A COUPLED EULERIAN-LAGRANGIAN EXTENDED FINITE ELEMENT FORMULATION FOR MOVING INTERFACE PROBLEMS AND DAMAGE TRANSPORT IN HYPERELASTIC MEDIA

By

Anup Aryal

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

CIVIL ENGINEERING

August, 2014

Nashville, Tennessee

Approved:

Professor Ravindra Duddu

Professor Caglar Oskay

To my family, teachers and friends

ACKNOWLEDGMENTS

I feel proud to be a member of the Vanderbilt family. I am indebted to the Civil Engineering department for providing me full scholarship to pursue a Masters degree. There are many people to whom I want to express my gratitude for supporting me to accomplish the work presented here. First and foremost, this thesis would not have been complete without the continuous support and guidance of my research advisor Dr. Ravindra Duddu. Dr. Duddu always encouraged me to think in a novel way so as to hone my problem solving skills. He was always available to guide me on the problems encountered during the research.

I would like to thank Prof. P.K. Basu for being my guardian during my stay at Vanderbilt. I always enjoyed talking with him, no matter whether we talked about academic or any other non academic subject matters. I would also like to thank Louis Foucard and Prof. Franck Vernerey from University of Colorado, Boulder for sharing their matlab code and for the fruitful discussions on the research work presented in this thesis.

My stay at Vanderbilt would not have been such pleasure and enjoyable without friends like Saideep, Chenzhao, Xiang and Guowei. I am grateful to all professors, staffs and students of civil engineering department for making my two years here at Vanderbilt memorable.

Last but not least, I thank Bishal, Kaji, Dev, Suraj and Nashpal group for regular support and encouragement. I feel Nashville as my second home just because of you all.

Thank you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Motivation

The present work details the formulation of coupled Eulerian- Lagrangian extended finite element method for moving interface problems and damage transport in hyperelastic material. Hyperelastic or Green elastic material is a type of material whose stress-strain relationship is derived from strain energy density function. The simplest example of such kind of material is rubber which undergoes large deformation when subjected to extreme loading. These materials have been profoundly used in automotives, medical devices, etc. There are many important and challenging problems in the areas of geophysics (e.g. ice sheet flow, mantle dynamics), soft materials (e.g. deformation of hydrogels and biological cells) and material science (e.g. metal forming) which involve large deformations or flow of solid material. In this thesis, we focus on soft hyperelastic solids with the goal of eventually modeling and characterizing hydrogels. A standard Lagrangian finite element formulation has difficulty in simulating very large deformations in soft solids because of mesh distortion. Instead, it can be convenient to work with a fully Eulerian description of solid deformation [1, 2], especially, when the domain boundaries are not moving. For problems where domain boundaries are free to move, along with the Eulerian description, a Lagrangian (material) description is required to map solid deformation between reference and current configurations because after updating the position of interface (or the solid body's boundary), some nodes close to the interface do not remain in the elastic body. Such a moving boundary problem also needs the introduction of specialized numerical methods that can track an interface without remediating to expensive remeshing techniques so as to save lots of computational time and effort. Moreover, these solids may have some distributed defects in the materials which can grow and affect their strength, stability, residual

life and ultimately cause damage. So, the behavior of defects and cracks within materials has always been of interest to the researchers. However, in an Eulerian scheme, since the material enters and leaves the computational domain, so does damage along with the material; therefore, it becomes important not only to evolve damage but also consistently advect it according to the solid's flow velocity. In this thesis, we propose to address the challenges with describing the evolution of free boundaries through the introduction of a coupled Eulerian–Lagrangian formulation and a combined numerical method based on the extended finite element method (XFEM) and the grid based particle method (GPM) [3]. In addition, in order to evolve or transport damage features, such as voids and cracks at the microscale, a continuum damage mechanics formulation is explored.

1.2 Overview of solid mechanics formulation

Two classical descriptions of motion: the Lagrangian description and the Eulerian description are often used to describe the numerical simulation of multidimensional problems in solid mechanics. Arbitrary Lagrangian-Eulerian (ALE) method [4] combines the advantages of both above mentioned formulations. These formulations are shortly described now.

1.2.1 Lagrangian and Eulerian Formulation

In the Lagrangian finite element description, nodes are coincident with the material points and hence nodes and material points move together. Each particle is assigned with the quantities of flow at varying time. With this formulation, it is easier to track free surfaces precisely and hence apply boundary conditions; however, under high strains, the mesh becomes severely distorted thus giving a poor quality solution. Frequent remeshing is necessary in order to avoid a badly distorted mesh, which increases computational time. On the other hand, in the Eulerian finite element description, nodes stay fixed while material flows through the mesh. Rather than following each particle as in Lagrangian description,

here, properties are recorded at each space point at different times thus recording the flow evolution. The advantage of Eulerian approach is that the mesh undergoes no distortion thus, large deformations can be easily handled. Difficulty in tracking the free surfaces as boundary nodes do not coincide with the boundary can be considered a disadvantage of the Eulerian formulation. Consequently, boundary conditions need to be assigned at points which are not finite element nodes.

1.2.2 Arbitrary Lagrangian Eulerian (ALE) Formulation

As the name suggests, this method uniquely combines the Lagrangian and the Eulerian formulation [5]. The computational mesh inside the domain moves arbitrarily to optimize the shape of the elements while the mesh on the boundaries and interfaces of the domain moves along with the material to precisely track the boundaries and interfaces[5]. The ALE formulation can be reduced to either the Eulerian formulation by fixing the mesh in space or the Lagrangian formulation by equating mesh motion to material motion.

1.3 Background

Traditionally, a purely Lagrangian finite element formulation is used for solving myraid of solid mechanics problems. The Lagrangian formulation is simple to implement, less expensive computationally and as the particles are attached with the materials, it does not need any interface-tracking techniques; however, in the case of severe material distortion, it may suffer from numerical issues due to excessive mesh distortion. The use of remeshing with interpolation techniques between old and new meshes [6, 7, 8] may somewhat deal this problem, but this could be less accurate and computationally challenging. Another approach consists of using the Deforming Spatial Domain or Stabilized Space Time (DSD/DST) formulation [9, 10] proposed to handle fluid-structure interactions with large deformation of the fluid-solid interface. A more popular approach is to employ the arbitrary Lagrangian–Eulerian (ALE) formulation, wherein, a deformation step on a distorted

Lagrangian mesh is followed with a remap step onto a spatially fixed Eulerian mesh. Some early work on ALE formulations can be found in [11, 12, 13, 5, 14, 15, 16] and for a complete literature review of ALE and Space-Time methods for moving boundaries and interfaces, the reader is referred to [17]. In both the ALE and DSD/DST formulations, the main objective is to reduce the frequency of remeshing [18], thus decrease the computational efforts. A purely Eulerian formulation has also been used for solving solid mechanics problems; however, it is less popular compared to the purely Lagrangian or ALE formulations. A handful of Eulerian formulations have been proposed in the literature [1, 19, 20, 2, 21]. Benson [14] has pointed out two main advantages of Eulerian approaches: (1) it can handle arbitrarily large deformations, so it can be suitable for studying soft matter and viscoelastic fluids and (2) it allows the creation, merging and vanishing of free surfaces or interfaces in a natural manner, so it can be used for studying growth and phase transformation processes. In the Eulerian approach, material flows through the underlying mesh so mesh distortion is not an issue. However, one of the drawbacks of the Eulerian formulation for solids is its higher computational cost due to the need for computing velocity and deformation variables separately, as opposed to a Lagrangian formulation, wherein only the velocity needs to be computed. In three-dimensions, this means in an Eulerian finite element formulation for a compressible hyperelastic medium, there will be 13 unknown (3 velocity, 9 deformation gradient, 1 Jacobian determinant) nodal degrees of freedom (DOFs), whereas in a Lagrangian finite element formulation there will be only 3 unknown (3 velocity) nodal DOFs [2]. Moreover, in Eulerian formulations material interfaces and free boundaries need to be tracked using moving interface methods, which adds to the numerical challenge and computational expense. But, as remeshing is not needed, this indeed saves computational time. In case of moving interface problems, tracking the deformation-driven motion of the interface can be accomplished by for instance, using the grid based particle method (GPM) [3] or, alternatively, using an interface-capturing scheme such as the level set method [22, 23]. Among the two options, the GPM is computationally

less expensive, simpler to implement, and can also handle arbitrary topological transitions such as merging and vanishing of material interfaces. Another issue arising from the existence of a moving interface is the imposition of interface constraints, however, recent work provides viable weak formulations by employing either the Lagrange multiplier method [24, 25] or a Nitsche's method [26, 27]. Therefore, the imposition of interface conditions is not currently addressed herein and the reader is refereed to the above cited work.

