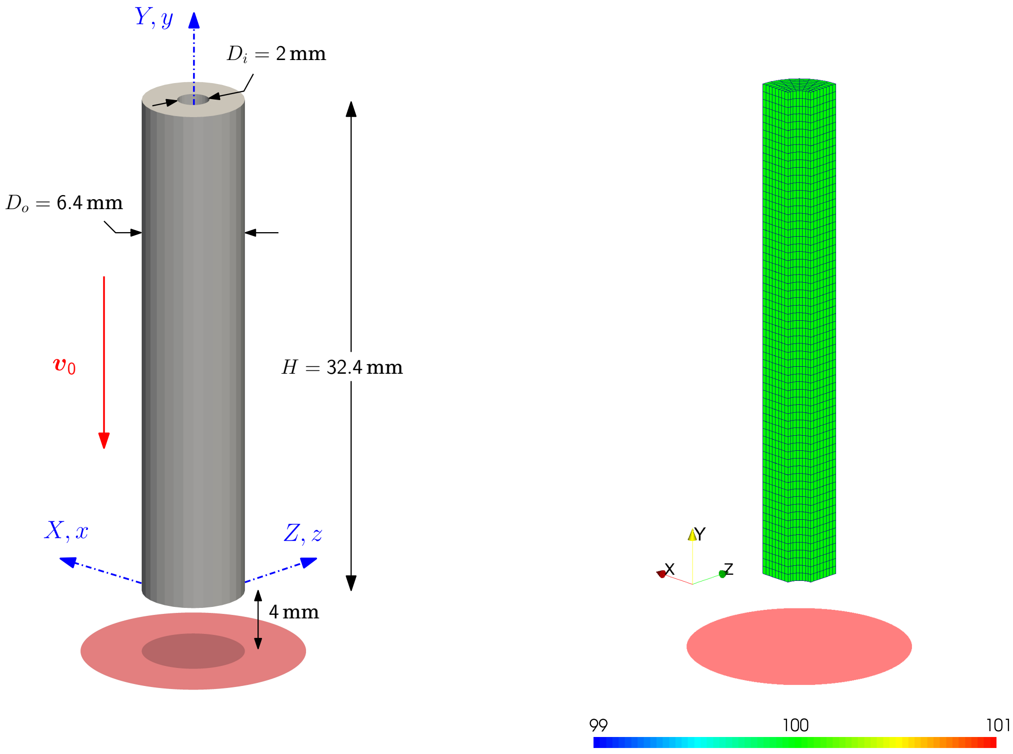
FIGURE 9.4: Ring impact: Time evolution of components of (a) global angular momentum; and (b) global linear momentum using various mesh sizes. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH contact algorithm with velocity $\boldsymbol{v}_0 = [0, -0.59, 0]^T$ m/s. A neo-Hookean material is used with density $\rho_0 = 1000$ kg/m^3, Young's modulus $E = 1$ MPa, Poisson's ratio $\nu = 0.4$ and $\alpha_{\text{CFL}} = 0.3$.



(a) 420 cells

(b) 6400 cells

FIGURE 9.5: Ring impact: Mesh refinement of deformed shapes with pressure distribution at $t = 0.18$ s using meshes comprising of: (a) 420; and (b) 6400 structured hexahedral cells. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme with velocity $\boldsymbol{v}_0 = [0, -0.59, 0]^T$ m/s. A neo-Hookean material is used with $\rho_0 = 1000$ kg/m^3, $E = 1$ MPa, $\nu = 0.4$ and $\alpha_{\text{CFL}} = 0.3$.



(a) Initial configuration

(b) Initial velocity (m/s) and mesh

FIGURE 9.6: Bar rebound: Problem setup

9.1.2 Bar rebound

Previously explored in [145] in two dimensions, an extension of the plate rebound contact example to three dimensions is carried out by considering the rebound of a hollow circular bar of outer diameter $D_0 = 6.4\,\mathrm{mm}$, inner diameter $D_i = 2\,\mathrm{mm}$ and height $H = 32.4\,\mathrm{mm}$ (see Fig. 9.6). The bar impacts against a rigid frictionless wall with an initial velocity $\boldsymbol{v}_0 = [0, -100, 0]^T$ m/s where the separation distance between the bar and wall is 4 mm. Upon impact the bar undergoes large compressive deformation until $t = 150$ μs when all the kinetic energy of the bar is converted to potential energy. Soon afterwards, tensile forces start developing and a bounce-off motion begins. At approximately $t = 250$ μs the bar completely detaches itself from the wall and continues to deform. The material parameters used are density $\rho_0 = 8930$ kg/m^3, Young's modulus $E = 585$ MPa and Poisson's ratio $\nu = 0.45$. Due to the existence of two symmetry planes, only a quarter of the domain is simulated with appropriate symmetric roller supports and free boundary conditions.

A sequence of deformation of the hollow bar plotted with pressure distribution is shown in Fig. 9.7. No spurious pressure instabilities can be observed. In Fig. 9.8, a mesh refinement study is carried out with successive hexahedral meshes of 512, 4096 and 13824 elements. It is remarkable that the deformation obtained with the coarse mesh agrees extremely well with the fine mesh. Pressure contour is clearly enhanced as we refine the mesh density. In Fig. 9.9 and Fig. 9.10, two varying hexahedral meshes of 512 and 13824 are used. Fig. 9.9 shows time evolution of the global linear and angular momentum, whereas Fig. 9.10 shows time history of vertical y

displacement at the bottom plane $\boldsymbol{X}_B = \left[1/\sqrt{2}, 0, 1/\sqrt{2}\right]^T$mm (denoted by red lines) and at the top plane $\boldsymbol{X}_T = \left[1/\sqrt{2}, 32.4, 1/\sqrt{2}\right]^T$mm (denoted by blue lines). The difference between the red and blue lines indicate the amount of elongation/reduction in bar length. Reasonably accurate displacements are obtained using a coarse mesh, showing optimal convergence for the proposed method. Finally, we further examine this problem using a large value of Poisson's ratio $\nu = 0.499$. Fig. 9.11 shows a series of deformed states along with pressure, without displaying any numerical difficulties.

9.1.3 Torus impact

We investigate the impact of a rubber torus with an initial inner radius $R_i = 30$ mm and an outer radius of $R_o = 40$ mm against a rigid wall. The torus has an initial velocity $\boldsymbol{v}_0 = [0, -3, 0]^T$ m/s and is placed 4 mm away from the wall. Upon impact the outer part of the torus suffers from compression while the inner part experiences tensile forces. A neo-Hookean constitutive model is used with material properties density $\rho_0 = 1000$ kg/m^3, Youngs's modulus $E = 1$ MPa and Poisson's ratio $\nu = 0.4$. Fig. 9.12 shows the deformation along with the smooth pressure distribution.

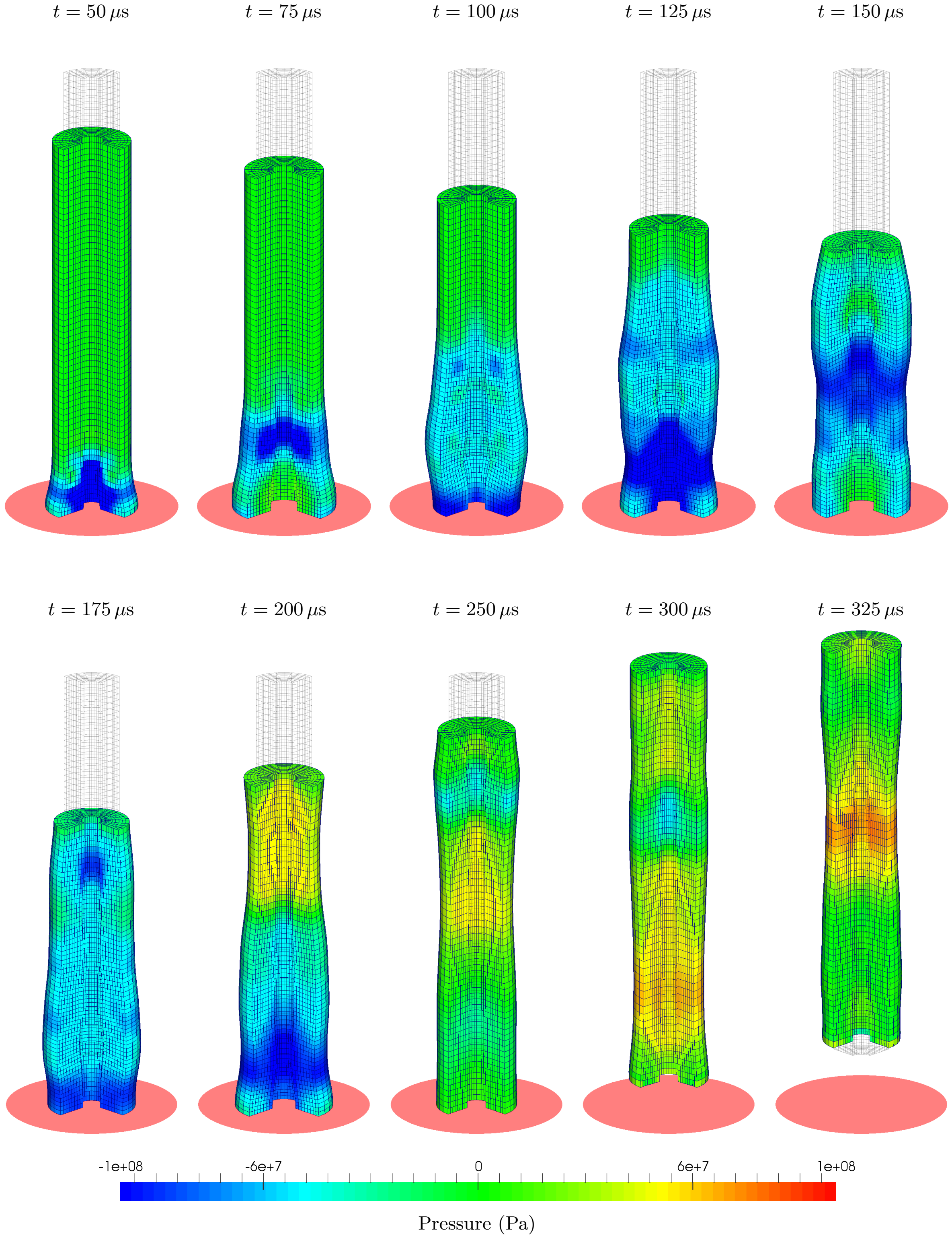
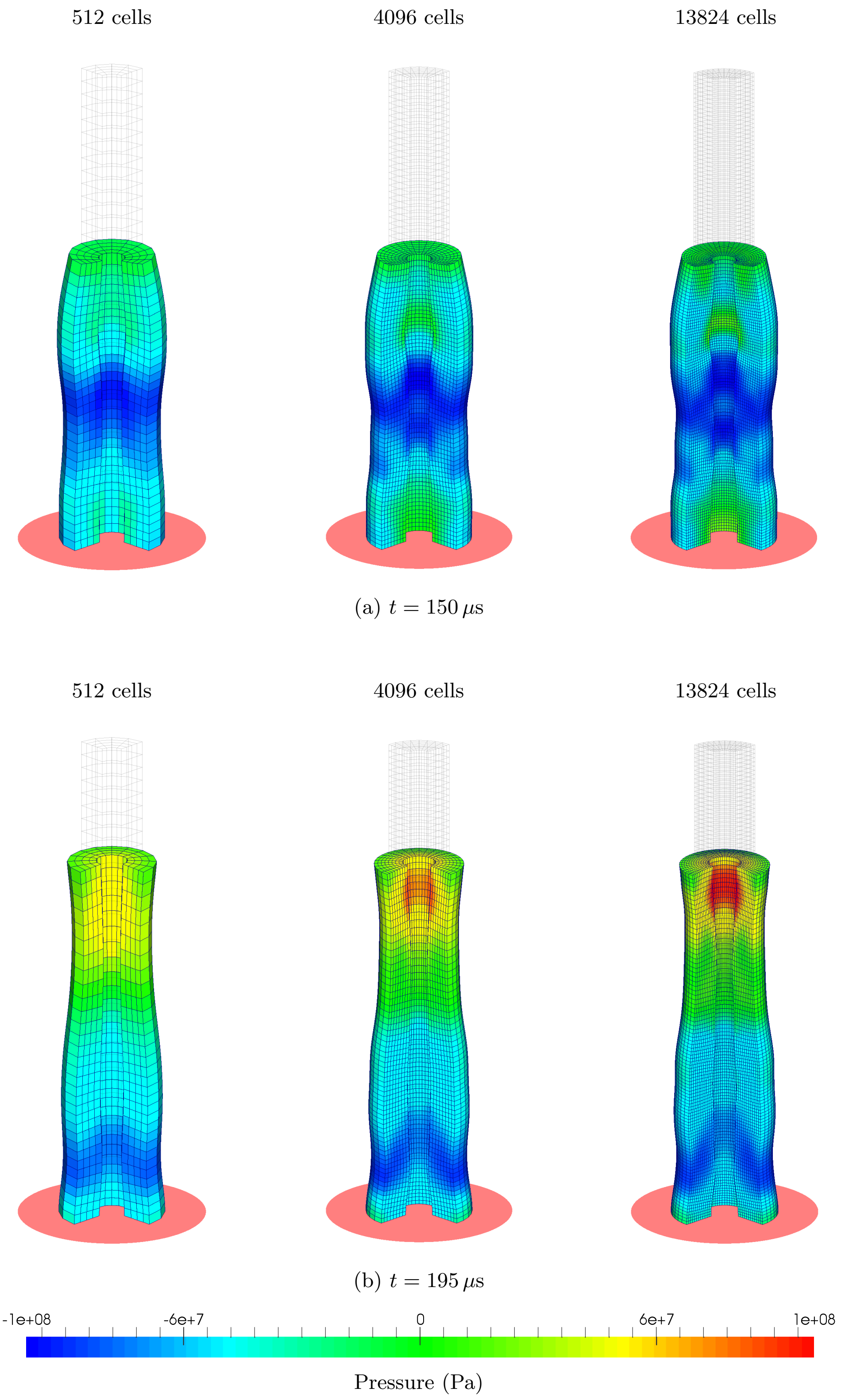


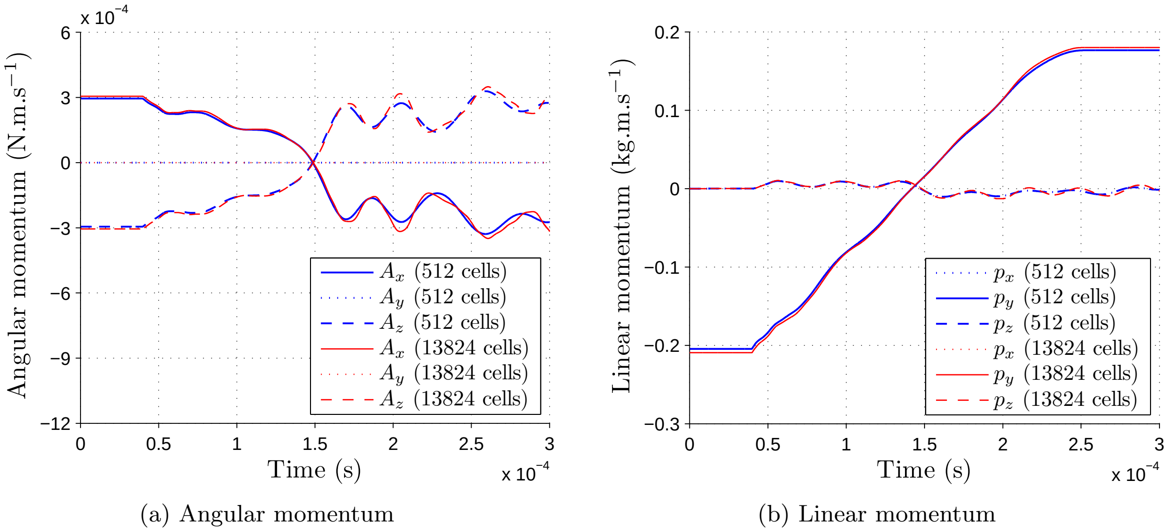
FIGURE 9.7: Bar rebound: Time evolution of the deformation along with the pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. Results obtained with velocity $\boldsymbol{v}_0 = [0, -100, 0]^T$ m/s using 4096 hexahedral elements in quarter domain. A neo-Hookean constitutive model is used with $\rho_0 = 8930$ kg/m^3, $E = 585$ MPa, $\nu = 0.45$, $\alpha_{\rm CFL} = 0.3$ and $\Delta t \approx 7 \times 10^{-8}$ s.



(a) $t = 150\,\mu$s

(b) $t = 195\,\mu$s

FIGURE 9.8: Bar rebound: Mesh refinement of deformed shapes along with pressure distribution at times: (a) $t = 150\,\mu$s; and (b) $t = 195\,\mu$s using meshes of 512, 4096 and 13824 hexahedral elements in quarter domain. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with velocity $\boldsymbol{v}_0 = [0, -100, 0]^T$ m/s. A neo-Hookean material is used with $\rho_0 = 8930$ kg/m^3, $E = 585$ MPa, $\nu = 0.45$ and $\alpha_{\rm CFL} = 0.3$.