1.4 Objectives and Strategy

The primary objective of this work is to develop a coupled Eulerian-Lagrangian (CEL) formulation and level set representation of moving interface associated with the large deformations of soft solids. The secondary objective is to model the evolution or transport of damage within the continuum mechanics framework using the Eulerian, updated Lagrangian and total Lagrangian descriptions. This is accomplished by solving the momentum equations and the transport equations for the deformation gradient in a staggered manner in time. The velocity field is first calculated by solving the momentum equation in an Eulerian framework, and is then used to update the isochoric and volumetric parts of the deformation gradient, separately, using an updated Lagrangian description. The position of the material interface is tracked using the GPM [3] and the velocity field projected in the direction normal to the interface. The standard finite element shape function is enriched with a Heaviside step function which allows the incorporation of a sharp discontinuity in the material properties across the embedded interface within a finite element. Finally, a single scalar continuum damage variable is introduced into the formulation that averages the effect of microcracks and microvoids in the material microstructure at the macroscale. Only the advection of damage is studied herein, but not damage growth.

1.5 Organization of Thesis

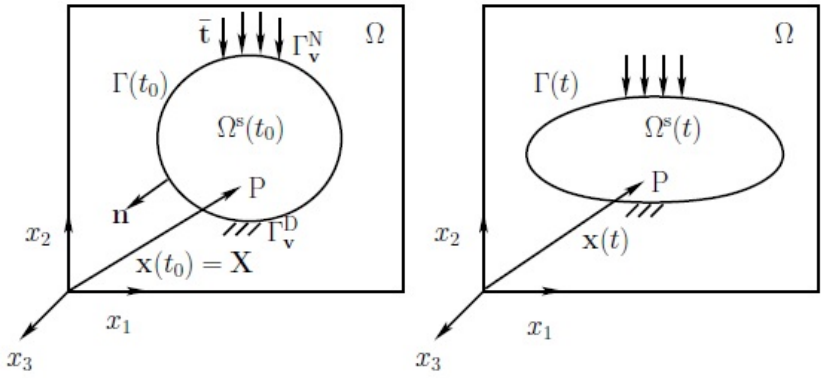
The organization of the thesis is as follows: Chapter 2 introduces the kinematics, the governing and constitutive equations and the resulting weak form for the mechanical equilibrium of an elastic body. In chapter 3, we present a numerical strategy to discretize the weak form, the tracking of the interface and the Lagrangian transport of the deformation gradient tensor components. We then introduce the notion of damage and the various approaches for damage transport. In chapter 4, we describe hyperfoam and hyperelastic material to model large deformation in commercial software Abaqus with the help of both in-built material models and user defined subroutines. Finally, the numerical convergence and accuracy of the coupled Eulerian-Lagrangian method is considered in chapter 5 through the examples of a uniaxial extension of a rectangular bar and the simple shear of a rectangular block. The mesh-independent geometric discretization and the absence of mesh distortion issue are then demonstrated with the split cylinder test set up and the indentation of a rectangular block. The latter results are validated by comparing them with those from traditional Lagrangian formulation in the commercial software Abaqus. Another example problem illustrating damage transport within hyperelastic material is presented by employing the Eulerian, updated Lagrangian and total Lagrangian descriptions. Some concluding remarks along with the future work are given in chapter 6.

CHAPTER 2

Formulation of the governing equations

2.1 Kinematics

In this study, we consider a domain Ω containing an elastic body in the region $\Omega^s(t)$. The domain Ω is delimited by a boundary $\partial\Omega$ while the interface describing the current shape of the elastic body is denoted by $\Gamma(t)$. Thus, Γ splits the domain Ω into the solid domain $\Omega^s(t)$ and its complement denoted by $\Omega \setminus \Omega^s(t)$. We employ the Eulerian description of the motion and choose a fixed right-handed Cartesian system of coordinates $\{\mathbf{x} = x_j\ \hat{e}_j, j = 1,2,3\}$, where $\hat{e}_j$ are the orthonormal basis vectors [28]. The motion of a physical particle P is expressed by the mapping function $\mathbf{x} = \boldsymbol{\chi}(\mathbf{X},t)$ between its reference coordinates $\{\mathbf{X} = X_j\ \hat{e}_j, j = 1,2,3\}$ at an initial time $t = t_0$ and its current coordinates $\mathbf{x}$ at a subsequent time $t > t_0$ (see Fig. 2.1). We assume that the function $\boldsymbol{\chi}(\mathbf{X},t)$ is sufficiently differentiable and single valued within the region $\Omega^s(t)$. From an Eulerian viewpoint, the spatial velocity field



(a)

Figure 2.1: Schematic diagram of the domain and its evolution under an applied traction. The moving interface $\Gamma(t)$ at any time t separates the solid domain $\Omega^s(t)$ from its complement $\Omega \setminus \Omega^s(t)$.

$\mathbf{v}(\mathbf{x},t)$ gives the velocity of the particle located at $\mathbf{x}$ at time t and is defined as

$$\mathbf{v} = \left(\frac{\partial \boldsymbol{\chi}}{\partial t}\right)_{\mathbf{X}} = \left(\frac{\partial \mathbf{x}}{\partial t}\right)_{\mathbf{X}} = \mathbf{g}\left[\mathbf{x}(\mathbf{X},t),t\right], \tag{2.1}$$

where $\mathbf{g}$ is a sufficiently differentiable function. The deformation of a solid particle is described by the tensor given by $\mathbf{F} = \dfrac{\partial \mathbf{x}}{\partial \mathbf{X}}$. The isochoric part of this tensor $\hat{\mathbf{F}}$ is given by:

$$\mathbf{F} = J^{-1/3}\,\hat{\mathbf{F}}, \tag{2.2}$$

where $J = \det[\mathbf{F}]$ is the Jacobian determination of the deformation. The rate of change of the deformation gradient is given by,

$$\dot{\mathbf{F}} = \frac{\mathrm{d}\mathbf{F}}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}}\right) = \left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right)\cdot\left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}}\right) = \mathbf{L}\mathbf{F}, \tag{2.3}$$

where $\dfrac{\mathrm{d}}{\mathrm{d}t}$ denotes the material time derivative and $\mathbf{L} = \dfrac{\partial \mathbf{v}}{\partial \mathbf{x}} = (\nabla \mathbf{v})^T$ is the velocity gradient with respect to the current coordinates. We note here that the superscript T is used for the transpose of a tensor. The above equation can be split into its volumetric and isochoric parts as,

$$\frac{\mathrm{d}\hat{\mathbf{F}}}{\mathrm{d}t} = \frac{\partial \hat{\mathbf{F}}}{\partial t} + \mathbf{v}\cdot\nabla\hat{\mathbf{F}} = \left[\mathbf{L} - \frac{1}{3}(\nabla\cdot\mathbf{v})\mathbf{I}\right]\hat{\mathbf{F}}, \tag{2.4}$$

$$\frac{\mathrm{d}J}{\mathrm{d}t} = \frac{\partial J}{\partial t} + \mathbf{v}\cdot\nabla J = J\,\nabla\cdot\mathbf{v}, \tag{2.5}$$

where $\nabla = \dfrac{\partial}{\partial x_j}\hat{e}_j$ is the spatial gradient operator, $\mathbf{I}$ is the second order identity tensor and $\cdot$ denotes the dot product.