(a) Angular momentum

(b) Linear momentum

Figure 9.9: Bar rebound: Time evolution of components of (a) global angular momentum; and (b) global linear momentum using meshes of 512 and 13824 hexahedral elements in quarter domain. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with velocity $\boldsymbol{v}_0 = [0, -100, 0]^T$ m/s. A neo-Hookean constitutive model is used with $\rho_0 = 8930$ kg/m^3, $E = 585$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

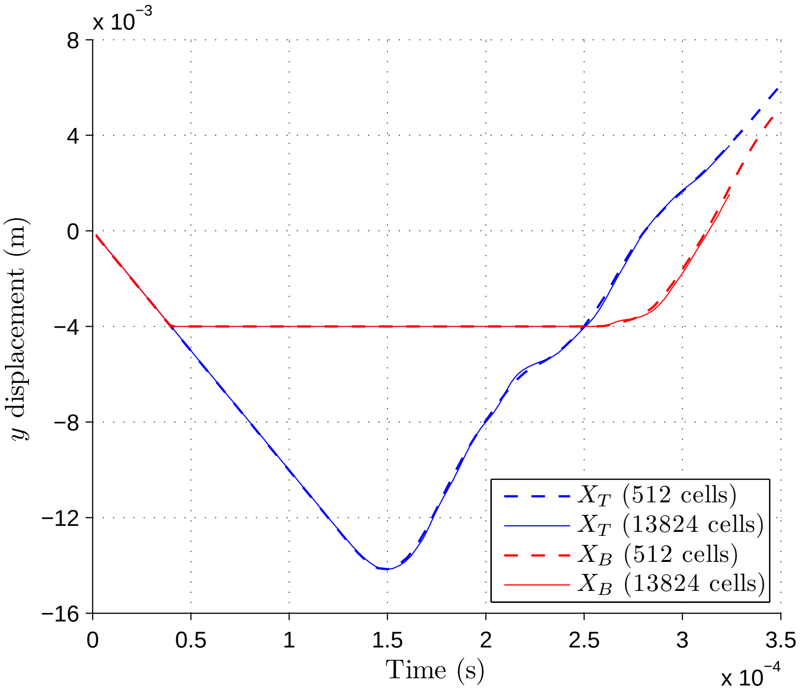


Figure 9.10: Bar rebound: Time evolution of vertical y displacements u_y of the points on the top plane $\boldsymbol{X}_T = \left[1/\sqrt{2}, 32.4, 1/\sqrt{2}\right]^T$mm and the bottom plane $\boldsymbol{X}_B = \left[1/\sqrt{2}, 0, 1/\sqrt{2}\right]^T$mm. Results obtained with velocity $\boldsymbol{v}_0 = [0, -100, 0]^T$ m/s using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. A neo-Hookean constitutive model is used with $\rho_0 = 8930$ kg/m^3, $E = 585$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

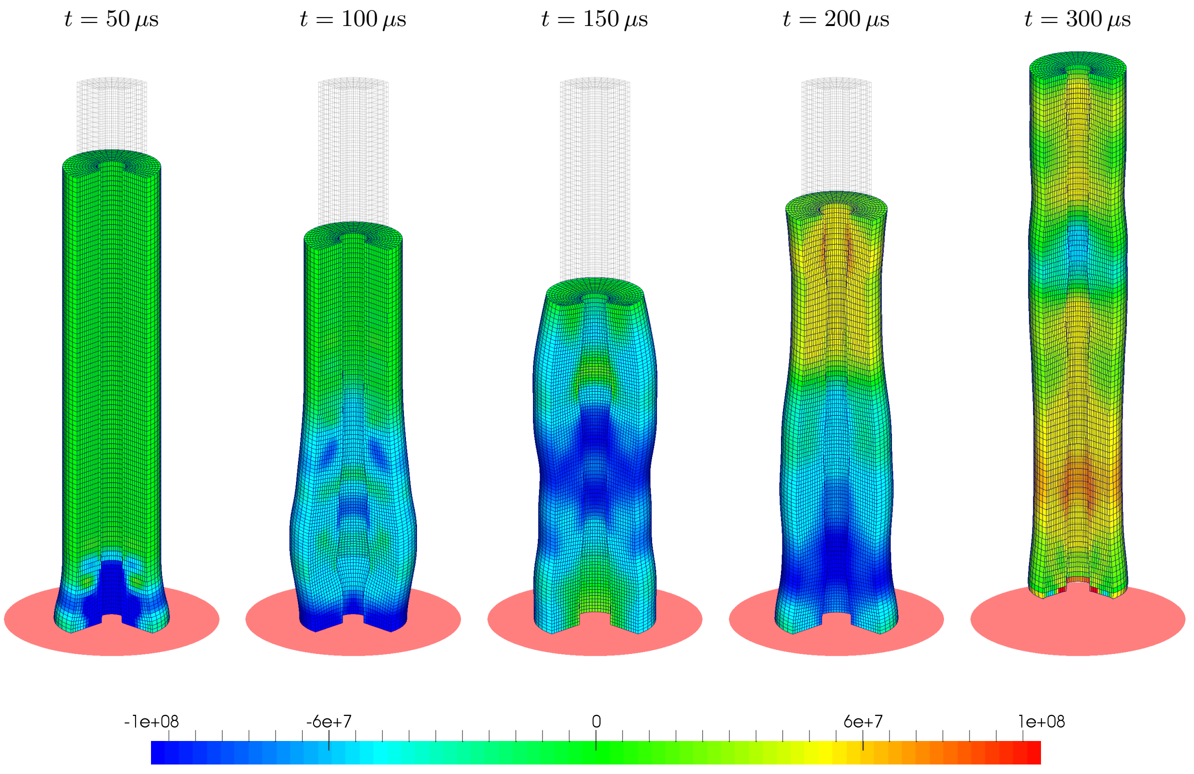


FIGURE 9.11: Bar rebound: Time evolution of the deformation along with the pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme ($\tilde{\kappa} = 3\kappa$). Results obtained with velocity $\boldsymbol{v}_0 = [0, -100, 0]^T$ m/s using 4096 hexahedral elements. A neo-Hookean constitutive model is used with $\rho_0 = 8930$ kg/m^3, $E = 585$ MPa, $\nu = 0.499$, $\alpha_{\text{CFL}} = 0.3$ and $\varDelta t \approx 5 \times 10^{-8}$ s.

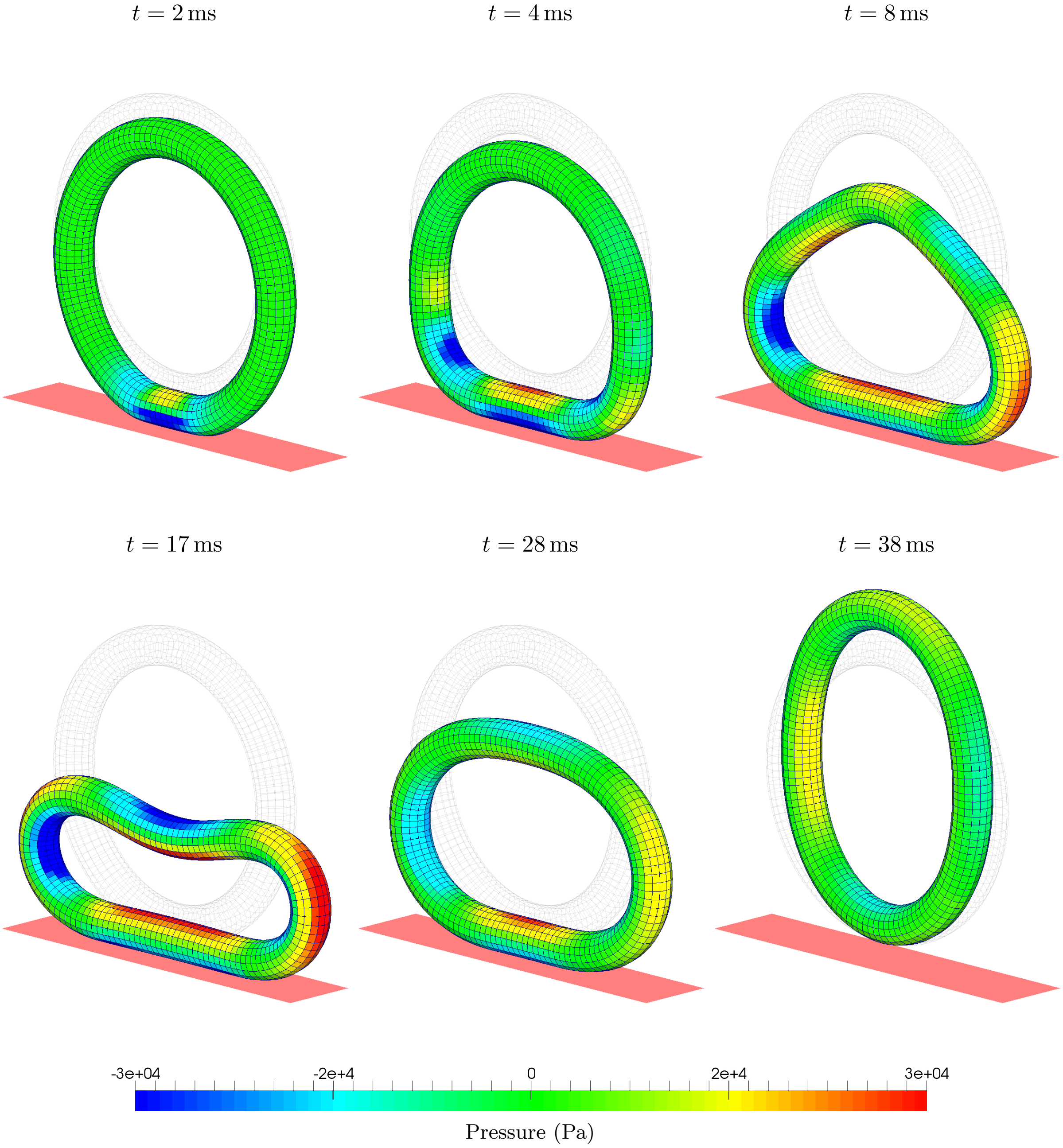


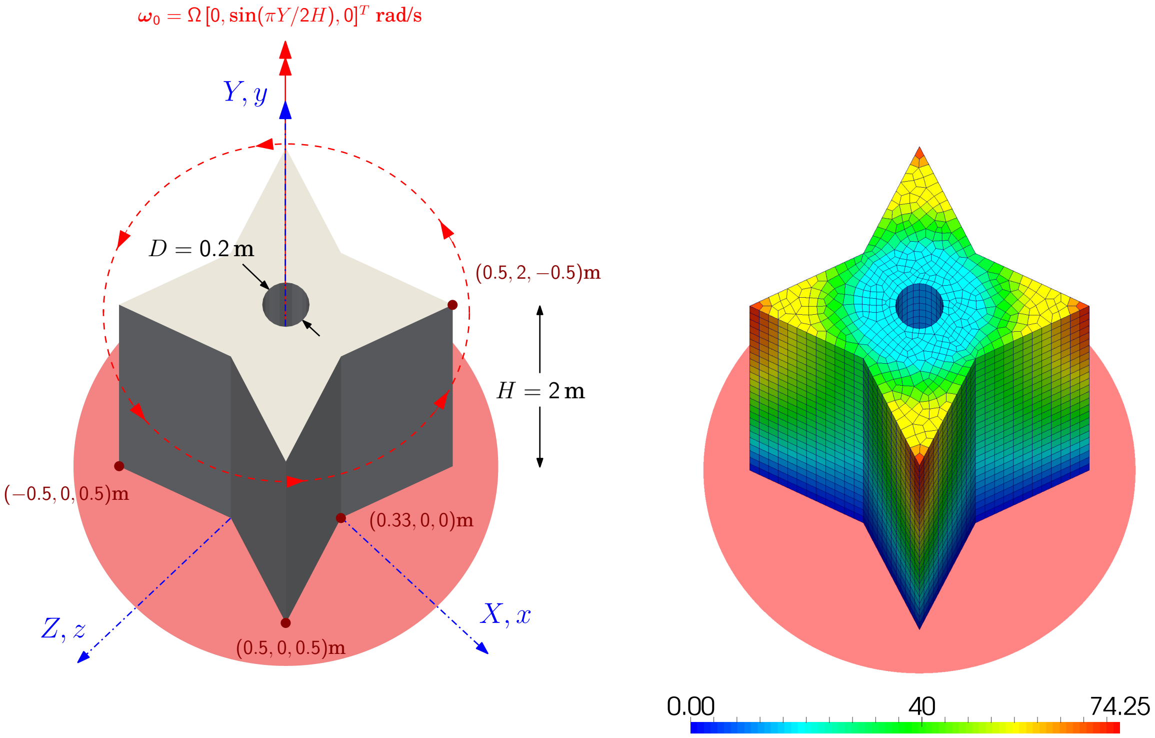
FIGURE 9.12: Torus impact: Time evolution of the deformation plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}\}$ C-TOUCH scheme. Results obtained with velocity $\boldsymbol{v}_0 = [0, -3, 0]^T$ m/s and a discretisation of 10400 hexahedral cells. A neo-Hookean material is used with $\rho_0 = 1000$ kg/m^3, $E = 1$ MPa, $\nu = 0.4$, $\alpha_{\text{CFL}} = 0.3$ and $\Delta t \approx 3 \times 10^{-6}$ s.

9.2 Algorithm robustness

9.2.1 Complex twisting

In this section, an alternative version of the twisting column example (see Section 8.4.2) is presented to check the robustness of the algorithm on a more complicated geometry. The column has a star shaped cross section with the furthest point $\sqrt{0.5}$ m and the nearest point 0.33 m away from the central Y axis Fig. 9.13a. It is 2 m high and has a hole of diameter 0.2 m along the vertical axis. The column is initialised with an angular velocity given by $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 105$ rad/s and $H = 2$ m (see Fig. 9.13b). A neo-Hookean material is used with density $\rho_0 = 1100$ kg/m^3, Young's Modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.45$ (unless otherwise stated).

Fig. 9.14 shows the evolution of deformation and pressure distribution in the domain using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH algorithm. Locking-free deformation of the column can be observed, without the appearance of any spurious pressure oscillations. In Fig. 9.15, a comparison of the proposed cell centred finite volume methodologies: (a) C-TOUCH; (b) P-TOUCH with penalisation parameters ($\xi_{\{\boldsymbol{F},\boldsymbol{H},J\}} = 0.1$); and (c) X-GLACE schemes is shown. The three methodologies produce very similar results in terms of deformation and pressure distribution. Moreover, in Fig. 9.16 a mesh refinement study is carried out at an increased Poisson's ratio $\nu = 0.499$ with $\tilde{\kappa} = 3\kappa$. It is clearly shown that no locking or pressure instabilities are encountered even in this high incompressibility regime. Moreover, once again it is proven that a coarse mesh of 13225 elements produces practically identical results in terms of deformation as compared to the finer mesh (68350 elements).


$\omega_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s
Y, y
$D = 0.2\,\mathrm{m}$
$(0.5, 2, -0.5)\mathrm{m}$
$H = 2\,\mathrm{m}$
$(-0.5, 0, 0.5)\mathrm{m}$
$(0.33, 0, 0)\mathrm{m}$
Z, z
X, x
$(0.5, 0, 0.5)\mathrm{m}$
0.00
40
74.25

(a) Initial configuration

(b) Initial velocity in X-Z plane (m/s)

FIGURE 9.13: Complex twisting: Problem setup.

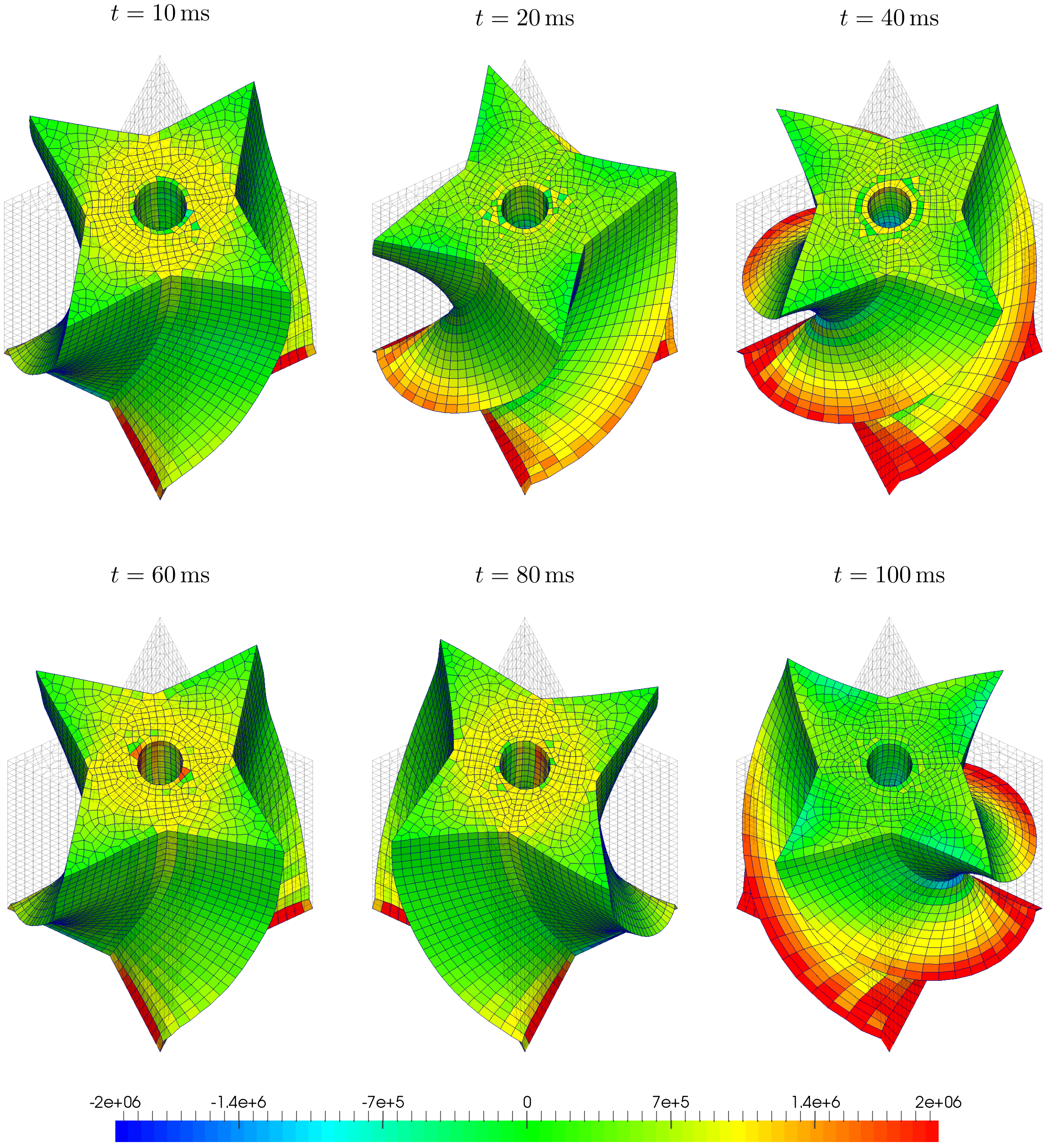
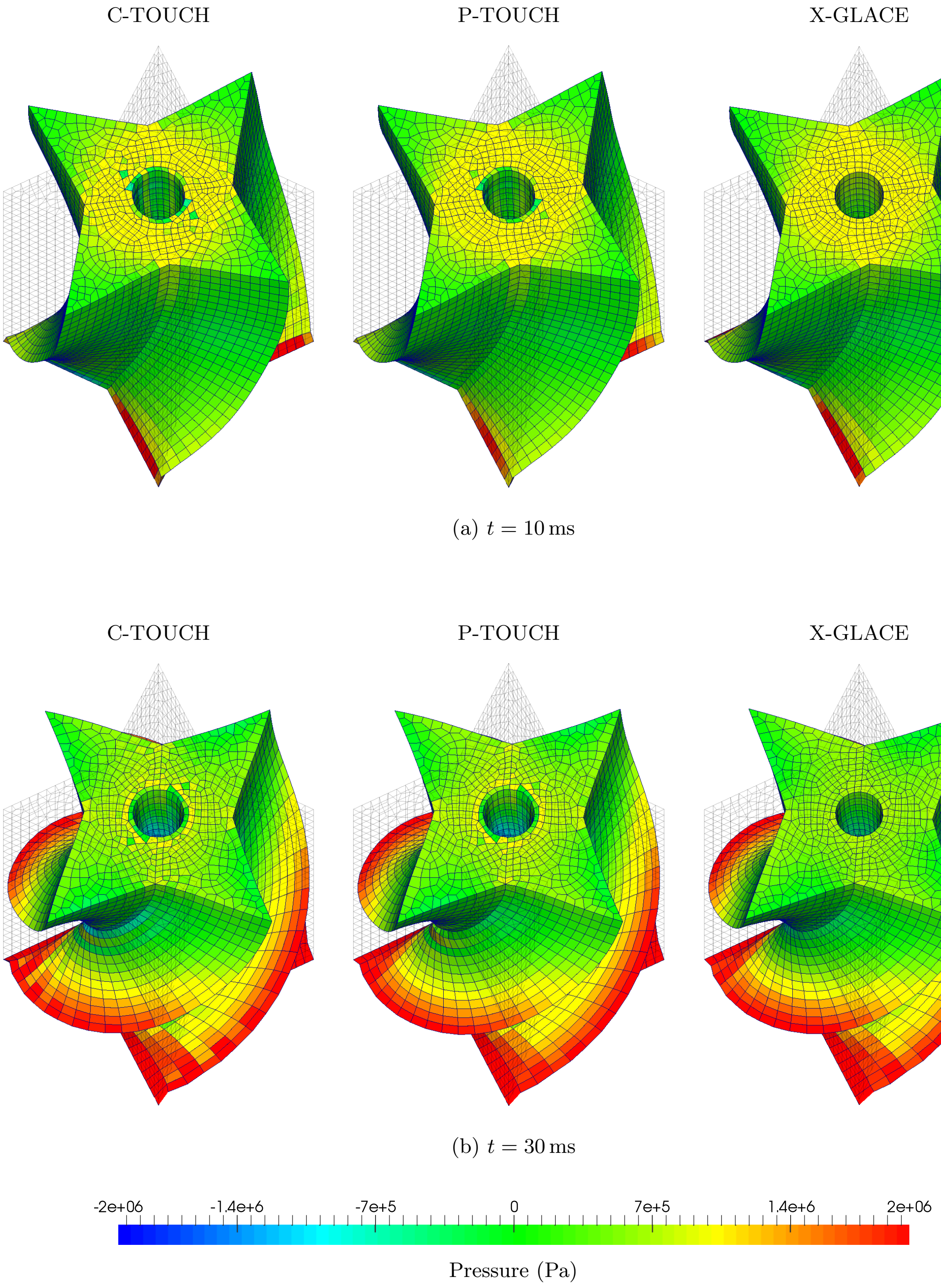