2.2 Strong from

The strong form of the governing equations, which include both mechanical equilibrium and transport equations, can be written in the elastic body in $\Omega^{\mathrm{s}}(t)$ as [2]:

$$\begin{aligned} \nabla\cdot\boldsymbol{\sigma}+\rho\mathbf{f} &= \mathbf{0}, && (2.6)\\ \frac{\mathrm{d}J}{\mathrm{d}t}-J\nabla\cdot\mathbf{v} &= 0, && (2.7)\\ \frac{\mathrm{d}\hat{\mathbf{F}}}{\mathrm{d}t}-\left[\nabla\mathbf{v}-\frac{1}{3}(\nabla\cdot\mathbf{v})\mathbf{I}\right]\hat{\mathbf{F}} &= \mathbf{0}, && (2.8) \end{aligned}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\mathbf{f}$ the body force per unit volume in the current configuration. The above equations are subjected to the following boundary and initial conditions:

$$\begin{aligned} \mathbf{v} &= \bar{\mathbf{v}} \text{ on } \Gamma^{\mathrm{D}}_{\mathbf{v}}, && (2.9)\\ \mathbf{n}\cdot\boldsymbol{\sigma} &= \bar{\mathbf{t}} \text{ on } \Gamma^{\mathrm{N}}_{\mathbf{v}}, && (2.10)\\ \hat{\mathbf{F}}(t=0) &= \mathbf{I} \text{ in } \Omega^{\mathrm{s}}(0), && (2.11)\\ J(t=0) &= 1 \text{ in } \Omega^{\mathrm{s}}(0), && (2.12) \end{aligned}$$

where $\Gamma^{\mathrm{D}}_{\mathbf{v}}$ and $\Gamma^{\mathrm{N}}_{\mathbf{v}}$ represent the Dirichlet and Neumann parts of Γ for the boundary conditions on velocity and traction, respectively.

Remark 1 *If the transport equations (2.7) and (2.8) are discretized using an Eulerian description, one needs to specify boundary conditions on* $\hat{\mathbf{F}}$ *and J, in addition to initial conditions (2.11) and (2.12). To simplify our analysis, we propose here to use a Lagrangian (particle) description to update* $\hat{\mathbf{F}}$ *and J and does not necessitate the use of above boundary conditions. Ultimately, the formulation only requires initial conditions on* $\hat{\mathbf{F}}$ *and J.*

2.3 Constitutive equations

To model the large deformations of the compressible hyperelastic domain, we consider the isochoric-volumetric decomposition of the strain energy function as,

$$W = U(J) + \hat{W}(\hat{\mathbf{b}}), \tag{2.13}$$

where $\hat{\mathbf{b}} = \hat{\mathbf{F}}\hat{\mathbf{F}}^T = J^{-2/3}\mathbf{F}\mathbf{F}^T$. The specific functional forms of U and $\hat{W}$ are to be chosen to satisfy physical conditions. Herein, we assume the functions proposed by Simo et al. [29, 30] as,

$$\begin{aligned} U(J) &= \frac{\kappa}{2}[\,\ln(J)\,]^2, \\ \hat{W}(\hat{\mathbf{b}}) &= \frac{\mu}{2}\left[\mathrm{tr}(\hat{\mathbf{b}}) - 3\right], \end{aligned} \tag{2.14}$$

where 'tr' denotes the trace of the tensor and $\hat{\mathbf{b}} = \hat{\mathbf{F}}\hat{\mathbf{F}}^T$, μ and κ represent the shear and bulk modulus of the material, respectively and are calculated from Young's modulus (E) and Poisson's ratio (ν) by,

$$\kappa = \frac{E}{3(1-2\nu)} \;\text{ and }\; \mu = \frac{E}{2(1+\nu)} \tag{2.15}$$

The expression for the Cauchy stress is [2],

$$\boldsymbol{\sigma}(J,\hat{\mathbf{F}}) = \frac{1}{J}\left[\kappa \ln(J)\mathbf{I} + \mu\ \mathrm{dev}(\hat{\mathbf{b}})\right] \tag{2.16}$$

where $\mathrm{dev}(\hat{\mathbf{b}}) = \hat{\mathbf{b}} - \frac{1}{3}\mathrm{tr}(\hat{\mathbf{b}})\mathbf{I}$ is deviatoric part.

Remark 2 *The above functional form for $U(J)$ is chosen so that it satisfies several requirements [31]. First, in the limit case when Ω^s is compressed to a single point or is stretched*

to be infinitely large the strain energy always tends to positive infinity, that is,

$$\lim_{J\to+0} U = +\infty \;\; \textit{and} \;\; \lim_{J\to+\infty} U = +\infty. \tag{2.17}$$

Second, the volumetric stress tends to negative infinity when Ω^{s} *is compressed to a single point and to positive infinity when stretched to infinitely large, that is,*

$$\lim_{J\to+0} \frac{\partial U}{\partial J} = -\infty \;\; \textit{and} \;\; \lim_{J\to+\infty} \frac{\partial U}{\partial J} = +\infty. \tag{2.18}$$

Hence U *has to be an even power of* $\ln(J)$ *so that* $U > 0$ *for all* J.

Remark 3 *As pointed out in [31] the assumed volumetric part of the strain energy function* U *does not satisfy the polyconvexity because* $\dfrac{\partial^2 U}{\partial J^2} < 0$ *for* $\ln(J) > 1$*, that is, for all* $J > e$ *where* e *is Euler's number. However, this inconsistency is not an issue in the current work as all our investigations are carried out for* $J < e$.

2.4 Weak Form

Introducing the test functions $\mathbf{w}$, integrating by parts and using the divergence theorem, the weak form of the equilibrium equation in the elastic domain can be written as: *find* $\mathbf{v} \in \mathscr{V}$ *for all* $\mathbf{w} \in \mathscr{V}^0$ *such that,*

$$-(\nabla\mathbf{w}, \boldsymbol{\sigma})_{\Omega^{\mathrm{s}}} - (\mathbf{w}, \bar{\mathbf{t}})_{\Gamma^{\mathrm{N}}_{\mathbf{v}}} + (\mathbf{w}, \rho\mathbf{f})_{\Omega^{\mathrm{s}}} \;\; = \;\; 0, \tag{2.19}$$

where the notation $(\cdot,\cdot)_{\Omega^{\mathrm{s}}}$ indicates the L^2 inner product with respect to the domain Ω^{s}, and $\mathscr{V}$ and $\mathscr{V}^0$ are spaces of sufficiently smooth functions for the continuous fields and their variations. By construction, we also require that the test function $\mathbf{w}$ vanishes on the Dirichlet boundaries. We do not write the weak form for the transport equations (related to $\hat{\mathbf{F}}$ and J) because we will use an explicit scheme to update of the variables for each Lagrangian particle, individually, as described in section 3.5.

CHAPTER 3

Solution Strategy for CEL formulation

We now present a novel numerical strategy that couples the Eulerian and updated Lagrangian formulations with the objective of evolving the moving interface on a fixed Eulerian grid. For this the solid boundary Γ is represented by an evolving level set function across which a discontinuity in velocity and deformation is described using the extended finite element method. The interface describing the deformed shape of the elastic body is moved in an incremental manner until equilibrium. For any increment, the method consists in computing the velocity field $\mathbf{v}$ in the current domain $\Omega^s(t_n)$ by solving the equilibrium equation (2.19). The field variables $\hat{\mathbf{F}}$ and J are then updated point-wise by using an explicit updated Lagrangian mapping algorithm between the material configurations at the current and next increments (denoted by pseudo-time steps $t_n = t$ and $t_{n+1} = t + dt$, respectively). Between these increments the interface is moved using the particle-based moving interface method. In the following sections, a more detailed description of the methodology is presented.