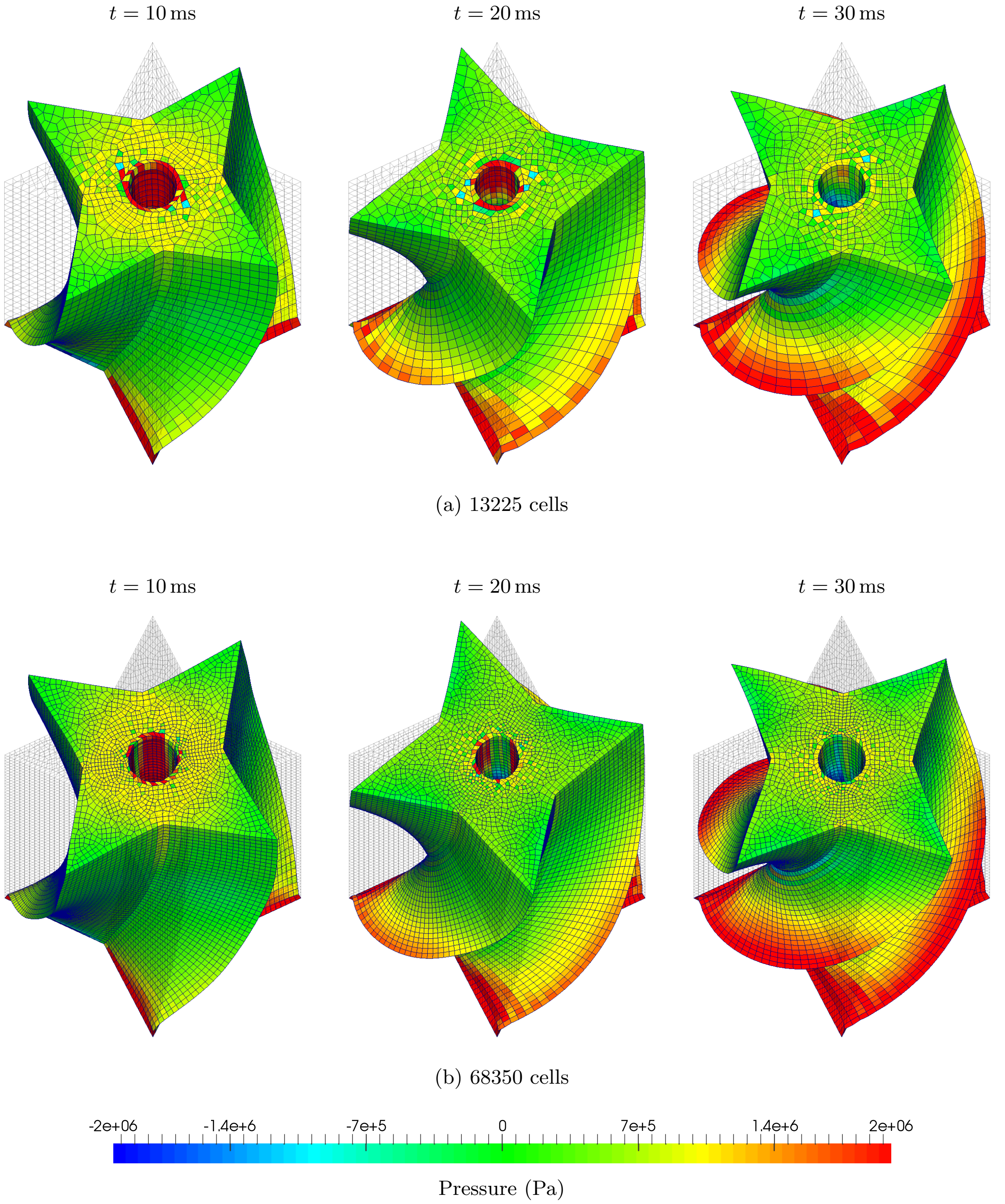
FIGURE 9.14: Complex twisting: Time evolution of the deformation along with the pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with a discretisation of 13225 hexahedral elements. Results obtained with an angular velocity $\boldsymbol{\omega}_0 = \mathit{\Omega}\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\mathit{\Omega} = 105$ rad/s and $H = 2$ m. A neo-Hookean constitutive model is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$, $\alpha_{\text{CFL}} = 0.3$ and $\mathit{\Delta} t \approx 1 \times 10^{-5}$ s.



(a) $t = 10\,\text{ms}$

(b) $t = 30\,\text{ms}$

FIGURE 9.15: Complex twisting: Comparison of deformed shapes along with pressure distribution at times (a) $t = 10$ ms; and (b) $t = 30$ ms using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ cell centred finite volume methodologies C-TOUCH, P-TOUCH ($\xi_{\{\boldsymbol{F},\boldsymbol{H},J\}} = 0.1$) and X-GLACE schemes. Results obtained with discretisation of 13225 hexahedral elements along with an angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 105$ rad/s and $H = 2$ m. A neo-Hookean constitutive model is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.



(a) 13225 cells

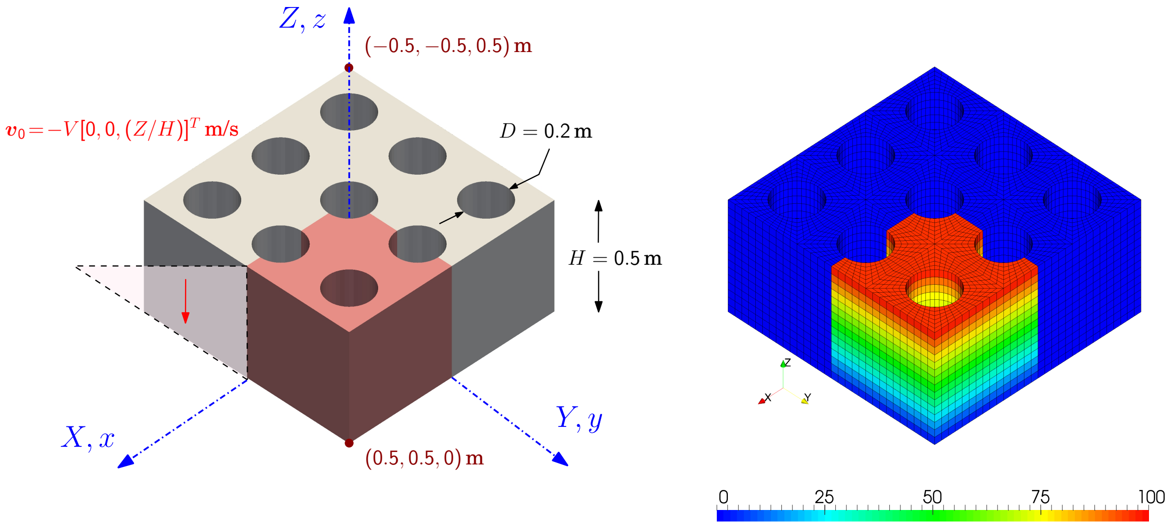
(b) 68350 cells

Figure 9.16: Complex twisting: Mesh refinement of deformation along with the pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with $\tilde{\kappa} = 3\kappa$. Results obtained using a discretisation of (a) 13225; and (b) 68350 hexahedral elements along with an angular velocity $\boldsymbol{\omega}_0 = \Omega\,[0, \sin(\pi Y/2H), 0]^T$ rad/s where $\Omega = 105$ rad/s and $H = 2$ m. A neo-Hookean material is used with $\rho_0 = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\text{CFL}} = 0.3$

9.2.2 Punch cube

A more challenging example is presented in this section where a $1 \times 1\,\mathrm{m}^2$ cross section block of height $H = 0.5$ m is considered with nine equally spaced holes of diameter 0.2 m (see Fig. 9.17a). The problem is initialised with a linear velocity profile $\boldsymbol{v}_0 = -V[0, 0, (Z/H)]^T$ m/s in quarter of the domain ($X \geq 0$, $Y \geq 0$) (see Fig. 9.17b). A neo-Hookean constitutive model is used for the numerical simulation with material properties density $\rho = 1100$ kg/m^3, Youngs's Modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.45$ (unless otherwise stated).

Fig. 9.18 shows the time evolution of the deformation pattern of block when subjected to the initial velocity. A very smooth pressure distribution can be observed despite employing a complicated geometry. In Fig. 9.19, a similar evolution pattern is shown from bottom plane ($Z = 0$) at different time instants to highlight the extreme deformation experienced in this region. The problem is then analysed in the near incompressibility regime by increasing the Poisson's ratio to $\nu = 0.499$. The subsequent deformation of the block is portrayed in Figs. 9.20 and 9.21. Once again, the results show that no pressure oscillations and locking are encountered. Moreover, a mesh refinement study has also been carried out in Fig. 9.22 using 32400 and 86400 structured hexahedral elements. In this figure, right half of the domain has been clipped to show interior pressure distribution whilst the wireframe displays undeformed mesh. Remarkably, it is clear that despite increasing the number of elements from 32400 to 86400, both the deformation and pressure resolution obtained are practically identical. More importantly, Fig. 9.23 highlights the significance of the preconditioned Riemann solver. When using the value of $\tilde{\kappa} = \kappa$, spurious pressure modes are accummulated over time which would eventually lead to breakdown of the numerical scheme (see Fig. 9.23a). This shortcoming can be eliminated by resorting to preconditioning with $\tilde{\kappa} = 3\kappa$ in order to recover a correct scaling for numerical stabilisation.



(a) Initial configuration

(b) Initial velocity profile (m/s) and mesh

FIGURE 9.17: Punch test: Problem setup.

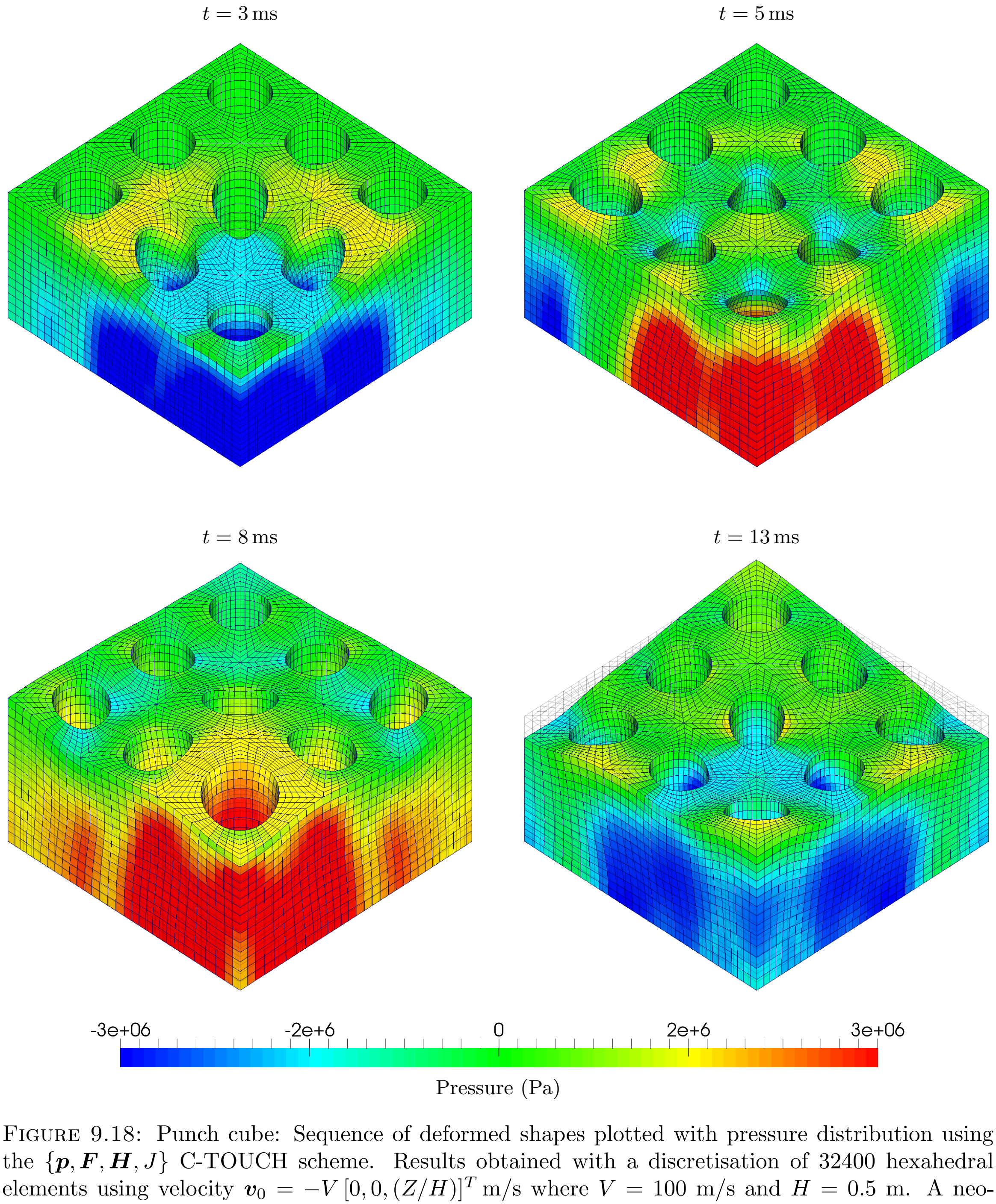


FIGURE 9.18: Punch cube: Sequence of deformed shapes plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. Results obtained with a discretisation of 32400 hexahedral elements using velocity $\boldsymbol{v}_0 = -V\,[0, 0, (Z/H)]^T$ m/s where $V = 100$ m/s and $H = 0.5$ m. A neo-Hookean constitutive model is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$, $\alpha_{\mathrm{CFL}} = 0.3$ and $\Delta t \approx 1 \times 10^{-5}$ s.

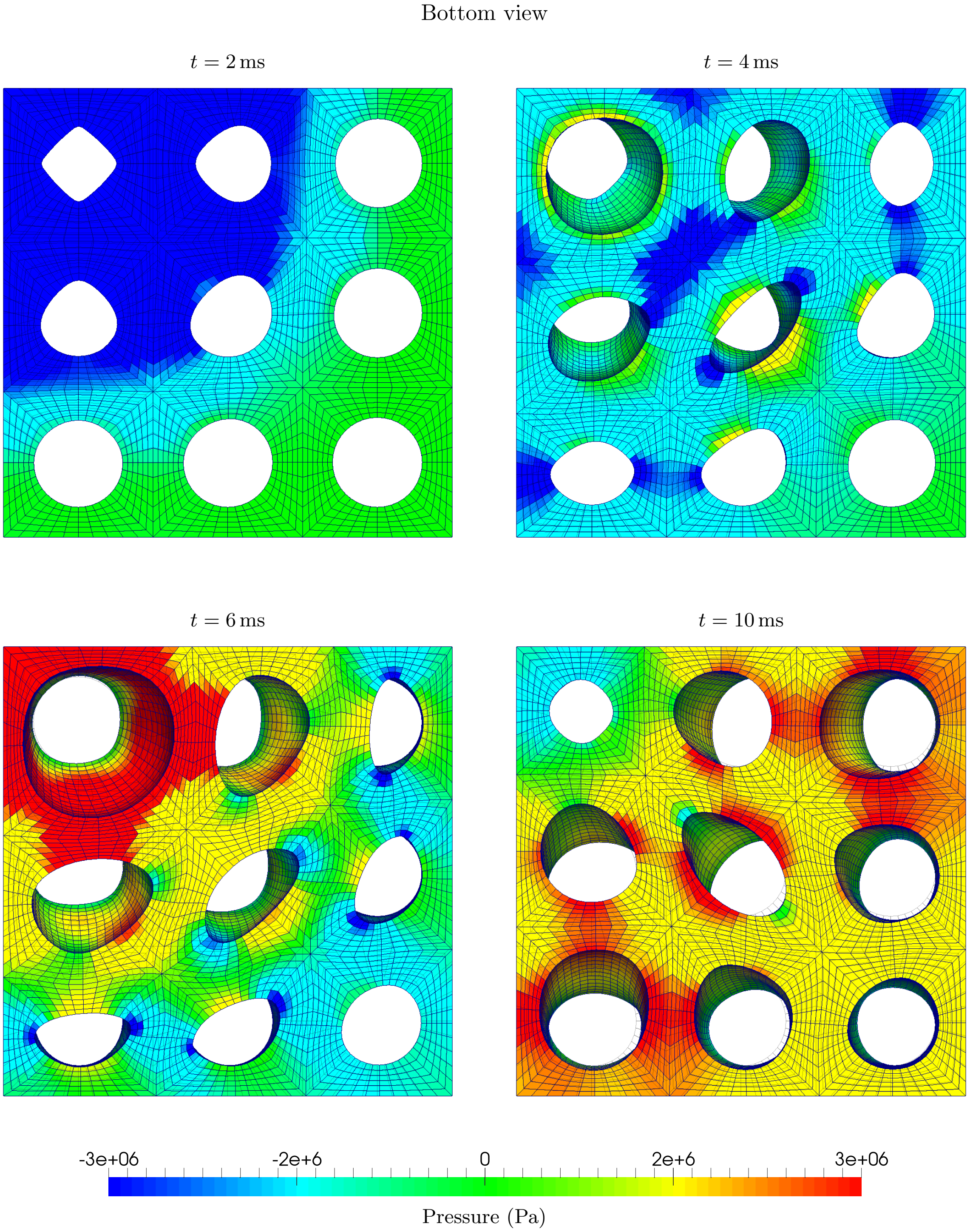


FIGURE 9.19: Punch cube: Sequence of deformed shapes plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. Results obtained with a discretisation of 32400 hexahedral elements using velocity $\boldsymbol{v}_0 = -V\,[0, 0, (Z/H)]^T$ m/s where $V = 100$ m/s and $H = 0.5$ m. A neo-Hookean constitutive model is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\mathrm{CFL}} = 0.3$.

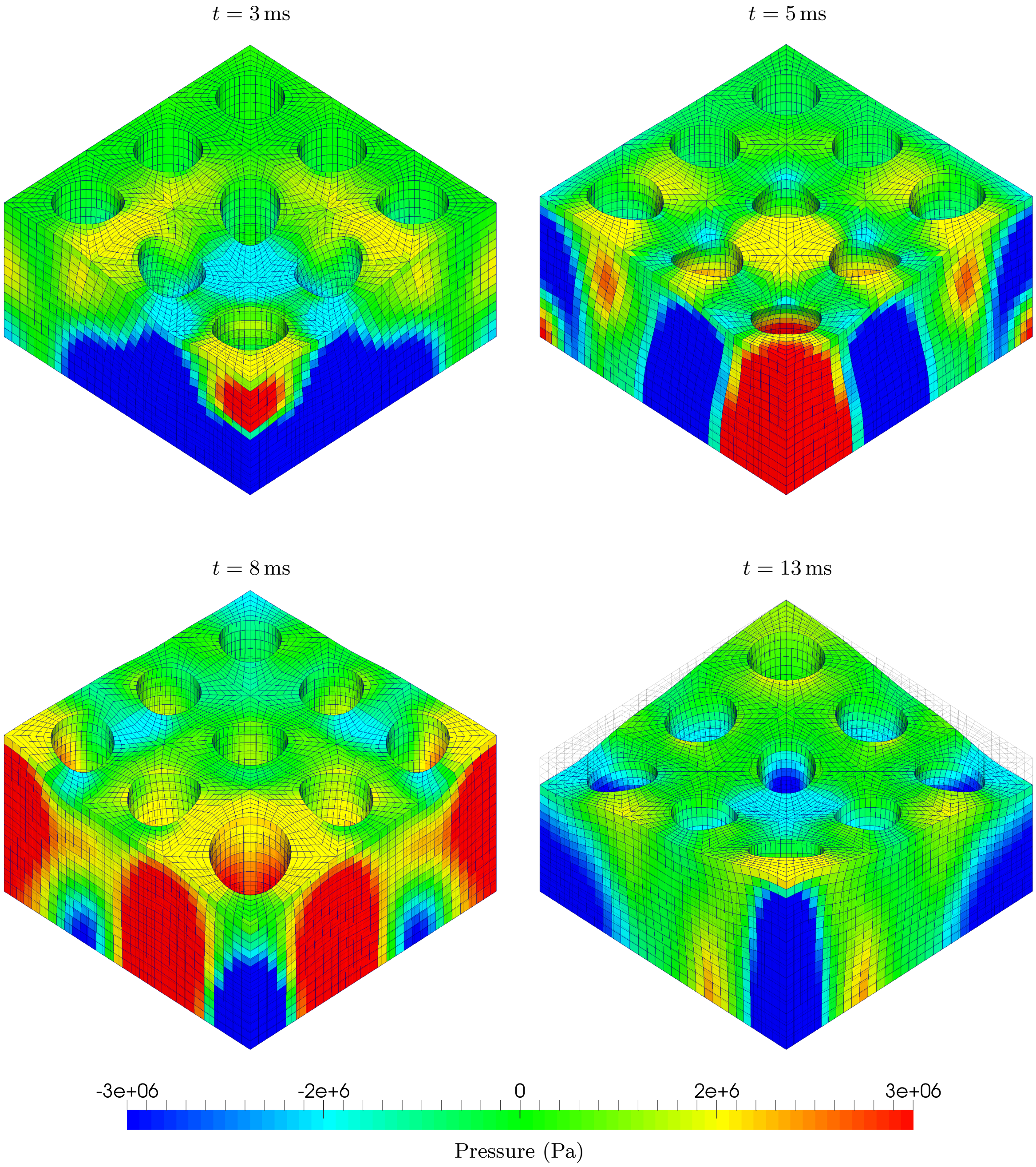


FIGURE 9.20: Punch cube: Sequence of deformed shapes plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with $\tilde{\kappa} = 3\kappa$. Results obtained with a discretisation of 32400 hexahedral elements using velocity $\boldsymbol{v}_0 = -V\,[0, 0, (Z/H)]^T$ m/s where $V = 100$ m/s and $H = 0.5$ m. A neo-Hookean constitutive model is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\text{CFL}} = 0.3$.

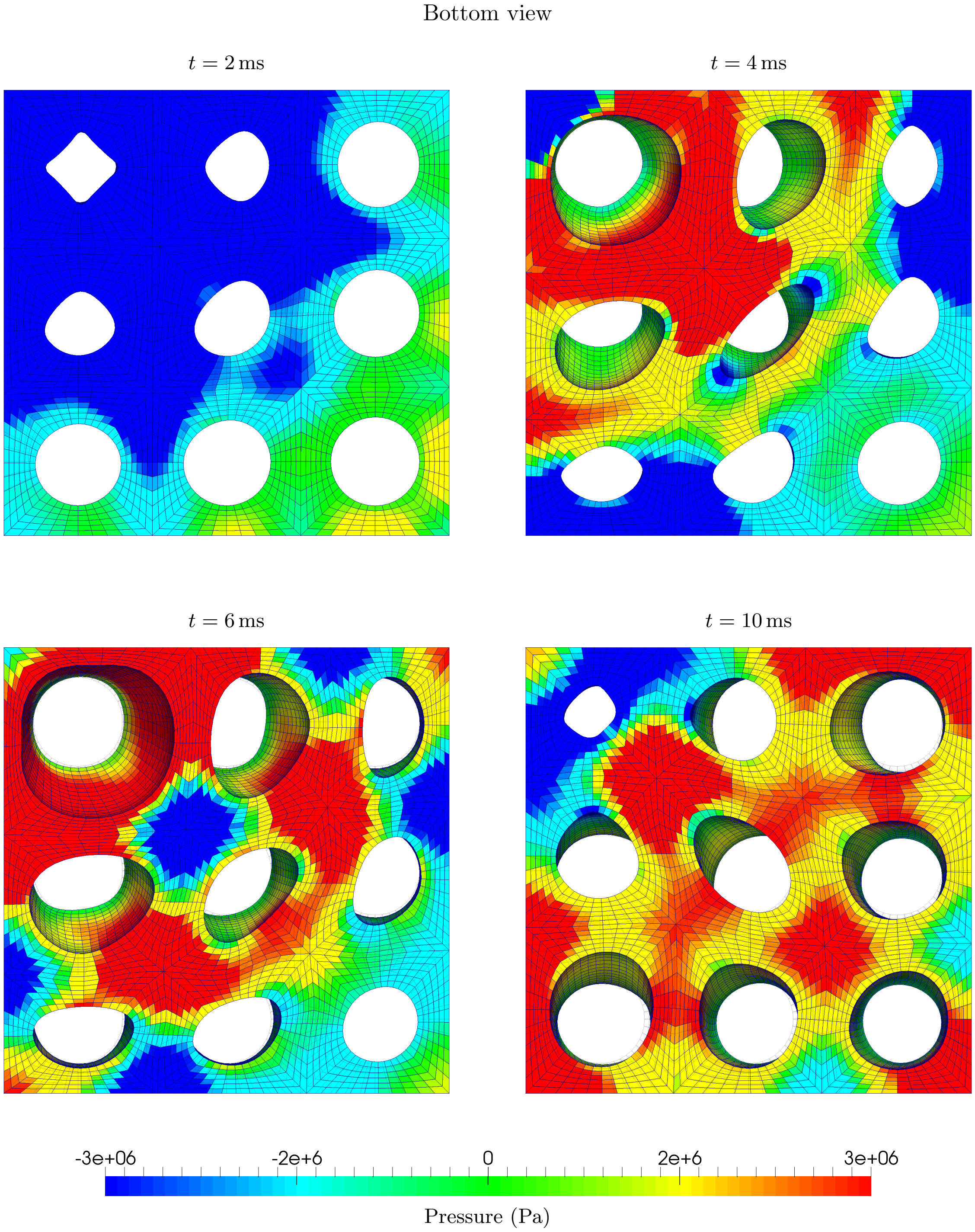
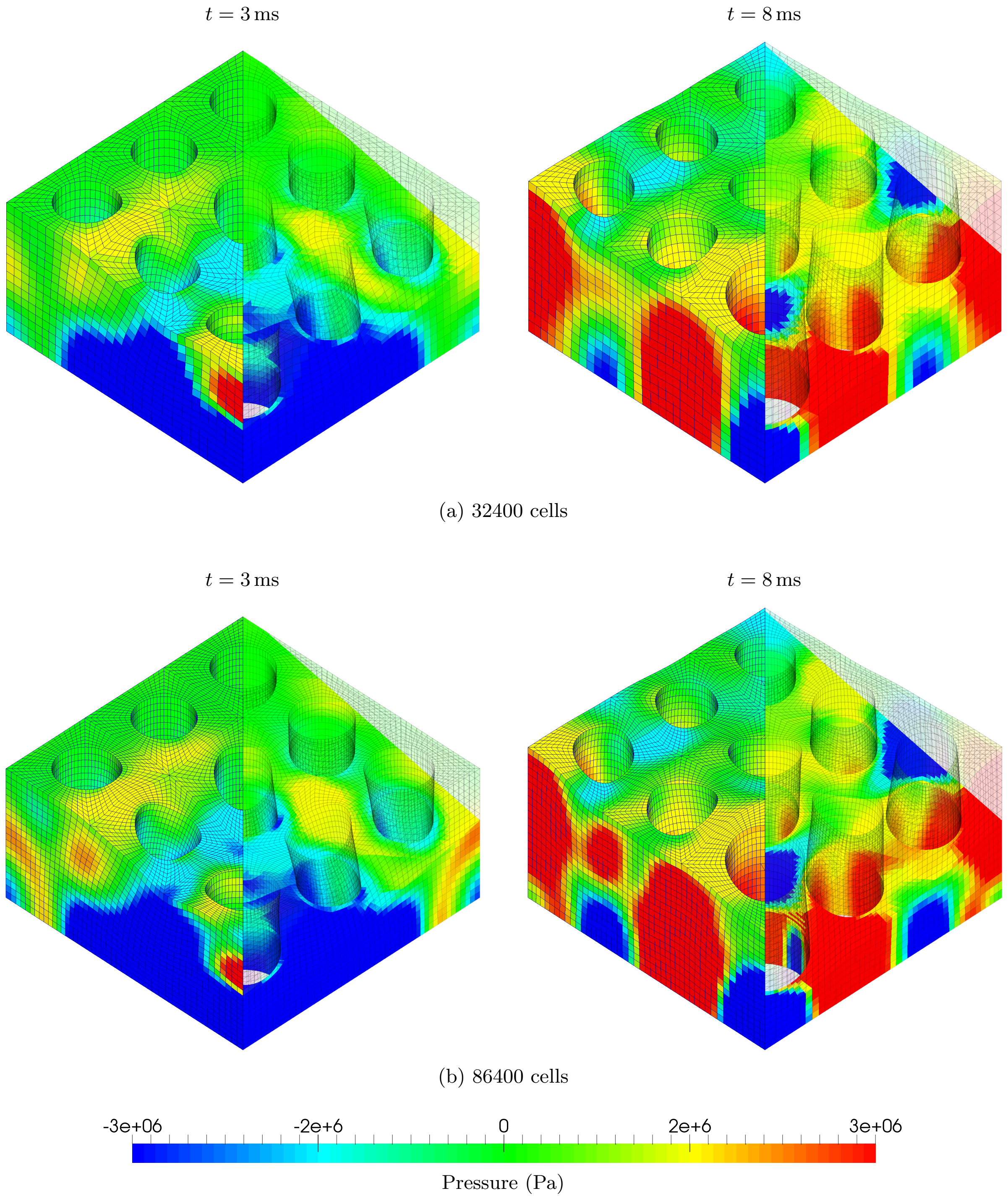


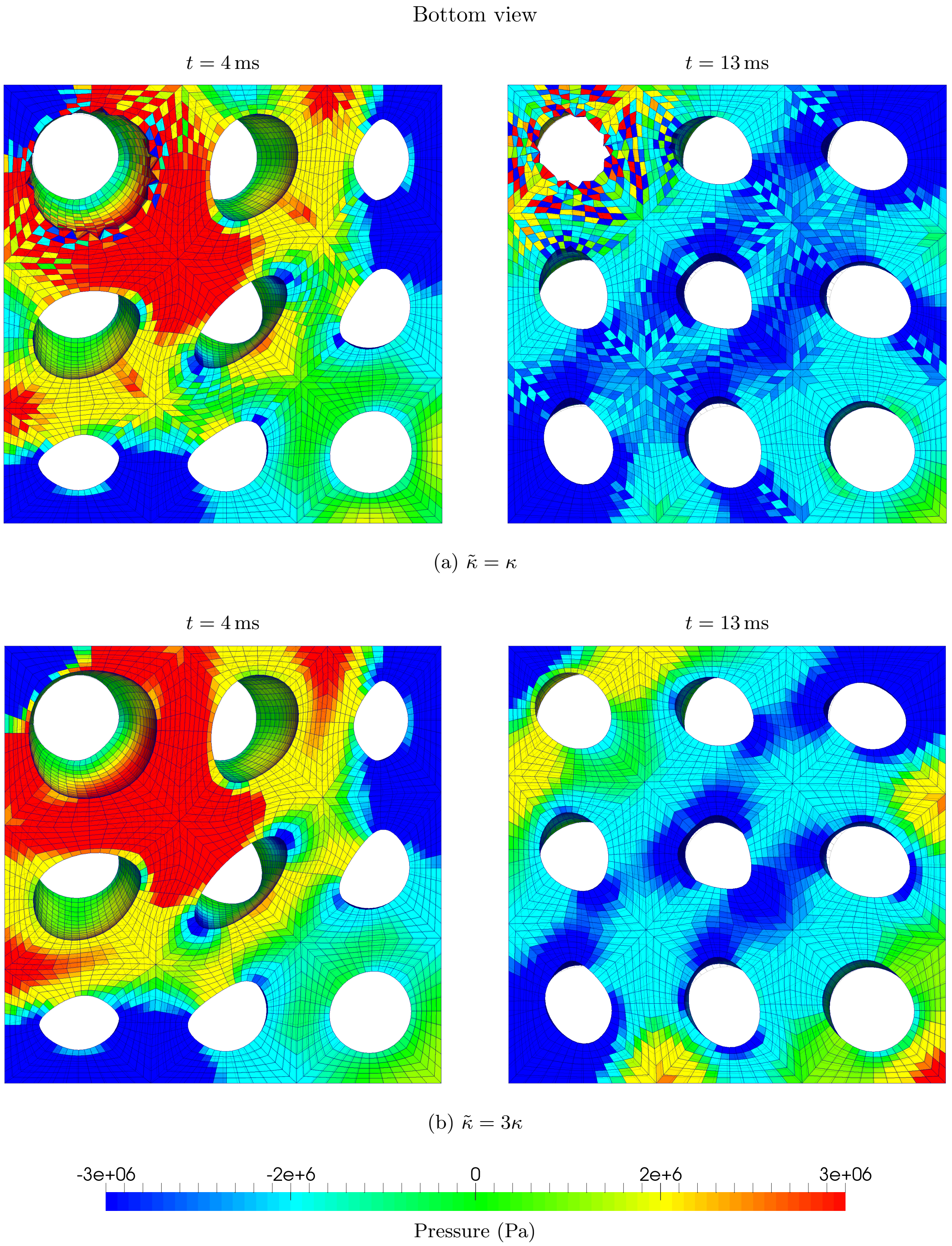
FIGURE 9.21: Punch cube: Sequence of deformed shapes plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with $\tilde{\kappa} = 3\kappa$. Results obtained with a discretisation of 32400 hexahedral elements using velocity $\boldsymbol{v}_0 = -V\,[0, 0, (Z/H)]^T$ m/s where $V = 100$ m/s and $H = 0.5$ m. A neo-Hookean constitutive model is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\mathrm{CFL}} = 0.3$.