3.1 Level set representation of solid interface

Mathematically, we here represent the boundary $\Gamma(t)$ of the solid domain with the level set function ϕ defined as,

$$\Gamma = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x},t) = 0\}, \tag{3.1}$$

so that $\phi < 0$ inside the solid domain $\Omega^s(t)$ and $\phi > 0$ outside the solid domain $\Omega \setminus \Omega^s(t)$. Although there are several choices for ϕ, for stability, we choose it to be the signed distance function defined by:

$$\phi(\mathbf{x},t) = \pm \min_{\mathbf{x}' \in \Gamma} ||\mathbf{x} - \mathbf{x}'|| \quad \text{for all} \quad \mathbf{x} \in \Omega. \tag{3.2}$$

Practically, the function ϕ can be reinitialized at every increment using the locations of the interface in order to maintain the properties of the signed distance function. When the interface is described by particles, the reinitialization procedure is discussed in [3]. A more detailed description of the implementation of the interface evolution algorithm will be given later in this section 3.4.

Remark 4 *We note here that the level set method (LSM) [32] may also be used for evolving the interface in time. However, the GPM [3] employed here provides an attractive solution to explicitly track Lagrangian particles on the solid boundary instead of resorting to solving a level set evolution equation.*

3.2 Extended finite element approximation

Following an Eulerian approach, a fixed and structured finite element discretization is introduced for the entire physical domain Ω (including the space that does not belong to the solid body). The boundary of the body Ω^{s} is then defined with a multi-segment closed surface Γ that cuts through some of the elements. Since the field variables, namely, $\mathbf{v}$, $\hat{\mathbf{F}}$ and J have non-zero values in Ω^{s} but identically vanish in $\Omega \setminus \Omega^{\mathrm{s}}(t)$, field discontinuities naturally occur across Γ. To handle this issue, we employ the extended finite element method (XFEM), as it is able to capture the presence of discontinuities within elements and thus render the discretization of the interface Γ easy and computationally efficient. We adopt a mixed formulation wherein the velocity field $\mathbf{v}$ is interpolated with nine-node (biquadratic) element shape functions and the isochoric part of the deformation gradient $\hat{\mathbf{F}}$ and its Jacobian J are interpolated with four-node (bilinear) element shape functions (see Figure 3.1). This reads:

$$v_i(\mathbf{x},t) = \sum_{I=1}^{9} N^I(\mathbf{x}) v_i^I(t) + \sum_{I=1}^{9} N^I(\mathbf{x}) \mathscr{S}^I(\mathbf{x},t) a_i^I(t), \tag{3.3}$$

$$\hat{F}_{ij}(\mathbf{x},t) = \sum_{I=1}^{4} \hat{N}^I(\mathbf{x}) \hat{F}_{ij}^I(t) + \sum_{I=1}^{4} \hat{N}^I(\mathbf{x}) \mathscr{S}^I(\mathbf{x},t) D_{ij}^I(t), \tag{3.4}$$

$$J(\mathbf{x},t) = \sum_{I=1}^{4} \hat{N}^I(\mathbf{x}) J^I(t) + \sum_{I=1}^{4} \hat{N}^I(\mathbf{x}) \mathscr{S}^I(\mathbf{x},t) C^I(t), \tag{3.5}$$

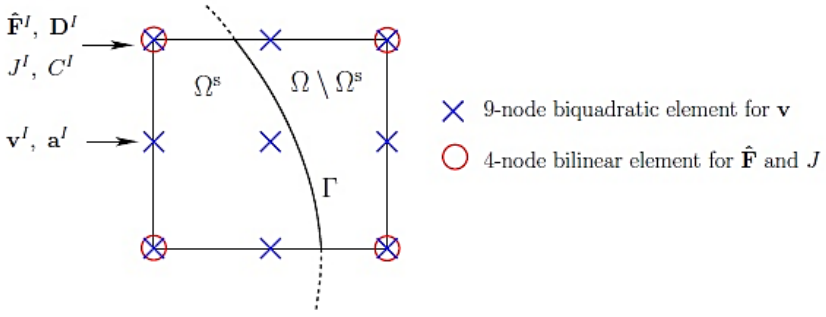
where N^I and $\hat{N}^I$ denote the 9-node element and the 4-node element Lagrange shape functions, respectively; the superscript index I is used for node numbering and the subscript indices i, j are used for numbering the Cartesian components; $\mathbf{v}^I$, $\hat{\mathbf{F}}^I$, J^I denote the standard degrees of freedom (DOFs) and $\mathbf{a}^I$, $\mathbf{D}^I$, C^I denote the corresponding enriched DOFs at node I, respectively; the step enrichment function $\mathscr{S}^I$ at enriched node I used to incorporate the jump discontinuity in the fields is defined as,

$$\mathscr{S}^I = H\left(\phi(\mathbf{x},t)\right) - H\left(\phi(\mathbf{x}^I,t)\right) \tag{3.6}$$

and the Heaviside function H is defined as,

$$H(\phi(\mathbf{x},t)) = \begin{cases} 1 & \phi > 0, \\ 0 & \phi < 0. \end{cases} \tag{3.7}$$

Note that the level set function ϕ is continuous across the interface and so that it can be interpolated using the shape functions N^I and $\hat{N}^I$.



(a)

Figure 3.1: Illustration of the mixed extended finite element and the location of the degrees of freedom. Circles (○) show the location of bilinear element nodes and crosses (×) show the location of biquadratic element nodes. The interface cutting through the element is represented implicitly using the level set function ϕ.

Remark 5 *Previously, Duddu et al. [2] proposed the above mixed formulation to ensure stability in the case of nearly incompressible elastic solids (e.g. rubber with Poisson's ratio* $\nu = 0.48 - 0.5$. *However, even for a compressible solid, the mixed formulation results in better accuracy and hence requires less number of iterations to reach the tolerance limit for the residual.*

In this study, we reduce the dimension of the domain by considering that it is uniform in the x_3 direction (plane strain conditions apply). This implies that $v_3(\mathbf{x},t) = 0$, $F_{33}(\mathbf{x},t) = 1$, $F_{13}(\mathbf{x},t) = F_{32}(\mathbf{x},t) = 0$; this allows us to not consider them as nodal degrees of freedom (DOFs) in our analysis. Moreover, for clarity, we write the linear system in the following matrix form:

$$\begin{cases} \mathbf{v}(\mathbf{x},t) = \bar{\mathbf{N}}_{\mathbf{v}}(\mathbf{x},t)\bar{\mathbf{v}}(t), \\ \hat{\mathbf{F}}(\mathbf{x},t) = \bar{\mathbf{N}}_{\hat{\mathbf{F}}}(\mathbf{x},t)\bar{\mathbf{F}}(t), \\ J(\mathbf{x},t) = \bar{\mathbf{N}}_{J}(\mathbf{x},t)\bar{J}(t). \end{cases} \tag{3.8}$$

Here, the shape function matrices $\bar{\mathbf{N}}_{\mathbf{v}}$, $\bar{\mathbf{N}}_{\hat{\mathbf{F}}}$ and $\bar{\mathbf{N}}_{J}$ and element vectors $\bar{\mathbf{v}}$, $\bar{\mathbf{F}}$ and $\bar{J}$ contain both standard and enriched DOFs and are defined as:

$$\text{nodal DOFs}\begin{cases}\bar{\mathbf{v}} = \left[\bar{\mathbf{v}}^{\text{reg}}; \bar{\mathbf{v}}^{\text{enr}}\right]_{36\times 1}, \\ \bar{\mathbf{v}}^{\text{reg}} = \left[v_1^1,\ v_2^1,\ \ldots\ v_1^9,\ v_2^9\right]^T_{18\times 1}; \bar{\mathbf{v}}^{\text{enr}} = \left[a_1^1,\ a_2^1,\ \ldots\ a_1^9,\ a_2^9\right]^T_{18\times 1} \\ \bar{\mathbf{F}} = \left[\bar{\mathbf{F}}^{\text{reg}}; \bar{\mathbf{F}}^{\text{enr}}\right]_{32\times 1}, \\ \bar{\mathbf{F}}^{\text{reg}} = \left[\hat{F}_{11}^1,\ \hat{F}_{22}^1,\ \hat{F}_{12}^1,\ \hat{F}_{21}^1,\ \ldots\ \hat{F}_{11}^4,\ \hat{F}_{22}^4,\ \hat{F}_{12}^4,\ \hat{F}_{21}^4\right]^T_{16\times 1}, \\ \bar{\mathbf{F}}^{\text{enr}} = \left[D_{11}^1,\ D_{22}^1,\ D_{12}^1,\ D_{21}^1,\ \ldots\ D_{11}^4,\ D_{22}^4,\ D_{12}^4,\ D_{21}^4\right]^T_{16\times 1} \\ \bar{J} = \left[\bar{J}^{\text{reg}}; \bar{J}^{\text{enr}}\right]_{8\times 1}, \\ \bar{J}^{\text{reg}} = \left[J^1, \ldots J^4\right]^T_{4\times 1}; \bar{J}^{\text{enr}} = \left[C^1, \ldots C^4\right]^T_{4\times 1}, \end{cases} \tag{3.9}$$

$$\text{shape functions}\begin{cases}\bar{\mathbf{N}}_{\mathbf{v}} = \left[\ \mathbf{N}_{\mathbf{v}}^{\text{reg}},\ \mathbf{N}_{\mathbf{v}}^{\text{enr}}\ \right]_{2\times 36} \\ \bar{\mathbf{N}}_{\hat{\mathbf{F}}} = \left[\ \mathbf{N}_{\mathbf{F}}^{\text{reg}},\ \mathbf{N}_{\mathbf{F}}^{\text{enr}}\ \right]_{4\times 32} \\ \bar{\mathbf{N}}_{J} = \left[\ \mathbf{N}_{J}^{\text{reg}},\ \mathbf{N}_{J}^{\text{enr}}\ \right]_{1\times 8}\end{cases} \tag{3.10}$$

with

$$\mathbf{N}_{\mathbf{v}}^{\text{reg}} = \left[\ \mathbf{N}_{\mathbf{v}}^1, \ldots, \mathbf{N}_{\mathbf{v}}^9\ \right]_{2\times 18}, \mathbf{N}_{\mathbf{v}}^{\text{enr}} = \left[\ \mathscr{S}^1\mathbf{N}_{\mathbf{v}}^1, \ldots, \mathscr{S}^9\mathbf{N}_{\mathbf{v}}^9\ \right]_{2\times 18}$$
$$\mathbf{N}_{\mathbf{F}}^{\text{reg}} = \left[\ \mathbf{N}_{\mathbf{F}}^1, \ldots, \mathbf{N}_{\mathbf{F}}^4\ \right]_{4\times 16}, \mathbf{N}_{\mathbf{F}}^{\text{enr}} = \left[\ \mathscr{S}^1\mathbf{N}_{\mathbf{F}}^1, \ldots, \mathscr{S}^4\mathbf{N}_{\mathbf{F}}^4\ \right]_{4\times 16}$$
$$\mathbf{N}_{J}^{\text{reg}} = \left[\ \hat{N}^1, \ldots, \hat{N}^4\ \right]_{1\times 4}, \mathbf{N}_{J}^{\text{enr}} = \left[\ \mathscr{S}^1\hat{N}^1, \ldots, \mathscr{S}^4\hat{N}^4\ \right]_{1\times 4}$$

and $\mathbf{N}_{\mathbf{v}}^I = \begin{bmatrix} N^I & 0 \\ 0 & N^I \end{bmatrix}$, $\mathbf{N}_{\mathbf{F}}^I = \begin{bmatrix} \hat{N}^I & 0 & 0 & 0 \\ 0 & \hat{N}^I & 0 & 0 \\ 0 & 0 & \hat{N}^I & 0 \\ 0 & 0 & 0 & \hat{N}^I \end{bmatrix}$.

3.3 Linearization of equilibrium equation

The discretized form of the equilibrium equation can be written from the weak form (2.19) by introducing the XFEM approximation. Neglecting body forces, the element residual vector is given by,

$$\mathbf{R}_{\mathbf{v}} \;=\; -\int_{\Omega^{\mathrm{e}}} \mathbf{B}_{\mathbf{v}}^{T}\,\tilde{\boldsymbol{\sigma}}\;\mathrm{d}\Omega - \int_{\Gamma_{\mathbf{v}}^{e}} \bar{\mathbf{N}}_{\mathbf{v}}^{T}\,\bar{\mathbf{t}}\;\mathrm{d}\Gamma, \tag{3.11}$$

where Ω^{e} is the part of the domain Ω^{s} contained in the finite element e, $\Gamma_{\mathbf{v}}^{e}$ is the segment of the Neumann boundary $\Gamma_{\mathbf{v}}^{\mathrm{N}}$ intersecting the element e. Furthermore, $\tilde{\boldsymbol{\sigma}} = [\sigma_{11},\ \sigma_{22},\ \sigma_{21},\ \sigma_{12}]_{4\times 1}^{T}$ is the symmetric Cauchy stress matrix, $\tilde{\mathbf{t}} = [\bar{t}_1,\ \bar{t}_2]_{2\times 1}^{T}$ is the surface traction vector and the gradient matrix of the 9-node element shape functions $\mathbf{B}_{\mathbf{v}}$ is given by:

$$\mathbf{B}_{\mathbf{v}} = \left[\mathbf{B}_{\mathbf{v}}^{1},\ ...,\ \mathbf{B}_{\mathbf{v}}^{9},\ \ \mathscr{S}^{1}\mathbf{B}_{\mathbf{v}}^{1},\ ...,\ \mathscr{S}^{9}\mathbf{B}_{\mathbf{v}}^{9}\right]_{4\times 36} \ \text{ with } \ \mathbf{B}_{\mathbf{v}}^{I} = \begin{bmatrix} \frac{\partial N^{I}}{x_1} & 0 \\ 0 & \frac{\partial N^{I}}{x_2} \\ \frac{\partial N^{I}}{x_2} & 0 \\ 0 & \frac{\partial N^{I}}{x_1} \end{bmatrix}_{4\times 2} \tag{3.12}$$

Although, due to symmetry it is sufficient to represent the stress tensor $\tilde{\boldsymbol{\sigma}}$ as a 3×1 array in Voight notation, for the purpose of matrix operations related to the divergence of stress in the above residual we prefer to use 4×1 array. Using a Taylor's expansion, we linearize equation (3.11) at time t_{n+1} as [1] and obtain:

$$\mathbf{0} = \mathbf{R}_{\mathbf{v}}\left(t_{n+1}\right) = \mathbf{R}_{\mathbf{v}}\left(t_{n}\right) + \left[\frac{\partial \mathbf{R}_{\mathbf{v}}}{\partial t}\right]\,\Delta t, \tag{3.13}$$

which yields:

$$\left[\frac{\partial \mathbf{R}_{\mathbf{v}}}{\partial t}\right] = -\frac{1}{\Delta t}\,\mathbf{R}_{\mathbf{v}}\left(t_{n}\right), \tag{3.14}$$

Using the transport equations (2.7) and (2.8) together with the constitutive relation (2.16), the above equation can be written in the form of a linear system as [2]:

$$\mathbf{K}_{\mathbf{v}}(t_n)\bar{\mathbf{v}}(t_{n+1}) = -\frac{1}{\Delta t}\,\mathbf{R}_{\mathbf{v}}(t_n), \tag{3.15}$$

where $\mathbf{K}_{\mathbf{v}}$ is the consistent tangent stiffness matrix. The discretization of the domain integrals using the extended finite element approximation for obtaining the tangent matrix $\mathbf{K}_{\mathbf{v}}$ is described below.