(a) 32400 cells

(b) 86400 cells

FIGURE 9.22: Punch cube: Mesh refinement of deformed shapes along with pressure distribution at times $t = 3$ ms and $t = 8$ ms using mesh sizes of (a) 32400; and (b) 86400 hexahedral elements. Results obtained using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme using $\tilde{\kappa} = 3\kappa$ with velocity $\boldsymbol{v}_0 = -V\,[0, 0, (Z/H)]^T$ m/s where $V = 100$ m/s and $H = 0.5$ m. A neo-Hookean constitutive model is utilised with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\mathrm{CFL}} = 0.3$.



Bottom view

$t = 4\,\mathrm{ms}$

$t = 13\,\mathrm{ms}$

(a) $\tilde{\kappa} = \kappa$

$t = 4\,\mathrm{ms}$

$t = 13\,\mathrm{ms}$

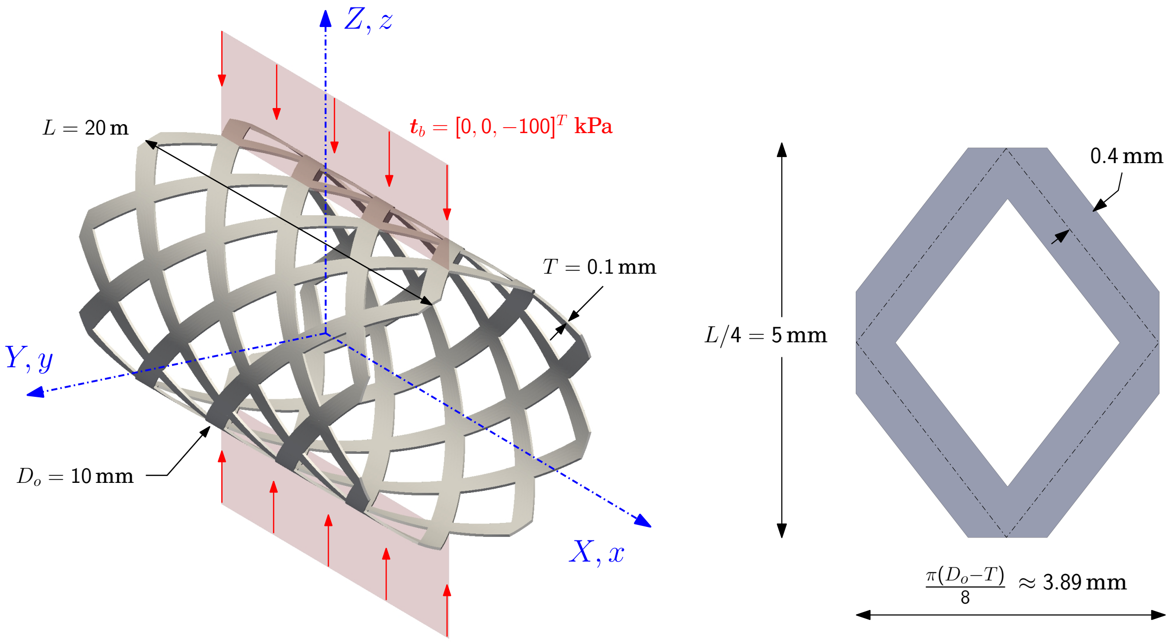
(b) $\tilde{\kappa} = 3\kappa$

Figure 9.23: Punch cube: Comparison of deformed shapes plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme with (a) $\tilde{\kappa} = \kappa$; and (b) $\tilde{\kappa} = 3\kappa$. Results obtained with a discretisation of 32400 hexahedral elements using velocity $\boldsymbol{v}_0 = -V\,[0, 0, (Z/H)]^T$ m/s where $V = 100$ m/s and $H = 0.5$ m. A neo-Hookean constitutive model is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\mathrm{CFL}} = 0.3$.

9.2.3 Stent-like structure

The objective of this example is to demonstrate robustness of the proposed C-TOUCH scheme on complicated, real-life problems. The geometry presented in Fig. 9.24a is very similar to a cardiovascular stent widely used in biomedical applications. This stent-like structure has an initial outer diameter $D_o = 10$ mm, a thickness $T = 0.1$ mm and a total length $L = 20$ mm. The dimensions of one of the repeated symmetric patterns, when folded out on a planar surface are shown in Fig. 9.24b. In this problem, we simulate the crushing of this stent-like structure by applying a constant traction $\boldsymbol{t}_b = [0, 0, -100]^T$ kPa at the top and bottom of the stent along the X-Z plane. Due to the presence of three symmetry planes, $1/8^{\text{th}}$ of the problem is simulated with appropriate symmetric boundary conditions, where the rest of the geometry has zero traction (free) boundary condition. The stent-like structure is made of a polyconvex material with density $\rho = 1100$ kg/m^3, Youngs's Modulus $E = 17$ MPa and Poisson's ratio $\nu = 0.45$ (unless otherwise stated).

Fig. 9.25 shows deformation of the structure along with smooth pressure distribution throughout the computational domain. It is remarkable that employing only 2 elements along the thickness of structure and a fairly coarse mesh of 6912 elements, the results show no pressure instabilities. Fig. 9.26 the deformation at time $t = 500\,\mu$s with zoomed views in areas of sharp spatial gradients. Fig. 9.27 shows the capability of proposed $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH algorithm to handle a complex geometry in conjunction with a highly incompressible material ($\nu = 0.499$).


Z, z
$L = 20$ m
$\boldsymbol{t}_b = [0, 0, -100]^T$ kPa
$T = 0.1$ mm
Y, y
$D_o = 10$ mm
X, x
0.4 mm
$L/4 = 5$ mm
$\frac{\pi(D_o - T)}{8} \approx 3.89$ mm

(a) Initial configuration

(b) Planar $1/32^{\text{th}}$ geometry

FIGURE 9.24: Stent-like structure: Problem setup.

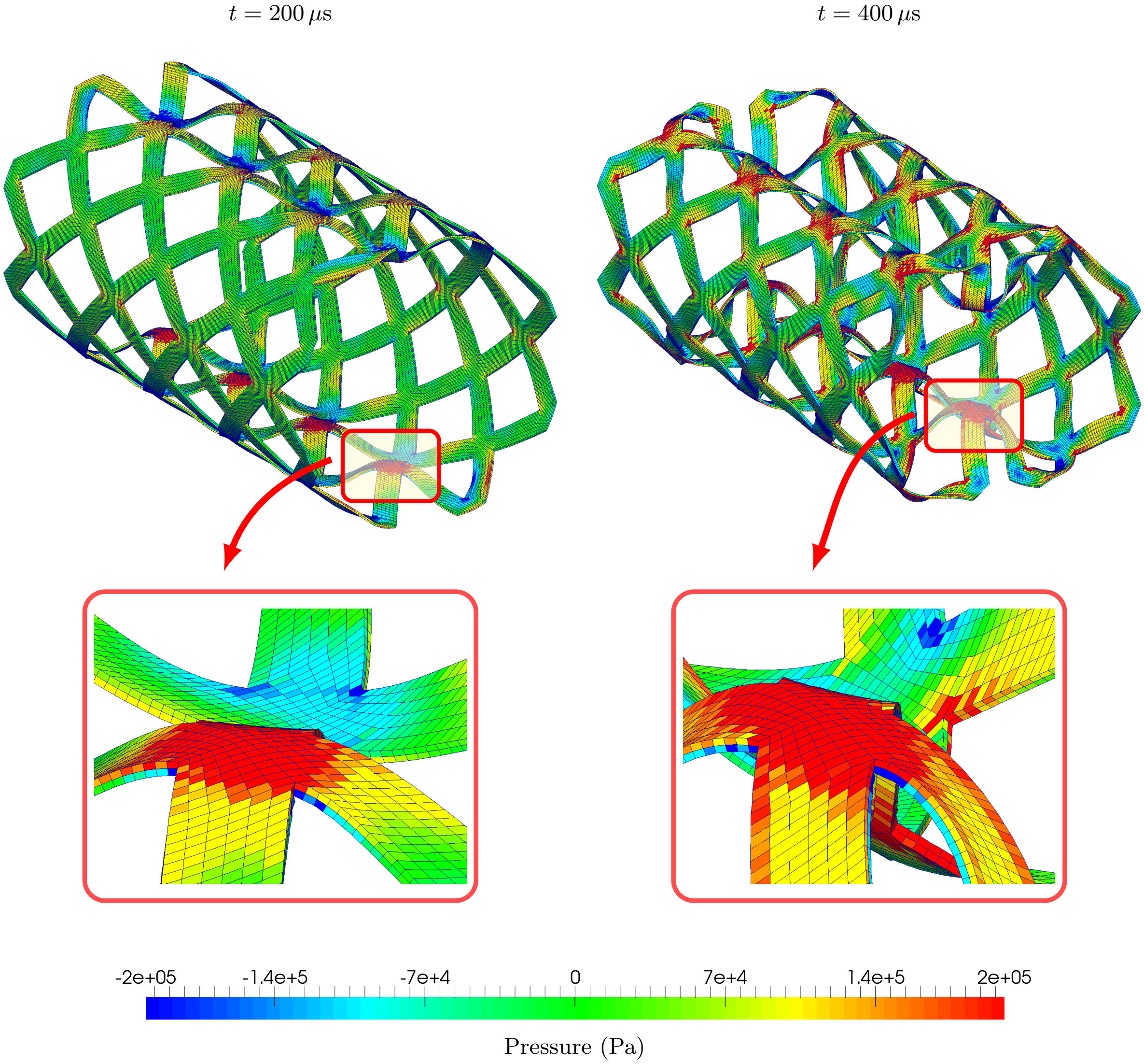


FIGURE 9.25: Stent-like structure: Sequence of deformed shapes at time $t = 200\,\mu$s and $t = 400\,\mu$s plotted with pressure distribution using the $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme. Results obtained with a discretisation of 6912 hexahedral elements using traction loading $\boldsymbol{t}_b = [0, 0, -100]^T$ kPa. A neo-Hookean material is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$, $\alpha_{\text{CFL}} = 0.3$ and $\Delta t \approx 5\times10^{-8}$ s.

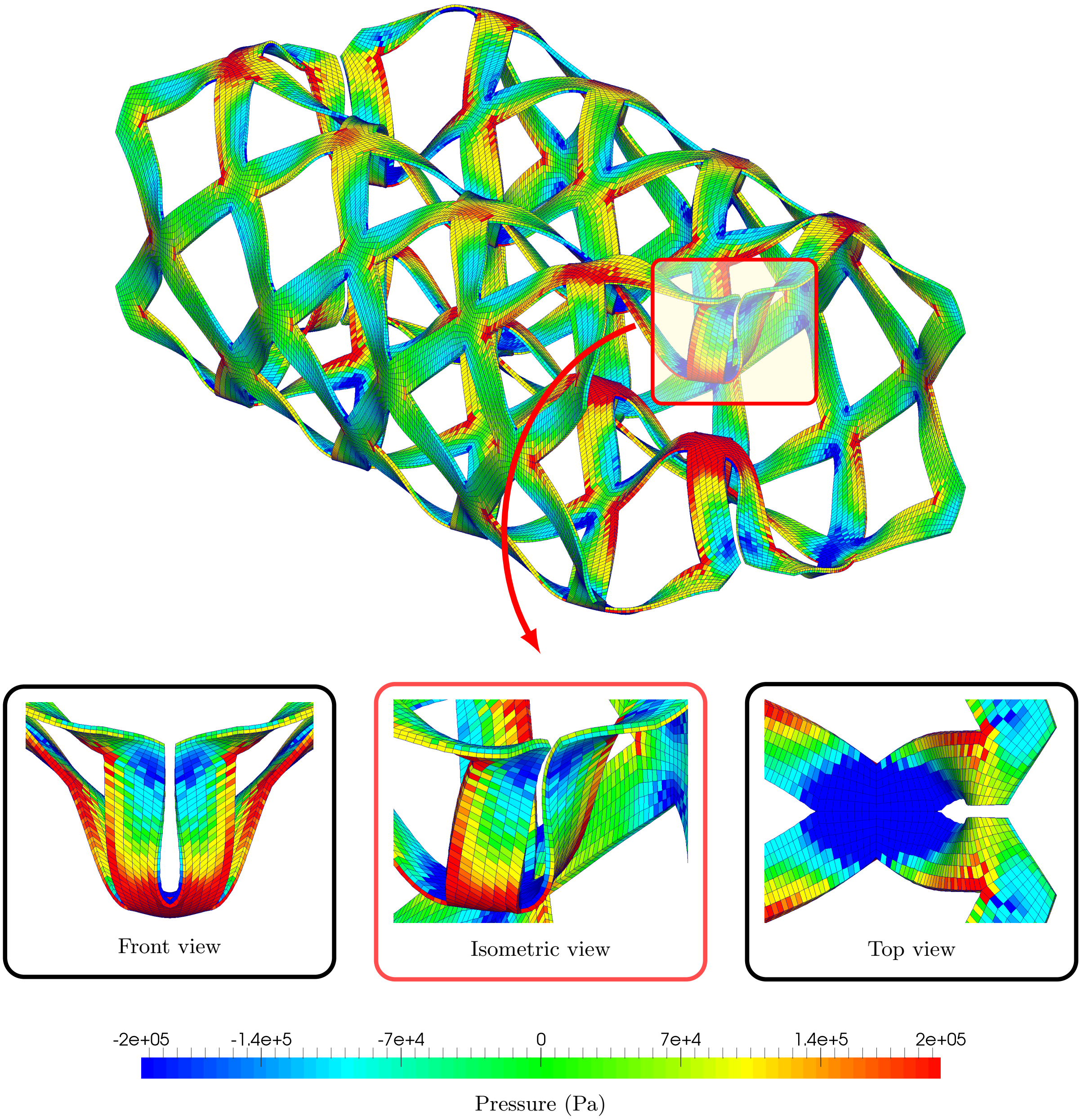


FIGURE 9.26: Stent-like structure: Snapshot of deformed shape highlighting the pressure distribution in key region using $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme at time $t = 500\,\mu$s. Results obtained with a discretisation of 6912 hexahedral elements using traction loading $\boldsymbol{t}_b = [0, 0, -100]^T$ kPa. A neo-Hookean material is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.45$ and $\alpha_{\text{CFL}} = 0.3$.

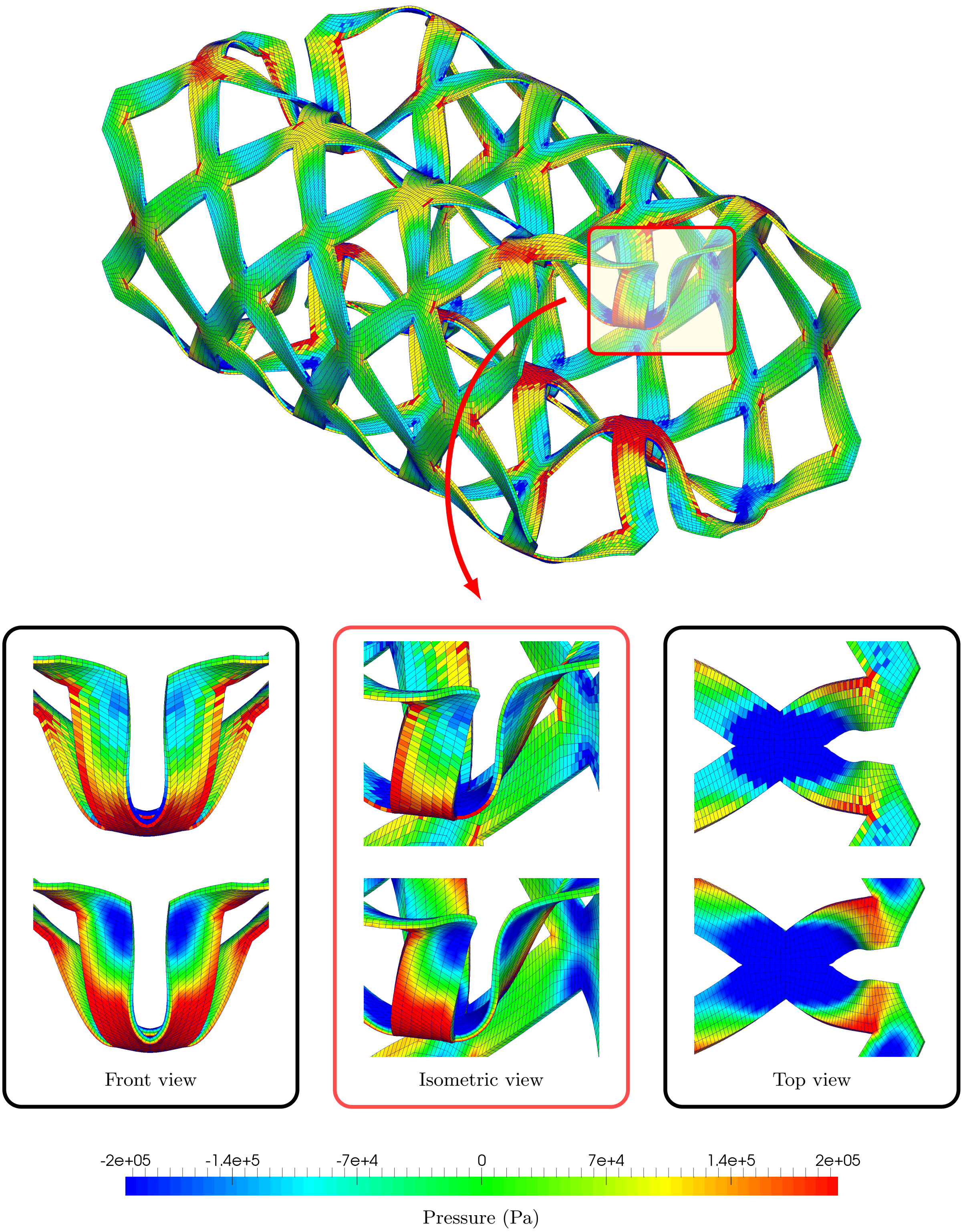


Figure 9.27: Stent-like structure: Snapshot of deformed shape highlighting the pressure distribution in the key region using $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J\}$ C-TOUCH scheme ($\tilde{\kappa} = 3\kappa$) at time $t = 500\,\mu$s. The top row displays the cell center pressure whereas the bottom row shows the interpolated/extrapolated pressure at the nodes. Results obtained with a discretisation of 6912 hexahedral elements using traction loading $\boldsymbol{t}_b = [0, 0, -100]^T$ kPa. A neo-Hookean material is used with $\rho = 1100$ kg/m^3, $E = 17$ MPa, $\nu = 0.499$ and $\alpha_{\text{CFL}} = 0.3$.

Chapter 10

CONCLUDING REMARKS

10.1 Summary

OpenFOAM [130] is a free and open source CFD software package extensively used in both academic and industrial environments. Based on the cell centred Finite Volume Method, OpenFOAM provides some basic solid solvers in addition to the extensive list of robust fluid solvers. The built-in solid solvers rely on the traditional displacement based formulation, only capable of solving solid behaviour within the small strain linear elastic regime [126]. The main objective of this thesis is to improve the capability of solid solvers in OpenFOAM so that they can mimic large strain behaviour, particularly in the case of nearly incompressible materials. This is achieved by implementing from scratch, a new solid solver entitled "`mixedSolidFoam`" within the OpenFOAM environment. The solver is implemented in a three dimensional setting via a mixed-based computational framework, aiming to bridge the gap between Computational Fluid Dynamics and Computational Solid Dynamics.

Following the work of Lee *et. al.* [6], the mixed-based methodology is formulated in the form of a system of first order hyperbolic conservation laws [6, 8, 10, 11, 13, 16–19] presented in a Total Lagrangian setting. The primary unknown variables are the linear momentum $\boldsymbol{p}$ and deformation gradient tensor $\boldsymbol{F}$ of the system. Moreover, an extended set of formulation is also presented, where two additional geometric conservation laws (one for the area map $\boldsymbol{H}$ and the other for volume map J) and another conservation law for the total energy of the system E is introduced. This proposed cell centred finite volume methodology is entitled **TOUCH**, an abbreviation of **TO**tal Lagrangian **U**pwind **C**ell centred Finite Volume Method for **H**yperbolic conservation laws. For closure of the system, several isothermal constitutive models are presented, namely polyconvex Mooney-Rivlin and isotropic von-Mises plasticity models. For completeness, a thorough eigenvalue structure analysis of the full system is carried out guaranteeing the existence of physical (real) wave speeds (thus material stability [16]) and the satisfaction of rank one convexity condition.

From the spatial discretisation point of view, a second order monotonicity-preserving cell centred Finite Volume Method is presented which utilises a linear reconstruction procedure along with a Barth and Jespersen slope limiter. This discretisation is presented in terms of the standard Godunov-type cell centred FVM where the fluxes are located at the face, as well as the nodal cell centred FVM where the fluxes are located at the nodes. In order to evaluate these numerical (contact) fluxes, an acoustic Riemann solver is presented. The nodal Riemann solver in entitled X-GLACE, since it is an extension of the hyperelastic GLACE scheme proposed in [5]. Moreover, the Godunov-type Riemann solver is further extended so that it is capable of simulating contact scenarios. More crucially, in order to extend the application range towards incompressibility limit, the proposed schemes are further enhanced through a preconditioned Riemann solver [113, 146]. It has been clearly shown that the preconditioned flux evaluation, obtained with a correct scaling of the numerical stabilisation, effectively alleviates the appearance of spurious modes when solving nearly incompressible solids ($\kappa/\mu > 500$).

Unfortunately, the standard Godunov-type finite volume update for the deformation gradient $\boldsymbol{F}$ and its cofactor $\boldsymbol{H}$ do not necessarily ensure the satisfaction of involutions (also known as compatibility conditions) [6, 107, 117]. Thus, $\boldsymbol{F}$ is not curl-free and $\boldsymbol{H}$ is not divergence-free over a long term response leading to the appearance of spurious mechanisms which eventually cause breakdown of the numerical scheme. To overcome this, two alternative evolutionary frameworks [13, 117] in ensuring $\text{CURL}\boldsymbol{F} = \boldsymbol{0}$ and $\text{DIV}\boldsymbol{H} = \boldsymbol{0}$ are introduced namely, (a) Constrained-TOUCH; and (b) Penalised-TOUCH schemes. It is important to notice that the nodal finite volume framework satisfies these involutions by construction and therefore no ad-hoc procedure is required.

The second order semi-discretisation in space is supplemented with an equal order temporal discretisation utilising an explicit one-step two-stage Total Variation Diminishing Runge-Kutta time integration scheme. Exactly the same time integrator is used in updating the geometry. Moreover, the use of a global posteriori angular momentum projection procedure within the space-time integrator enables the preservation of angular momenta for all the proposed schemes.

Finally, a series of benchmark numerical examples are simulated to access the convergence characteristics, momentum-preservation properties, numerical dissipation and locking-free nature of the proposed $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J, E\}$ cell centred finite volume methodologies. For comparison purposes, the numerical results are also benchmarked against the popular B-bar method and an ample spectrum of in-house mixed methodologies. The methodologies show excellent behaviour in bending dominated nearly incompressible scenarios ($\kappa/\mu \geq 500$). Moreover, in order to check the robustness of the proposed numerical schemes, more challenging examples are simulated with special emphasis on contact problems and complex geometry.

Table 10.1 summarises the novelties introduced as part of this work in the context of cell centred finite volume framework.

	Novelties of this thesis
Chapter 2	• An extended set of $\{\boldsymbol{p}, \boldsymbol{F}, \boldsymbol{H}, J, E\}$ first order hyperbolic conservation laws for solid dynamics. • Advanced polyconvex constitutive model.
Chapter 3	• Second order spatial discretisation using nodal cell centred finite volume framework. • A generalised description for gradient evaluation including enhanced gradient calculation incorporating additional boundary points.
Chapter 4	• A comprehensive description of boundary conditions. • Acoustic Riemann solver presented for the nodal cell centred FVM in a Total Lagrangian framework. • Extension of the formulation to contact scenarios. • Introduction of preconditioned dissipation in the Riemann solver to deal with highly incompressible scenarios.
Chapter 5	• Satisfaction of underlying involutions of the system through C-TOUCH and P-TOUCH schemes.
Chapter 6	• Monolithic time integration procedure where the geometry is also updated through the TVD Runge-Kutta scheme. • A new global posteriori projection procedure which ensures preservation of angular momentum.
Chapter 7	• Implementation of a new solid solver `mixedSolidFoam` from scratch in the open source CFD software package OpenFOAM.
Chapters 8 and 9	• Numerical simulations of various benchmark problems in a three-dimensional setting. • Ability to simulate materials in the near incompressibility regime $\kappa/\mu \approx 500$. • Robustness of the proposed cell centred schemes for complex geometries.

Table 10.1: A summary of novelties presented in this thesis within the cell centred FVM framework.

10.2 Future work

The work presented in this thesis opens up other possible lines of research which can be explored in future. Some of these are

□ ***Parallelisation of computational framework in OpenFOAM:***

One of the biggest advantages of OpenFOAM lies in the fact that its existing solvers allow parallel computations which is ideal to simulate real-life problems. OpenFOAM relies on domain decomposition for parallel processing, implying that the geometry and its associated fields are divided and allocated to different processors for solution [126]. The solver is then run separately on individual processors which must communicate efficiently near the processor-processor boundaries for correct computation of quantities. OpenFOAM uses the openMPI implementation of the standard Message Passing Interface (MPI) to facilitate communication between the parallel processes as the solution is computed [126]. Current work involves additional computations such as the evaluation of nodal quantities which is generally not common in standard finite volume codes. This is the reason why, at present, the computational framework presented in this thesis is not parallelised. Once the platform is parallelised, the code will be released to the OpenFOAM solid mechanics community so that it can be tested in various applications. A journal article on the implementation of proposed cell centred Finite Volume Method is under preparation and planned to be submitted to Computer Physics Communications journal.

□ ***Connection between HDG, Constrained-TOUCH and X-GLACE frameworks:***

It is also possible to show that the Hybridizable Discontinuous Galerkin (HDG) , Godunov-type cell centred FVM and the nodal cell centred FVM schemes can be linked together. This will probably help to better understand the problem in a general manner. It may also give a breakthrough in utilising tetrahedral meshes for the cell centred schemes which is very useful for meshing complicated real-life geometries. A journal article is under preparation and planned to be submitted to the Journal of Computational Physics.

□ ***Roe-type Riemann solver for solids:***

An acoustic Riemann solver has been utilised in this study for the evaluation of numerical contact fluxes. More advanced Riemann solvers such as a Roe-type Riemann solver in the case of solids could be employed. This would allow better shock capturing capabilities useful to simulate more complex physics, specially in the case of contact problems. Another journal article is under preparation on this topic and planned to be submitted to the Journal of Computational Physics.

□ ***Applicability to quasi-static problems:***

The presented framework also allows for the simulation of quasi-static problems by introducing Raleigh damping or artificial viscosity into the system.

- □ ***Inclusion of a thermo-mechanical constitutive model:***

  Since only isothermal constitutive models are employed in this thesis, energy equation E is decoupled from rest of the system. In order to extend the range of applications, temperature dependent constitutive models could be included.

- □ ***ALE description for multi material modelling:***

  An Alternative Arbitrary Lagrangian-Eulerian (ALE) description [138, 141, 147] of motion could be utilised for multi material modelling. Moreover, it would also help to overcome the drawback of mesh entanglement which could be experienced in the Lagrangian framework .

Appendix A

MATHEMATICAL OPERATORS

A.1 Gradient, divergence and curl operators

In this section, a nomenclature is used such that a scalar quantity is denoted by a, vector quantity by $\boldsymbol{a}$ and second order tensor quantity by $\boldsymbol{A}$. Using this nomenclature, the gradient operator is defined as

$$[\nabla a]_i = \frac{\partial a}{\partial x_i}; \qquad [\nabla \boldsymbol{a}]_{ij} = \frac{\partial a_i}{\partial x_j}; \qquad [\nabla \boldsymbol{A}]_{ijk} = \frac{\partial A_{ij}}{\partial x_k}. \tag{A.1}$$

The divergence operator for a vector and second order tensor field is specified as

$$\nabla \cdot \boldsymbol{a} = \frac{\partial a_i}{\partial x_i}; \qquad [\nabla \cdot \boldsymbol{A}]_i = \frac{\partial A_{ij}}{\partial x_j}, \tag{A.2}$$

where repeated indices denote Einstein notation. Furthermore, the curl of a second order tensor is defined as

$$[\nabla \times \boldsymbol{A}]_{iI} = \epsilon_{IJK} \frac{\partial A_{iK}}{\partial X_J}, \tag{A.3}$$

where ϵ is the alternating tensor or Permutation/Levi-Civita symbol such that

$$\begin{aligned} \epsilon_{IJK} &= 1 && \text{if } IJK = 123, 231, 312 \text{ (cyclic order)} \\ \epsilon_{IJK} &= 0 && \text{if any two indices are equal} \\ \epsilon_{IJK} &= -1 && \text{if } IJK = 321, 213, 132 \text{ (anti cyclic order)} \end{aligned} \tag{A.4}$$

Notice that, throughout this thesis, the operators ∇ (or grad), $\nabla\cdot$ (or div) and $\nabla\times$ (or CURL) have been used to denote spatial gradient, spatial divergence and spatial curl operators respectively. Similarly, ∇_0 (or GRAD), $\nabla_0\cdot$ (or DIV) and $\nabla_0\times$ (or CURL) represent material gradient, material divergence and material curl operators respectively.

A.2 Tensor cross product

The following list of properties are given in reference [16, 94]. Note that a is a scalar, $\boldsymbol{V}$ and $\boldsymbol{W}$ denote material vectors, $\boldsymbol{v}$ and $\boldsymbol{w}$ denote spatial vectors, $\boldsymbol{I}$ represents identity tensor with Kronecker delta components $[\boldsymbol{I}]_{iI} = \delta_{iI}$ and $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are second order two-point tensors.

$$\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B} = \boldsymbol{B} \boldsymbol{\times} \boldsymbol{A} \tag{A.5}$$

$$\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B} = \boldsymbol{A}^T \boldsymbol{\times} \boldsymbol{B}^T \tag{A.6}$$

$$\boldsymbol{A} \boldsymbol{\times} (\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{A} \boldsymbol{\times} \boldsymbol{B} + \boldsymbol{A} \boldsymbol{\times} \boldsymbol{C} \tag{A.7}$$

$$a\,(\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}) = (a\boldsymbol{A}) \boldsymbol{\times} \boldsymbol{B} = \boldsymbol{A} \boldsymbol{\times} (a\boldsymbol{B}) \tag{A.8}$$

$$(\boldsymbol{v} \otimes \boldsymbol{V}) \boldsymbol{\times} (\boldsymbol{w} \otimes \boldsymbol{W}) = (\boldsymbol{v} \times \boldsymbol{w}) \otimes (\boldsymbol{V} \times \boldsymbol{W}) \tag{A.9}$$

$$\boldsymbol{v} \boldsymbol{\times} (\boldsymbol{A} \boldsymbol{\times} \boldsymbol{V}) = (\boldsymbol{v} \boldsymbol{\times} \boldsymbol{A}) \boldsymbol{\times} \boldsymbol{V} = \boldsymbol{v} \boldsymbol{\times} \boldsymbol{A} \boldsymbol{\times} \boldsymbol{V} \tag{A.10}$$

$$\boldsymbol{A} \boldsymbol{\times} (\boldsymbol{v} \otimes \boldsymbol{V}) = -\boldsymbol{v} \boldsymbol{\times} \boldsymbol{A} \boldsymbol{\times} \boldsymbol{V} \tag{A.11}$$

$$(\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}) : \boldsymbol{C} = (\boldsymbol{B} \boldsymbol{\times} \boldsymbol{C}) : \boldsymbol{A} + (\boldsymbol{A} \boldsymbol{\times} \boldsymbol{C}) : \boldsymbol{B} \tag{A.12}$$

$$(\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B})\,(\boldsymbol{V} \times \boldsymbol{W}) = (\boldsymbol{A}\boldsymbol{V}) \times (\boldsymbol{B}\boldsymbol{W}) + (\boldsymbol{B}\boldsymbol{V}) \times (\boldsymbol{A}\boldsymbol{W}) \tag{A.13}$$

$$\boldsymbol{A} \boldsymbol{\times} \boldsymbol{I} = (\mathrm{tr}\boldsymbol{A})\,\boldsymbol{I} - \boldsymbol{A}^T \tag{A.14}$$

$$\boldsymbol{I} \boldsymbol{\times} \boldsymbol{I} = 2\boldsymbol{I} \tag{A.15}$$

$$(\boldsymbol{A} \boldsymbol{\times} \boldsymbol{A}) : \boldsymbol{A} = 6\ \det\boldsymbol{A} \tag{A.16}$$

$$\mathrm{Cof}\boldsymbol{A} = \frac{1}{2}\boldsymbol{A} \boldsymbol{\times} \boldsymbol{A} \tag{A.17}$$

$$(\boldsymbol{A}\boldsymbol{C}) \boldsymbol{\times} (\boldsymbol{B}\boldsymbol{C}) = (\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B})\,(\mathrm{Cof}\boldsymbol{C}) \tag{A.18}$$

$$[\boldsymbol{v} \boldsymbol{\times} \boldsymbol{A}] = \begin{bmatrix} v_y A_{zX} - v_z A_{yX} & v_y A_{zY} - v_z A_{yY} & v_y A_{zZ} - v_z A_{yZ} \\ v_z A_{xX} - v_x A_{zX} & v_z A_{xY} - v_x A_{zY} & v_z A_{xZ} - v_x A_{zZ} \\ v_x A_{yX} - v_y A_{xX} & v_x A_{yY} - v_y A_{xY} & v_x A_{yZ} - v_y A_{xZ} \end{bmatrix}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{V}] = \begin{bmatrix} A_{xY} V_Z - A_{xZ} V_Y & A_{xZ} V_X - A_{xX} V_Z & A_{xX} V_Y - A_{xY} V_X \\ A_{yY} V_Z - A_{yZ} V_Y & A_{yZ} V_X - A_{yX} V_Z & A_{yX} V_Y - A_{yY} V_X \\ A_{zY} V_Z - A_{zZ} V_Y & A_{zZ} V_X - A_{zX} V_Z & A_{zX} V_Y - A_{zY} V_X \end{bmatrix}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}] = \begin{bmatrix} [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{xX} & [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{xY} & [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{xZ} \\ [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{yX} & [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{yY} & [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{yZ} \\ [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{zX} & [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{zY} & [\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{zZ} \end{bmatrix}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{xX} = A_{yY} B_{zZ} - A_{yZ} B_{zY} + A_{zZ} B_{yY} - A_{zY} B_{yZ}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{xY} = A_{yZ} B_{zX} - A_{yX} B_{zZ} + A_{zX} B_{yZ} - A_{zZ} B_{yX}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{xZ} = A_{yX} B_{zY} - A_{yY} B_{zX} + A_{zY} B_{yX} - A_{zX} B_{yY}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{yX} = A_{xZ} B_{zY} - A_{xY} B_{zZ} + A_{zY} B_{xZ} - A_{zZ} B_{xY}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{yY} = A_{zZ} B_{xX} - A_{zX} B_{xZ} + A_{xX} B_{zZ} - A_{xZ} B_{zX}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{yZ} = A_{zX} B_{xY} - A_{zY} B_{xX} + A_{xY} B_{zX} - A_{xX} B_{zY}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{zX} = A_{xY} B_{yZ} - A_{xZ} B_{yY} + A_{yZ} B_{xY} - A_{yY} B_{xZ}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{zY} = A_{xZ} B_{yX} - A_{xX} B_{yZ} + A_{yX} B_{xZ} - A_{yZ} B_{xX}$$

$$[\boldsymbol{A} \boldsymbol{\times} \boldsymbol{B}]_{zZ} = A_{xX} B_{yY} - A_{xY} B_{yX} + A_{yY} B_{xX} - A_{yX} B_{xY}$$

Appendix B

FINITE ELEMENTS

The Jacobian of transformation (also known as Jacobian matrix) for an arbitrary element (see Fig. B.2) can be expressed as follows [50]:

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} = \left[\frac{\partial \boldsymbol{x}}{\partial \zeta} \ \frac{\partial \boldsymbol{x}}{\partial \eta} \right] = \sum_{a \in \Lambda_e^a} \boldsymbol{x}_a \otimes \nabla_{\boldsymbol{\alpha}} N_a^e(\zeta, \eta), \tag{B.1}$$

where $\boldsymbol{x}$ are the parent coordinates, $\boldsymbol{\alpha}$ are the isoparametric coordinates, $\frac{\partial \boldsymbol{x}}{\partial \zeta}$ and $\frac{\partial \boldsymbol{x}}{\partial \eta}$ are the tangent vectors associated to the point under consideration in the isoparametric domain and $\nabla_{\boldsymbol{\alpha}} N_a^e(\zeta, \eta)$ is the gradient of the nodal shape function N_a^e with respect to the isoparametric coordinates. The normals and the area elements can be obtained as follows [50]:

$$\boldsymbol{n}(\zeta, \eta) = \frac{\frac{\partial \boldsymbol{x}}{\partial \zeta} \times \frac{\partial \boldsymbol{x}}{\partial \eta}}{\left\| \frac{\partial \boldsymbol{x}}{\partial \zeta} \times \frac{\partial \boldsymbol{x}}{\partial \eta} \right\|}; \qquad da(\zeta, \eta) = \left\| \frac{\partial \boldsymbol{x}}{\partial \zeta} \times \frac{\partial \boldsymbol{x}}{\partial \eta} \right\| d\zeta d\eta. \tag{B.2}$$

The location of the Gauss points in the parent coordinate system can be obtained as

$$\boldsymbol{x}_g = \sum_{a \in \Lambda_e^a} N_a(\zeta_g, \eta_g)\, \boldsymbol{x}_a. \tag{B.3}$$

A detailed discussion on finite element shape functions can be found in [54, 148].