In equation (3.15), the tangent matrix corresponding to the linear system obtained by discretizing the equilibrium equation is given by,

$$\begin{aligned}
\mathbf{K}_{\mathbf{v}} &= \int_{\Omega^e} \left[\mathbf{B}_{\mathbf{v}}^T \frac{\partial \boldsymbol{\sigma}}{\partial \hat{\mathbf{F}}} \left(-\nabla\tilde{\hat{\mathbf{F}}}\bar{\mathbf{N}}_{\mathbf{v}} + \tilde{\hat{\mathbf{F}}}\mathbf{B}_{\mathbf{v}} - \frac{1}{3}\tilde{\hat{\mathbf{F}}}\check{\mathbf{B}}_{\mathbf{v}} \right) \right. \\
&+ \left. \mathbf{B}_{\mathbf{v}}^T \frac{\partial \boldsymbol{\sigma}}{\partial J} \left(-\nabla\tilde{J}^T\bar{\mathbf{N}}_{\mathbf{v}} + \tilde{J}\check{\mathbf{B}}_{\mathbf{v}} \right) \right] \mathrm{d}\Omega
\end{aligned} \tag{3.16}$$

In the above equation,

$$\begin{aligned}
\frac{\partial \boldsymbol{\sigma}}{\partial J} &= \frac{1}{J}\left[\frac{\kappa}{J}\{1 \;\; 0 \;\; 0 \;\; 0\}^T - \tilde{\boldsymbol{\sigma}} \right]_{4\times 1} \\
\frac{\partial \boldsymbol{\sigma}}{\partial \hat{\mathbf{F}}} &= \begin{bmatrix} \delta\sigma_{1111} & \delta\sigma_{1122} & \delta\sigma_{1112} & \delta\sigma_{1121} \\ \delta\sigma_{2211} & \delta\sigma_{2222} & \delta\sigma_{2212} & \delta\sigma_{2221} \\ \delta\sigma_{1211} & \delta\sigma_{1222} & \delta\sigma_{1212} & \delta\sigma_{1221} \\ \delta\sigma_{2111} & \delta\sigma_{2122} & \delta\sigma_{2112} & \delta\sigma_{2121} \end{bmatrix}_{4\times 4} \\
\delta\sigma_{ijlm} &= \frac{\mu}{J}\left(\delta_{li}\hat{F}_{jm} + \delta_{lj}\hat{F}_{im} - \frac{2}{3}\delta_{ij}\hat{F}_{lm} \right).
\end{aligned}$$

The tilde superscript $\tilde{(\cdot)}$ indicates that $\nabla\tilde{\hat{\mathbf{F}}}$, $\tilde{\hat{\mathbf{F}}}$ and $\tilde{\hat{J}}$ are interpolated using the fields from

the previous time step $\hat{\mathbf{F}}^t$ and J^t as follows:

$$\begin{aligned}
\nabla\tilde{\bar{\mathbf{F}}} &= \left[\mathbf{B}_{\mathbf{F}1}\bar{\mathbf{F}}(t)\ \ \mathbf{B}_{\mathbf{F}2}\bar{\mathbf{F}}(t)\right]_{4\times 2} \\
\nabla\tilde{J} &= \mathbf{B}_J\bar{J}(t) \\
\tilde{\bar{\mathbf{F}}} &= \left[\bar{\mathbf{N}}_{\mathbf{F}1}\bar{\mathbf{F}}(t)\ \ \bar{\mathbf{N}}_{\mathbf{F}2}\bar{\mathbf{F}}(t)\ \ \bar{\mathbf{N}}_{\mathbf{F}3}\bar{\mathbf{F}}(t)\ \ \bar{\mathbf{N}}_{\mathbf{F}4}\bar{\mathbf{F}}(t)\right] \\
\tilde{J} &= \bar{\mathbf{N}}_J\bar{J}(t)
\end{aligned}$$

and the matrices $\mathbf{B}_{\mathbf{F}k}$, $\mathbf{B}_J$, $\mathbf{N}_{\mathbf{F}l}$ and $\check{\mathbf{B}}_{\mathbf{v}}$ are written, for $k = 1,2$ and $l = 1,2,3,4$:

$$\begin{aligned}
\mathbf{B}_{\mathbf{F}k} &= \left[\mathbf{B}^1_{\mathbf{F}k},\ ...,\ \mathbf{B}^4_{\mathbf{F}k},\ \mathscr{S}^1\mathbf{B}^1_{\mathbf{F}k},\ ...,\ \mathscr{S}^4\mathbf{B}^4_{\mathbf{F}k}\right]_{4\times 32} \\
\hat{\mathbf{B}}_J &= \left[\mathbf{B}^1_J,\ ...,\ \mathbf{B}^4_J,\ \mathscr{S}^1\mathbf{B}^1_J,\ ...,\ \mathscr{S}^4\mathbf{B}^4_J\right]_{2\times 8} \\
\mathbf{N}_{\mathbf{F}l} &= \left[\mathbf{N}^1_{\mathbf{F}l},\ ...,\ \mathbf{N}^4_{\mathbf{F}l},\ \mathscr{S}^1\mathbf{N}^1_{\mathbf{F}k},\ ...,\ \mathscr{S}^4\mathbf{N}^4_{\mathbf{F}l}\right]_{4\times 32} \\
\check{\mathbf{B}}_{\mathbf{v}} &= \left[\check{\mathbf{B}}^1_{\mathbf{v}},\ ...,\ \check{\mathbf{B}}^9_{\mathbf{v}},\ \mathscr{S}^1\check{\mathbf{B}}^1_{\mathbf{v}},\ ...,\ \mathscr{S}^9\check{\mathbf{B}}^9_{\mathbf{v}}\right]_{1\times 36}
\end{aligned}$$

with

$$
\begin{aligned}
\mathbf{B}^I_{\mathbf{F}1} &= \begin{bmatrix} \frac{\partial \hat{N}^I}{\partial x_1} & 0 & 0 & 0 \\ 0 & \frac{\partial \hat{N}^I}{\partial x_1} & 0 & 0 \\ 0 & 0 & \frac{\partial \hat{N}^I}{\partial x_1} & 0 \\ 0 & 0 & 0 & \frac{\partial \hat{N}^I}{\partial x_1} \end{bmatrix}_{4\times 4} \\
\mathbf{B}^I_{\mathbf{F}2} &= \begin{bmatrix} \frac{\partial \hat{N}^I}{\partial x_2} & 0 & 0 & 0 \\ 0 & \frac{\partial \hat{N}^I}{\partial x_2} & 0 & 0 \\ 0 & 0 & \frac{\partial \hat{N}^I}{\partial x_2} & 0 \\ 0 & 0 & 0 & \frac{\partial \hat{N}^I}{\partial x_2} \end{bmatrix}_{4\times 4} \\
\mathbf{B}^I_J &= \begin{bmatrix} \frac{\partial \hat{N}^I}{\partial x_1} & \frac{\partial \hat{N}^I}{\partial x_2} \end{bmatrix}_{2\times 1} \\
\check{\mathbf{B}}^1_{\mathbf{v}} &= \begin{bmatrix} \frac{\partial N^I}{x_1} & \frac{\partial N^I}{x_2} \end{bmatrix}_{1\times 2}.
\end{aligned}
$$

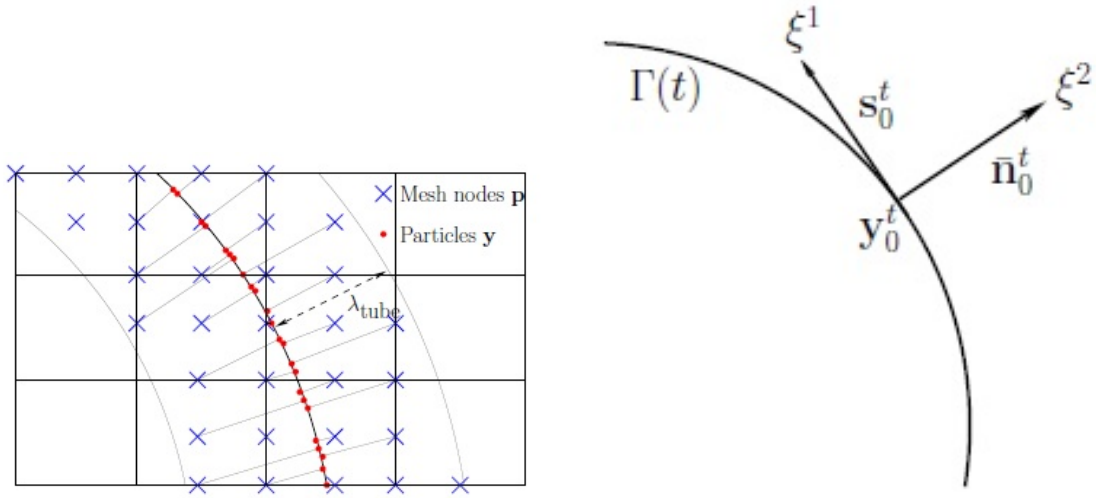
$$
\mathbf{N}_{\mathbf{F}1} = \begin{bmatrix} \hat{N}^I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{N}^I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{4\times 4} \qquad \mathbf{N}_{\mathbf{F}2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \hat{N}^I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{N}^I \end{bmatrix}_{4\times 4}
$$

$$
\mathbf{N}_{\mathbf{F}3} = \begin{bmatrix} 0 & 0 & \hat{N}^I & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{N}^I & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{4\times 4} \qquad \mathbf{N}_{\mathbf{F}4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{N}^I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{N}^I \end{bmatrix}_{4\times 4}
$$

3.4 Grid based particle method

To track the deformation of the interface Γ, we propose to use a grid based particle method similar to what was introduced in [3]. This method indeed possesses the double advantage

of tracking the interface explicitly with particles while using the underlying fixed finite element mesh; this ensures a fairly uniform repartition of the particles throughout the interface. Herein, we review the basic idea behind the particle based moving interface method and discuss the procedure to update of the interface position and deformations measures within the current numerical scheme. The interface particles on Γ, whose position is de-



(a) (b)

Figure 3.2: Grid particle scheme . Figure (a) shows the one to one correspondence between the particles and the nodes inside the computational tube, while (b) shows the local basis, centred on the particle $\mathbf{y}_0$ closest to the node considered $\mathbf{p}$.

noted by the vector $\mathbf{y}$, are chosen as the normal projection of the underlying mesh nodes with position vector $\mathbf{p}$. Since the interface is initially described implicitly as the zero level set of a signed distance function $\phi(\mathbf{p},0)$ at initial time $t = t_0$, its value gives the perpendicular distance between mesh point and interface point. Considering that the gradient of level set function gives the local interface normal, the initial coordinates of particles $\mathbf{y}$ are given by [3]:

$$\mathbf{y} \;=\; \mathbf{p} - \phi(\mathbf{p},0)\mathbf{n} = \mathbf{p} - \phi(\mathbf{p},0)\nabla\phi(\mathbf{p},0) \tag{3.17}$$

Note that the above expression is only valid when $\|\nabla\phi\| = 1$, that is, when ϕ is a signed distance function. To limit the number of particles, we define a so-called computational tube such that only nodes $\mathbf{p}$, whose distance to Γ is smaller than a cut-off value λ_{tube}, are taken into account (see Fig. 3.2(a)).

Remark 6 *It is important to note here that there is a one to one correspondence between each particle* $\mathbf{y}$ *and node* $\mathbf{p}$*, thus, providing each interface particle an Eulerian reference mesh point. This ensures a quasi-uniform repartition of particles along the interface throughout its evolution and avoids the need for node point redistribution schemes, unlike the standard marker particle methods [33]*

Between two subsequent time steps $t_n = t$ and $t_{n+1} = t + dt$, the particles are moved with the normal interface velocity $\mathbf{v}^{\perp} = (\mathbf{v}\cdot\mathbf{n})\mathbf{n}$ using a second order Runge-Kutta time integration procedure as follows:

$$\mathbf{y}^{t+dt/2} = \mathbf{y}^t + \mathbf{v}^{\perp}(\mathbf{y}^t,t)\frac{dt}{2} + \boldsymbol{\Omega}\cdot\mathbf{v}^{\perp}(\mathbf{y}^t,t)\frac{dt^2}{4} \tag{3.18}$$

$$\mathbf{y}^{t+dt} = \mathbf{y}^t + \mathbf{v}^{\perp}(\mathbf{y}^{t+dt/2},t)dt + \boldsymbol{\Omega}\cdot\mathbf{v}^{\perp}(\mathbf{y}^{t+dt/2},t)\frac{dt^2}{2}, \tag{3.19}$$

where $\boldsymbol{\Omega}$ is the matrix of the angular velocity of the interface normal. Introducing the local coordinates ξ_1 and ξ_2 that respectively run in the directions tangent and normal to the interface at point $\mathbf{y}^t$, the angular velocity can be written as,

$$\boldsymbol{\omega} = -\left(\mathbf{v}^{\perp}\cdot\mathbf{n}\right)_{,\xi^1}\mathbf{z} \quad \text{and} \quad \Omega_{ik} = \varepsilon_{ijk}\omega_j \tag{3.20}$$

with the permutation tensor $\varepsilon_{ijk} = \frac{1}{2}(i-j)(j-k)(k-i)$, indices $i,j,k = \{1,2.3\}$ and the normal vector out of plane $\mathbf{z} = [0\,0\,1]^T$. The term $(\mathbf{v}^{\perp}\cdot\mathbf{n})_{,\xi^1}$ indicates the derivative of the magnitude of the normal velocity with respect to the coordinate ξ^1. The relationship between the local and global coordinates ξ^1 and $\mathbf{y}$ is given bellow.

After the position of the interface has been updated, the particle distribution on Γ may become uneven, and this can affect the geometrical resolution of the interface. To overcome this issue, the interface is resampled after motion by recomputing the particles as the closest points on Γ to the nodes $\mathbf{p}$ inside the updated computational tube (which has moved with the interface). This is done by first approximating the interface with polynomials locally around each particle. The procedure, explained here in the two dimensional case, is as follows: for each node $\mathbf{p}$ inside the computational tube, the closest m particles $\mathbf{y}_0^t ... \mathbf{y}_m^t$ are collected at time t, carrying with them the tangent $\mathbf{s}_0^t ... \mathbf{s}_m^t$ and normal $\bar{\mathbf{n}}_0^t ... \bar{\mathbf{n}}_m^t$ to the interface before motion. Denoting $\mathbf{y}_0^t$ as the particle closest to $\mathbf{p}$, a polynomial of degree $n < m$ is fitted to the particles $\mathbf{y}_0^t ... \mathbf{y}_m^t$ in the local coordinate system $\{\mathbf{s}_0^t; \bar{\mathbf{n}}_0^t\}$ centered on $\mathbf{y}_0^t$. The location $\tilde{\mathbf{y}}_i^t$ of particle i in this local coordinate system is given by:

$$\tilde{\mathbf{y}}_i \;=\; \left\{ \begin{array}{c} \xi_i^1 \\ \xi_i^2 \end{array} \right\} \;=\; \mathbf{R}^t \cdot (\mathbf{y}_i^t - \mathbf{y}_0^t) \quad \text{with} \quad \mathbf{R}^t \;=\; \left[\begin{array}{c} (\mathbf{s}_0^t)^T \\ (\bar{\mathbf{n}}_0^t)^T \end{array} \right]. \tag{3.21}$$

Taking the example of a quadratic polynomial ($n = 2$), the interface around particle $\mathbf{y}_0$ is represented in the local referential as the graph function $\xi^2(\xi^1) = c_0 + c_1\xi^1 + c_2(\xi^1)^2$, where the coefficients c_0, c_1 and c_2 are found by minimizing the L^2 difference between the $\xi^2(\xi_i^1)$ and the ξ_i^2. The coordinates $\left\{\xi^1, \xi^2(\xi^1)\right\}$ define a local parameterization $\mathbf{r}^l(\xi^1)$ of Γ in the neighbourhood of $\mathbf{y}_0^t$ (Fig. 3.2(b)):

$$\mathbf{r}^l(\xi^1) \;=\; \left\{ \begin{array}{c} \xi^1 \\ \xi^2(\xi^1) \end{array} \right\}. \tag{3.22}$$