B.1 Triangular element

The bilinear shape functions for a triangular element in the isoparametric domain can be expressed as

$$N_1 = 1 - \zeta - \eta, \tag{B.4}$$

$$N_2 = \zeta, \tag{B.5}$$

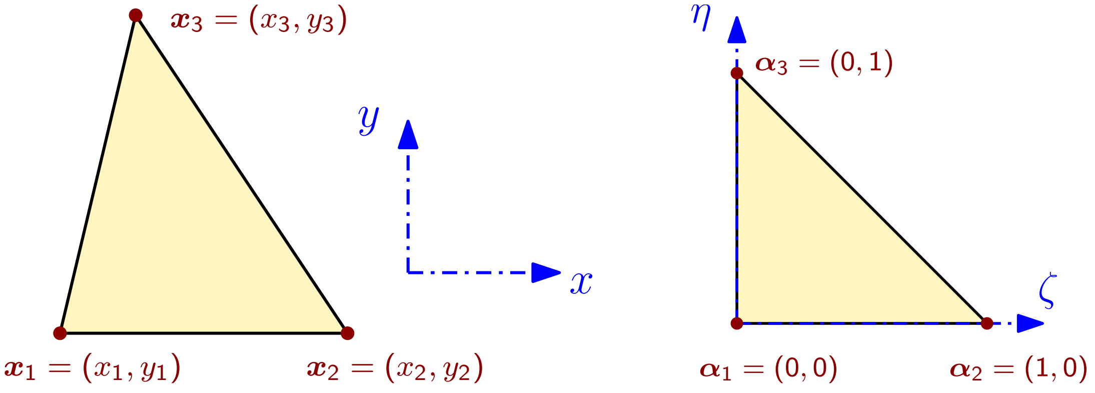
$$N_3 = \eta. \tag{B.6}$$

The gradient of shape functions can be obtained as

$$\nabla_{\boldsymbol{\alpha}} N_a^e(\zeta, \eta) = \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{B.7}$$

Using (B.1), we can express the Jacobian matrix for a triangular element as follows

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} = \sum_{a \in \Lambda_e^a} \boldsymbol{x}_a \otimes \nabla_{\boldsymbol{\alpha}} N_a^e(\zeta, \eta) = \begin{bmatrix} -x_1 + x_2 & -x_1 + x_3 \\ -y_1 + y_2 & -y_1 + y_3 \\ -z_1 + z_2 & -z_1 + z_3 \end{bmatrix}. \tag{B.8}$$



$\boldsymbol{x}_3 = (x_3, y_3)$
y
x
$\boldsymbol{x}_1 = (x_1, y_1)$
$\boldsymbol{x}_2 = (x_2, y_2)$
η
$\boldsymbol{\alpha}_3 = (0, 1)$
ζ
$\boldsymbol{\alpha}_1 = (0, 0)$
$\boldsymbol{\alpha}_2 = (1, 0)$

(a) Parent domain (b) Isoparametric domain

FIGURE B.1: Two dimensional (a) Parent and (b) Isoparametric domains for a triangular element

B.2 Quadrilateral element

We know that for a quadrilateral element, the bilinear shape function can be expressed in the isoparametric domain as

$$N_a = \frac{1}{4}(1+\zeta\zeta_a)(1+\eta\eta_a). \tag{B.9}$$

Using (B.1), we can express the Jacobian matrix for a quadrilateral element as follows

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\alpha}} = \sum_{a\in \Lambda_e^a} \begin{bmatrix} \frac{x_a\zeta_a}{4}(1+\eta\eta_a) & \frac{x_a\eta_a}{4}(1+\zeta\zeta_a) \\ \frac{y_a\zeta_a}{4}(1+\eta\eta_a) & \frac{y_a\eta_a}{4}(1+\zeta\zeta_a) \\ \frac{z_a\zeta_a}{4}(1+\eta\eta_a) & \frac{z_a\eta_a}{4}(1+\zeta\zeta_a) \end{bmatrix}. \tag{B.10}$$

B.3 Hexahedral element

The trilinear shape function for a hexahedral element at node a can be expressed in parametric coordinates as:

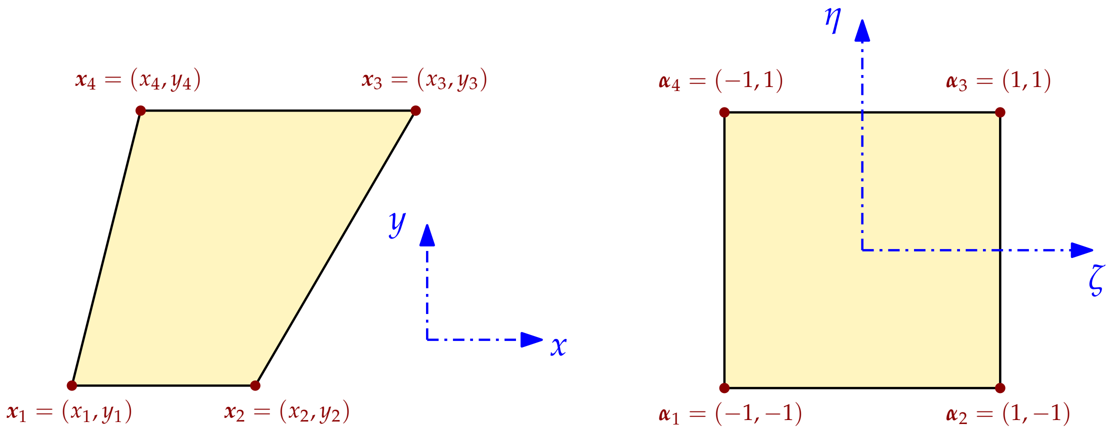
$$N_a = \frac{1}{8}(1+\zeta\zeta_a)(1+\eta\eta_a)(1+\mu\mu_a) \tag{B.11}$$

The derivatives of shape function can be computed as

$$\nabla_\zeta N_a = \frac{\zeta_a}{8}(1+\eta\eta_a)(1+\mu\mu_a), \tag{B.12}$$

$$\nabla_\eta N_a = \frac{\eta_a}{8}(1+\zeta\zeta_a)(1+\mu\mu_a), \tag{B.13}$$

$$\nabla_\mu N_a = \frac{\mu_a}{8}(1+\zeta\zeta_a)(1+\eta\eta_a). \tag{B.14}$$



(a) Parent domain

(b) Isoparametric domain

FIGURE B.2: Two dimensional (a) Parent and (b) Isoparametric domains for a quadrilateral element

Appendix C

OPENFOAM

In this section, the simulation workflow of using OpenFOAM for mixed large strain explicit solid dynamics is presented. One of the test cases (eg. L-shaped block) presented earlier in the numerical results is chosen as a reference in this appendix. The general simulation workflow in OpenFOAM is summarised in Fig. C.1.

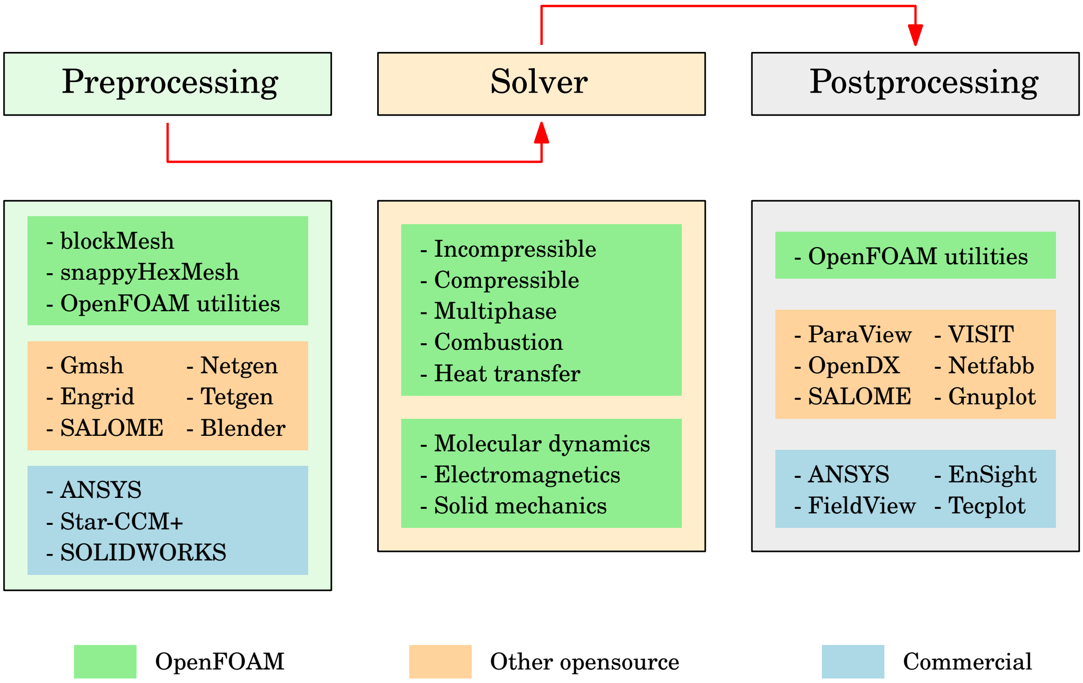
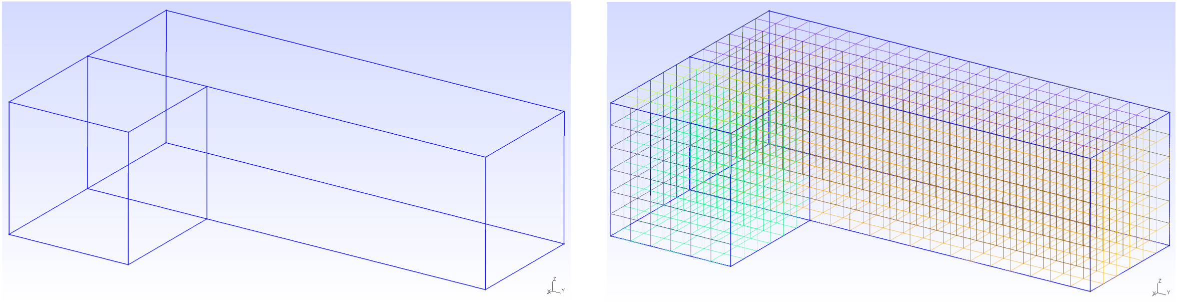


Figure C.1: Simulation workflow in OpenFOAM

C.1 Preprocessing

The geometry and mesh has been created using GMSH [127], a free 3D finite element mesh generator with a built-in CAD engine.This step will generate the necessary pre-processing files inside the `constant/polymesh` directory.



(a) Geometry

(b) Mesh

Figure C.2: Sample geometry and mesh created in GMSH.

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// DISCRETISATION INPUT
c = 2;      // 12x20x6 cells
//c = 4;    // 24x40x12 cells

// SECTION 1
Point(1) = {6, 0, 0};
Point(2) = {6, 3, 0};
Point(3) = {3, 3, 0};
Point(4) = {3, 0, 0};
Line(1) = {1,2};
Line(2) = {2,3};
Line(3) = {3,4};
Line(4) = {4,1};

Transfinite Line{1,2,3,4} = 3*c+1;
Line Loop(1) = {1,2,3,4};
Plane Surface(1) = {1};
Transfinite Surface{1} = {1,2,3,4};
Recombine Surface{1};

// SECTION 2
Point(5) = {3, 10, 0};
Point(6) = {0, 10, 0};
Point(7) = {0, 0,  0};
Line(5) = {3,5};
Line(6) = {5,6};
Line(7) = {6,7};
Line(8) = {7,4};

Transfinite Line{5} = 7*c+1;
Transfinite Line{6,8} = 3*c+1;
Transfinite Line{7} = 10*c+1;
Line Loop(2) = {5,6,7,8,-3};
Plane Surface(2) = {2};
Transfinite Surface{2} = {5,6,7,4};
Recombine Surface{2};

// EXTRUSION
Extrude{0,0,3}{
Surface{1,2};
Layers{3*c};Recombine;
}

// BOUNDARY CONDITIONS
Physical Surface("free")    = {17,21,29,30,1,40,48,44,52,57,2};

Physical Volume("volume") = {1,2};

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Listing C.1: lShapedBlock.geo

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

Mesh stats
    points:           1323
    faces:            3156
    internal faces:   2460
    cells:            936
    faces per cell:   6
    boundary patches: 1
    point zones:      0
    face zones:       0
    cell zones:       1

Overall number of cells of each type:
    hexahedra:     936
    prisms:        0
    wedges:        0
    pyramids:      0
    tet wedges:    0
    tetrahedra:    0
    polyhedra:     0

Checking topology...
    Boundary definition OK.
    Cell to face addressing OK.
    Point usage OK.
    Upper triangular ordering OK.
    Face vertices OK.
    Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
    Patch               Faces    Points   Surface topology
    free                696      698      ok (closed singly connected)

Checking geometry...
    Overall domain bounding box (0 0 0) (6 10 3)
    Mesh (non-empty, non-wedge) directions (1 1 1)
    Mesh (non-empty) directions (1 1 1)
    Boundary openness (0 0 0) OK.
    Max cell openness = 0 OK.
    Max aspect ratio = 1 OK.
    Minimum face area = 0.25. Maximum face area = 0.25.  Face area magnitudes OK.
    Min volume = 0.125. Max volume = 0.125.  Total volume = 117.  Cell volumes OK.
    Mesh non-orthogonality Max: 0 average: 0
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 1.9984014e-14 OK.
    Coupled point location match (average 0) OK.

Mesh OK.

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Listing C.2: checkMesh output

BIBLIOGRAPHY

[1] J. A. Trangenstein and P. Colella. A higher-order Godunov method for modeling finite deformation in elastic-plastic solids. *Communications on Pure and Applied Mathematics*, 44(1):41–100, 1991.

[2] J. A. Trangenstein and R. B. Pember. The Riemann problem for longitudinal motion in an elastic-plastic bar. *SIAM journal on scientific and statistical computing*, 12(1):180–207, 1991.

[3] J. A. Trangenstein and R. B. Pember. Numerical algorithms for strong discontinuities in elastic plastic solids. *Journal of Computational Physics*, 103(1):63–89, 1992.

[4] J. A. Trangenstein. A second-order Godunov algorithm for two-dimensional solid mechanics. *Computational mechanics*, 13(5):343–359, 1994.

[5] G. Kluth and B. Després. Discretization of hyperelasticity on unstructured mesh with a cell-centered Lagrangian scheme. *Journal of Computational Physics*, 229(24):9092–9118, 2010.

[6] 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*, 118: 13–38, 2013.

[7] P.-H. Maire, R. Abgrall, J. Breil, R. Loubère, and B. Rebourcet. A nominally second-order cell-centered Lagrangian scheme for simulating elastic-plastic flows on two-dimensional unstructured grids. *Journal of Computational Physics*, 235:626–665, 2013.

[8] 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*, 109(3):407–456, 2017. [Link].

[9] G. Georges, J. Breil, and P.-H. Maire. A 3D finite volume scheme for solving the Updated Lagrangian form of hyperelasticity. *International Journal for Numerical Methods in Fluids*, 84(1): 41–54, 2017.

[10] M. Aguirre, A. J. Gil, J. Bonet, and A. A. Carreño. A vertex centred finite volume Jameson–Schmidt–Turkel (JST) algorithm for a mixed conservation formulation in solid dynamics. *Journal of Computational Physics*, 259:672–699, 2014.

[11] 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*, 300:387–422, 2015.

[12] N. C. Nguyen and J. Peraire. Hybridizable Discontinuous Galerkin methods for partial differential equations in continuum mechanics. *Journal of Computational Physics*, 231:5955–5988, 2012.

[13] I. A. Karim, C. H. Lee, A. J. Gil, and J. Bonet. A two-step Taylor-Galerkin formulation for fast dynamics. *Engineering Computations*, 31(3):366–387, 2014.

[14] 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*, 276:659–690, 2014.

[15] 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*, 268:40–64, 2014.

[16] 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*, 283:689–732, 2015.

[17] 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*, 300: 146–181, 2016.

[18] 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*, 311:71–111, 2016.

[19] 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.

[20] B. Després and C. Mazeran. Lagrangian Gas Dynamics in Two Dimensions and Lagrangian systems. *Archive for Rational Mechanics and Analysis*, 178(3):327–372, 2005. ISSN 0003-9527.

[21] C. Hirsch. *Numerical computation of internal and external flows. Volume I: The fundamentals of computational fluid dynamics.* Elsevier, Second edition, 2007.

[22] M. Schäfer. *Computational engineering: Introduction to numerical methods.* Springer, 2006.

[23] N. P. Weatherill and O. Hassan. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 37(12):2005–2039, 1994.

[24] T. J. R. Hughes. *The finite element method: Linear static and dynamic finite element analysis.* Courier Corporation, 2012.

[25] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99(2):235–394, 1992. ISSN 0045-7825.

[26] D. J. Payen and K. J. Bathe. Improved stresses for the 4-node tetrahedral element. *Computers and Structures*, 89:1265–1273, 2011.

[27] D. J. Payen and K. J. Bathe. A stress improvement procedure. *Computers and Structures*, 112-113: 311–326, 2012.

[28] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 5(3):283–292, 1977. ISSN 1096-9845.

[29] W. L. Wood, M. Bossak, and O. C. Zienkiewicz. An alpha modification of Newmark's method. *International Journal for Numerical Methods in Engineering*, 15(10):1562–1566, 1980. ISSN 1097-0207.

[30] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics*, 60(2):371–375, June 1993. ISSN 0021-8936.

[31] D. D. Adams and W. L. Wood. Comparison of Hilber-Hughes-Taylor and Bossak 'α-methods' for the numerical integration of vibration equations. *International Journal for Numerical Methods in Engineering*, 19(5):765–771, 1983. ISSN 1097-0207.

[32] K. J. Bathe. *Finite element procedures.* Prentice Hall, 1996.

[33] 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*, 99:354–371, 2014.

[34] Y. Onishi and K. Amaya. Performance evaluation of the selective smoothed finite element methods using tetrahedral elements with deviatoric/hydrostatic split in large deformation analysis. *Theoretical and Applied Mechanics Japan*, 63:55–65, 2015.

[35] F. M. A. 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*, 20:569–583, 2004.

[36] L. R. G. Treloar. *The physics of rubber elasticity.* Oxford University Press, 1975.

[37] M. Mooney. A Theory of Large Elastic Deformation. *Journal of Applied Physics*, 11:582–592, 1940.

[38] L. Mullins and A. G. Thomas. *A theory of rubber-like elasticity: the chemistry and physics of rubber-like substance.* John Wiley and Sons, 1963.

[39] 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*, 47(9):1549–1568, 2000.

[40] 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*, 14(5):437–449, 1998.

[41] 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*, 17(8):551–561, 2001.

[42] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method : its basis & fundamentals*, volume 1. Butterworth-Heinemann, 6 edition, 2007.

[43] A. J. Gil and P. D. Ledger. A coupled *hp*-finite element scheme for the solution of two-dimensionalelectrostrictive materials. *International Journal for Numerical Methods in Engineering*, 91:1158–1183, 2012.

[44] D. Jin, P. D. Ledger, and A. J. Gil. An *hp*-fem framework for the simulation of electrostrictive and magnetostrictive materials. *Computers and Structures*, 133:131–148, 2014.

[45] D. Jin, P. D. Ledger, and A. J. Gil. *hp*-Finite Element solution of coupled stationary magnetohydrodynamics problems including magnetostrictive effects. *Computers & Structures*, 164:161–180, 2016.

[46] G. Scovazzi, B. Carnes, X. Zeng, and S. Rossi. A simple, stable, and accurate tetrahedral finite element for transient, nearly and fully incompressible solid dynamics: A dynamic variational multiscale approach. *International Journal for Numerical Methods in Engineering*, 106:799–839, 2015.

[47] P.-L. George and H. Borouchaki. Delaunay triangulation and meshing. 1998.

[48] P. J. Frey and P.-L. George. *Mesh generation: Application to finite elements.* ISTE, 2007.

[49] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear finite elements for continua and structures.* John Wiley and Sons, 2000.

[50] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press, Second edition, 2008.

[51] A. F. Bower. *Applied mechanics of solids.* CRC Press, 2010.

[52] T. Belytschko, J. S. Ong, W. K. Liu, and J. M. Kennedy. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 43:251–276, 1984.

[53] D. P. Flanagan and T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17(5):679–706, 1981.

[54] J. Donea and A. Huerta. *Finite element methods for flow problems.* John Wiley & Sons, 2003.

[55] M. W. Gee, C. R. Dohrmann, S. W. Key, and W. A. Wall. A uniform nodal strain tetrahedron with isochoric stabilization. *International Journal for Numerical Methods in Engineering*, 78(4): 429–443, 2009. ISSN 1097-0207.

[56] M. A. Puso and J. Solberg. A stabilized nodally integrated tetrahedral. *International Journal for Numerical Methods in Engineering*, 67(6):841–867, 2006. ISSN 1097-0207.

[57] E. A. Souza Neto, D. Perić, M. Dutko, and D. R. J. Owen. Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal of Solids and Structures*, 33:3277–3296, 1996.

[58] Y. D. Fryer, C. Bailey, M. Cross, and C.-H. Lai. A Control Volume procedure for solving the elastic stress-strain equations on an unstructured mesh. *Applied Mathematical Modelling*, 15 (11–12):639–645, 1991. ISSN 0307-904X.

[59] C. Bailey and M. Cross. A Finite Volume procedure to solve elastic solid mechanics problems in three dimensions on an unstructured mesh. *International Journal for Numerical Methods in Engineering*, 38(10):1757–1776, 1995. ISSN 1097-0207.

[60] A. K. Slone, C. Bailey, and M. Cross. Dynamic solid mechanics using Finite Volume Methods. *Applied Mathematical Modelling*, 27(2):69–87, 2003. ISSN 0307-904X.

[61] G. A. Taylor, C. Bailey, and M. Cross. A vertex-based Finite Volume Method applied to non-linear material problems in computational solid mechanics. *International Journal for Numerical Methods in Engineering*, 56(4):507–529, 2003. ISSN 1097-0207.

[62] H. Jasak and H. G. Weller. Application of the finite volume method and unstructured meshes to linear elasticity. *International Journal for Numerical Methods in Engineering*, 48(2):267–287, 2000.

[63] I. Bijelonja, I. Demirdžić, and S. Muzaferija. A finite volume method for incompressible linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:6378–6390, 2006.

[64] P. Cardiff, A. Karač, and A. Ivanković. Development of a finite volume contact solver based on the penalty method. *Computational Materials Science*, 64:283–284, 2012.

[65] P. Cardiff, A. Karač, and A. Ivanković. A large strain finite volume method for orthotropic bodies with general material orientations. *Computer Methods in Applied Mechanics and Engineering*, 268:318–335, 2014.

[66] P. Cardiff, Z. Tuković, P. De Jaeger, M. Clancy, and A. Ivanković. A Lagrangian cell-centred finite volume method for metal forming simulation. *International Journal for Numerical Methods in Engineering*, 2016.

[67] P. W. Randles and L. D. Libersky. Smoothed particle hydrodynamics: Some recent improvements and applications. *Computer methods in applied mechanics and engineering*, 139(1-4):375–408, 1996.

[68] G. R. Johnson and S. R. Beissel. Normalized smoothing functions for SPH impact computations. *International Journal for Numerical Methods in Engineering*, 39(16):2725–2741, 1996.

[69] Y. Vidal, J. Bonet, and A. Huerta. Stabilized Updated Lagrangian corrected SPH for explicit dynamic problems. *International Journal for Numerical Methods in Engineering*, 69(13):2687–2710, 2007.

[70] J. Bonet and S. Kulasegaram. Remarks on tension instability of Eulerian and Lagrangian corrected smooth particle hydrodynamics (CSPH) methods. *International Journal for Numerical Methods in Engineering*, 52(11):1203–1220, 2001.

[71] G. C. Ganzenmüller. An hourglass control algorithm for Lagrangian smooth particle hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 286:87–106, 2015.

[72] J. W. Swegle, D. L. Hicks, and S. W. Attaway. Smoothed particle hydrodynamics stability analysis. *Journal of computational physics*, 116(1):123–134, 1995.

[73] T. Belytschko, Y. Guo, W. K. Liu, and S. P. Xiao. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering*, 48(9):1359–1400, 2000.

[74] Joseph J Monaghan. SPH without a tensile instability. *Journal of Computational Physics*, 159 (2):290–311, 2000.

[75] MA Puso, JS Chen, E Zywicz, and W Elmer. Meshfree and finite element nodal integration methods. *International Journal for Numerical Methods in Engineering*, 74(3):416–446, 2008.

[76] G Carré, S Del Pino, Bruno Després, and Emmanuel Labourasse. A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228(14):5160–5183, 2009.

[77] Pierre-Henri Maire, Rémi Abgrall, Jérôme Breil, and Jean Ovadia. A Cell-Centered Lagrangian Scheme for Two-Dimensional Compressible Flow Problems. *SIAM Journal on Scientific Computing*, 29(4):1781–1824, 2007.

[78] Pierre-Henri Maire. A high order cell-centered Lagrangian scheme for two-dimensional cylindrical geometry. *Journal of Computational Physics*, 228:6882–6915, 2009.

[79] G. Georges, J. Breil, and P. H. Maire. A 3D GCL compatible cell-centered Lagrangian scheme for solving gas dynamics equations. *Journal of Computational Physics*, 305:921–941, 2016.

[80] D. E. Burton, T. C. Carney, N. R. Morgan, S. K. Sambasivan, and M. J. Shashkov. A cell-centered Lagrangian Godunov-like method for solid dynamics. *Computers and Fluids*, 83:33–47, 2013.

[81] S. Rossi, N. Abboud, and G. Scovazzi. Implicit finite incompressible elastodynamics with linear finite elements: A stabilized method in rate form. *Computer Methods in Applied Mechanics and Engineering*, 311:208–249, 2016.

[82] G. Scovazzi, M. A. Christon, T. J. R. Hughes, and J. N. Shadid. Stabilized shock hydrodynamics: I. A Lagrangian method. *Computer Methods in Applied Mechanics and Engineering*, 196(4):923–966, 2007.

[83] G. Scovazzi. Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations. *Computer Methods in Applied Mechanics and Engineering*, 196(4):967–978, 2007.

[84] G. Scovazzi. Galilean invariance and stabilized methods for compressible flows. *International journal for numerical methods in fluids*, 54(6-8):757–778, 2007.

[85] G. Scovazzi. A discourse on Galilean invariance, SUPG stabilization, and the variational multiscale framework. *Computer Methods in Applied Mechanics and Engineering*, 196(4):1108–1132, 2007.

[86] G. Scovazzi, E. Love, and M. J. Shashkov. Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations. *Computer methods in applied mechanics and engineering*, 197(9):1056–1079, 2008.

[87] G. Scovazzi, J. N. Shadid, E. Love, and W. J. Rider. A conservative nodal variational multiscale method for Lagrangian shock hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 199(49):3059–3100, 2010.

[88] G. Scovazzi and E. Love. A generalized view on Galilean invariance in stabilized compressible flow computations. *International Journal for Numerical Methods in Fluids*, 64(10-12):1065–1083, 2010.

[89] G. Scovazzi. Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach. *Journal of Computational Physics*, 231(24):8029–8069, 2012.

[90] C. Shu and Osher. Efficient implementation of Essentially Non-oscillatory Shock Capturing Schemes. *Journal of Computational Physics*, 77:439–471, 1988.

[91] J. Donea, A. Huerta, J.-Ph Ponthot, and A. Rodriguez Ferran. Encyclopedia of Computational Mechanics Vol. 1: Fundamentals., Chapter 14: Arbitrary Lagrangian-Eulerian Methods, 2004.

[92] A. J. Gil, A. A. Carreño, J. Bonet, and O. Hassan. The immersed structural potential method for haemodynamic applications. *Journal of Computational Physics*, 229(22):8613–8641, 2010.

[93] A. J. Gil, , A. A. Carreño, J. Bonet, and O. Hassan. An enhanced Immersed Structural Potential Method for fluid–structure interaction. *Journal of Computational Physics*, 250:178–205, 2013.

[94] J. Bonet, A. J. Gil, and R. Ortigosa. A computational framework for polyconvex large strain elasticity. *Computer Methods in Applied Mechanics and Engineering*, 283:1061–1094, 2015.

[95] R. de Boer. *Vektor-und Tensorrechnung für Ingenieure*. Springer-Verlag, 1982.

[96] 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*, 84:49–63, 2016.

[97] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: A practical introduction*. Springer-Verlag, Third edition, 2009.

[98] R. Abedi and R. B. Haber. Riemann solutions and spacetime discontinuous Galerkin method for linear elastodynamic contact. *Computer Methods in Applied Mechanics and Engineering*, 270: 150–177, 2014.

[99] G. H. Miller. An iterative Riemann solver for systems of hyperbolic conservation laws, with application to hyperelastic solid mechanics. *Journal of Computational Physics*, 193:198–225, 2003.

[100] G. A. Holzapfel. *Nonlinear solid mechanics: A continuum approach for engineering*. Wiley and Sons, 2000.

[101] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal of Numerical Methods in Engineering*, 47:1189–1124, 2000.

[102] R. J. LeVeque. *Numerical methods for conservation laws*, volume 132. Springer, 1992.

[103] R. J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge University Press, 2002.

[104] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA*, 89-0366, 1989.

[105] J. Blazek. *Computational Fluid Dynamics: Principles and applications*. Elsevier, 2005.

[106] H. K. Versteeg and W. Malalasekera. *An introduction to Computational Fluid Dynamics: The Finite Volume Method* . Pearson education limited, 2 edition, 2007.

[107] M. Torrilhon. Locally divergence-preserving upwind finite volume schemes for magnetohydrodynamic equations. *SIAM Journal on Scientific Computing*, 26(4):1166–1191, 2005.

[108] B. Després and E. Labourasse. Angular Momentum preserving cell-centered Lagrangian and Eulerian schemes on arbitrary grids. *Journal of Computational Physics*, 290:28–54, 2015.

[109] M. E. Hubbard. Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. *Journal of Computational Physics*, 155(1):54–74, 1999.

[110] C. M. Dafermos. *Quasilinear hyperbolic systems with involutions*. Springer, 1989.

[111] P. Wriggers and T. A. Laursen. *Computational contact mechanics*, volume 30167. Springer, 2006.

[112] T. A. Laursen. *Computational contact and impact mechanics: Fundamentals of modeling interfacial phenomena in nonlinear finite element analysis*. Springer Science & Business Media, 2013.

[113] H. Guillard and C. Viozat. On the behaviour of upwind schemes in the low Mach number limit. *Computers & fluids*, 28(1):63–86, 1999.

[114] H. Guillard. On the behavior of upwind schemes in the low Mach number limit. IV: P0 approximation on triangular and tetrahedral cells. *Computers & Fluids*, 38(10):1969–1972, 2009.

[115] H. Guillard and A. Murrone. On the behavior of upwind schemes in the low Mach number limit: II. Godunov type schemes. *Computers & fluids*, 33(4):655–675, 2004.

[116] S. P. Timoshenko and J. N. Goodier. *Theory of elasticity*. McGraw-Hill, 1970.

[117] R. Jeltsch and M. Torrilhon. On curl-preserving finite volume discretizations for shallow water equations. *BIT Numerical Mathematics*, 46(1):35–53, 2006.

[118] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T.s Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for the MHD equations. *Journal of Computational Physics*, 175(2):645–673, 2002.

[119] G. Tóth. The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes. *Journal of Computational Physics*, 161(2):605–652, 2000.

[120] J. U. Brackbill and D. C. Barnes. The effect of nonzero $\nabla \cdot B = 0$ on the numerical solution of the magnetohydrodynamic equations. *Journal of Computational Physics*, 35(3):426–430, 1980.

[121] B. Engquist and S. Osher. One-sided difference approximations for nonlinear conservation laws. *Mathematics of Computation*, 36(154):321–351, 1981.

[122] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *Mathematische Annalem*, 100:32–74, 1928.

[123] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631, 1998.

[124] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, 2000.

[125] T. Maric, J. Hopken, and K. Mooney. *The OpenFOAM technology primer*. Sourceflux, 2014.

[126] OpenFOAM user guide, 2014.

[127] C. Geuzaine and J.-F. Remacle. Gmsh: A 3D finite element mesh generator with built-in pre- and post-processing facilities. *International journal for numerical methods in engineering*, 79(11): 1309–1331, 2009.

[128] SolidWorks website. [Link].

[129] ParaView website. [Link].

[130] OpenFOAM website. [Link].

[131] S. K. Lahiri, J. Bonet, J. Peraire, and L. Casals. A variationally consistent fractional time-step integration method for incompressible and nearly incompressible Lagrangian dynamics. *International Journal for Numerical Methods in Engineering*, 63:1371–1395, 2005.

[132] J. C. Simo and N. Tarnow. The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 43(5):757–792, 1992.

[133] X. N. Meng and T. A. Laursen. Energy consistent algorithms for dynamic finite deformation plasticity. *Computer Methods in Applied Mechanics and Engineering*, 191(15):1639–1675, 2002.

[134] L. Noels, L. Stainier, and JP. Ponthot. An energy–momentum conserving algorithm for non-linear hypoelastic constitutive models. *International Journal for Numerical Methods in Engineering*, 59 (1):83–114, 2004.

[135] S. Dong. BDF-like methods for nonlinear dynamic analysis. *Journal of Computational Physics*, 229(8):3019–3045, 2010.

[136] G. Taylor. The use of flat-ended projectiles for determining dynamic yield stress. I. Theoretical considerations. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 194, pages 289–299. The Royal Society, 1948.

[137] J. C. Simo. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Computer Methods in Applied Mechanics and Engineering*, 99(1):61–112, 1992.

[138] A. Huerta and F. Casadei. New ALE applications in non-linear fast-transient solid dynamics. *Engineering Computations*, 11(4):317–345, 1994.

[139] D. Sulsky, S.-J. Zhou, and H. L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer physics communications*, 87(1):236–252, 1995.

[140] O. C. Zienkiewicz, J. Rojek, R. L. Taylor, and M. Pastor. Triangles and Tetrahedra in Explicit Dynamic Codes for Solids. *International Journal for Numerical Methods in Engineering*, 43:565–583, 1998.

[141] A. Rodríguez Ferran, F. Casadei, and A. Huerta. ALE stress update for transient and quasistatic processes. *International Journal for Numerical Methods in Engineering*, 43(2):241–262, 1998.

[142] J. Bonet, S. Kulasegaram, M. X. Rodriguez Paz, and M. Profit. Variational formulation for the Smooth Particle Hydrodynamics (SPH) simulation of fluid and solid problems. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1245–1256, 2004.

[143] J. W. Swegle. SPH behavior in tension. *Memo (13 August 1992)*, 1992.

[144] J. P Gray, J. J. Monaghan, and R. P. Swift. SPH elastic dynamics. *Computer Methods in Applied Mechanics and Engineering*, 190(49):6641–6662, 2001.

[145] S. K. Lahiri, J. Bonet, and J. Peraire. A variationally consistent mesh adaptation method for triangular elements in explicit Lagrangian dynamics. *International Journal for Numerical Methods in Engineering*, 82:1073–1113, 2010.

[146] F. Rieper. A low-Mach number fix for Roe's approximate Riemann solver. *Journal of Computational Physics*, 230(13):5263–5287, 2011.

[147] A. Rodríguez Ferran, A. Pérez Foguet, and A Huerta. Arbitrary Lagrangian Eulerian (ALE) formulation for hyperelastoplasticity. *International Journal for Numerical Methods in Engineering*, 53(8):1831–1851, 2002.

[148] B. Szabo and I. Babuska. *Finite element analysis.* John Wiley and Sons, 1991.