The relationship between the local parameterization $\mathbf{r}^l(\xi^1, \xi^2)$ and the global parameterization of the interface $\mathbf{r}(\xi^1, \xi^2)$ is then found via rotation and translation operations in the

form:

$$\mathbf{r}(\xi^1,\xi^2,t+dt) = (\mathbf{R}^t)^{-1}\,\mathbf{r}^l(\xi^1,\xi^2)+\mathbf{y}_0^t \tag{3.23}$$

$$\text{with } \mathbf{R}^t = [\mathbf{s}_0^t\ \bar{\mathbf{n}}_0^t]^T, \tag{3.24}$$

where $\mathbf{R}^t$ is the rotation matrix from the local basis $\{\mathbf{s}_0^t;\bar{\mathbf{n}}_0^t\}$ to the global basis $\{\mathbf{e}_1;\bar{\mathbf{e}}_2\}$. The parameterization $\mathbf{r}(\xi^1,t+dt)$ can now be used to resample the interface, i.e. recalculate the closest point on the interface to the nodes $\mathbf{p}$. This is done by minimizing the distance function $d(\mathbf{r}(\xi^1,t+dt);\mathbf{p}) = 1/2\left|\mathbf{r}(\xi^1,t+dt)-\mathbf{p}\right|$ with respect to ξ^1. In two dimensions, the solution can be found explicitly by solving a cubic equation. Other geometrical quantities can also be found using the parameterization $\mathbf{r}(\xi^1,t+dt)$, such as the updated basis $\left\{\mathbf{s}^{t+dt},\bar{\mathbf{n}}^{t+dt}\right\}$:

$$\mathbf{s}^{t+dt} = \mathbf{r}(\xi^1,t+dt)_{,1} = \mathbf{R}^t\,\frac{\partial \mathbf{r}^l(\xi^1,t+dt)}{\partial \xi^1} \tag{3.25}$$

$$\bar{\mathbf{n}}^{t+dt} = \mathbf{s}^{t+dt}\times\mathbf{z}/|\mathbf{s}^{t+dt}\times\mathbf{z}|. \tag{3.26}$$

Finally, a new level-set function $\phi(\mathbf{p},t+dt)$ can be calculated as the signed distance function to Γ at nodes $\mathbf{p}$ as follows [3]:

$$\phi(\mathbf{p},t+dt) = -\text{sgn}\left(\frac{\mathbf{y}^{t+dt}-\mathbf{p}}{|\mathbf{y}^{t+dt}-\mathbf{p}|}\cdot\bar{\mathbf{n}}_0^t\right)|\mathbf{y}^{t+dt}-\mathbf{p}|, \tag{3.27}$$

where $\mathbf{y}^{t+dt}$ is the particle associated with $\mathbf{p}$ at time $t+dt$ and the "sgn" is the sign or signum function. The reconstruction of the level set function using the local polynomial approximation of the interface is computationally inexpensive, and is used in the XFEM part of the algorithm. Let us summarize the GPM scheme in a pseudo algorithm as follows:

1. Given the initial level set function ϕ, find the coordinates of the particles that corresponds to the nodes inside the computational tube (initialization step).

2. Given the velocity field $\mathbf{v}^t$, update the position of the particle $\mathbf{y}^t$ to its current position $\mathbf{y}^{t+dt}$.

3. For each particle $\mathbf{y}_0$, find the neighbouring particles to construct a local polynomial interpolation $\mathbf{r}(\xi^1, t+dt)$ of the surface Γ around $\mathbf{y}_0$.

4. Given $\mathbf{r}(\xi^1, t+dt)$, find the new particles by projecting the nodes inside the computational tube on the surface Γ.

5. Compute the new geometrical quantities such as the normal $\bar{\mathbf{n}}^{t+dt}$ and the level set function ϕ^{t+dt}

3.5 Lagrange transport of deformation gradient and Jacobian

As the elastic body deforms, the material flows through the mesh, allowing finite element nodes to come in and out of the domain Ω^s. Because the interface $\Gamma(t)$ describing the solid domain is moving, the transport of $\hat{\mathbf{F}}$ and J cannot be described with a simple convection term, unlike when the domain boundaries are fixed [2]. To overcome this issue, herein we propose to transport deformation quantities by employing an updated Lagrangian description. It this scheme, first, the regular degrees of freedom $\bar{J}^{\text{reg}}$ and $\bar{\mathbf{F}}^{\text{reg}}$ are updated as follows:

1. Given the velocity field $\mathbf{v}^t$ computed with (3.15) and using the GPM, the domain $\Omega^s(t)$ and the interface $\Gamma(t)$ are updated to their new position $\Omega^s(t+dt)$ and $\Gamma(t+dt)$.

2. For each node i inside $\Omega^s(t+dt)$, material particle at its spatial location $\mathbf{x}_i^{t+dt}$ is backtracked to its position $\mathbf{x}_i^t$ at time t in accordance with the velocity field $\mathbf{v}^t$.

3. The fields J_i^t and $\hat{\mathbf{F}}_i^t$ known at time t are interpolated at point $\mathbf{x}_i^t$ in $\Omega^s(t)$ using the extended finite element approximation.

4. Using the transport equations (2.7) and (2.8), J_i^{t+dt} and $\hat{\mathbf{F}}_i^{t+dt}$ at point $\mathbf{x}_i^{t+dt}$ are computed as:

$$J_i^{t+dt} = J_i^t(1+\nabla\cdot\mathbf{v}^t(\mathbf{x}_i^t)dt), \tag{3.28}$$

$$\hat{\mathbf{F}}_i^{t+dt} = \hat{\mathbf{F}}_i^t(\mathbf{I}+\nabla\mathbf{v}^t(\mathbf{x}_i^t)dt-\frac{1}{3}\nabla\cdot\mathbf{v}^t(\mathbf{x}_i^t)\mathbf{I}dt), \tag{3.29}$$

and assigned to the new regular degrees of freedom $\bar{J}_i^{\text{reg}} = J_i^{t+dt}$ and $\bar{\mathbf{F}}_i^{\text{reg}} = \hat{\mathbf{F}}_i^{t+dt}$.

At the end of step 4, the regular degrees of freedom $\bar{J}^{\text{reg}}$ and $\bar{\mathbf{F}}^{\text{reg}}$ have been updated at each node inside the new domain $\Omega^s(t+dt)$. However, since the interface has moved, the intersection between Γ and the underlying mesh has changed and the enriched degrees of freedom $\bar{J}^{\text{enr}}$ and $\bar{\mathbf{F}}^{\text{enr}}$ have to be updated as well. This is done by solving the following equations in the elements cut by Γ:

$$J^{t+dt}-\tilde{J} = 0 \;\forall\, \mathbf{x}\in\Omega^\Gamma, \tag{3.30}$$

$$\hat{\mathbf{F}}^{t+dt}-\tilde{\mathbf{F}} = 0 \;\forall\, \mathbf{x}\in\Omega^\Gamma, \tag{3.31}$$

where Ω^Γ is the ensemble of the elements Ω_e that are cut by Γ. The terms $\tilde{J}$ and $\tilde{\mathbf{F}}$ are the updated values of the fields, which can be calculated at any points $\mathbf{x}$ inside element cuts by Γ using equations (3.28) and (3.29). The weak form of the above equations read,

$$\Big(w_J,\ (J^{t+dt}-\tilde{J})\Big)_{\Omega^\Gamma} = 0, \tag{3.32}$$

$$\Big(\mathbf{w}_F,\ (\hat{\mathbf{F}}^{t+dt}-\tilde{\mathbf{F}})\Big)_{\Omega^\Gamma} = 0, \tag{3.33}$$

and the corresponding discretized forms of are given by,

$$\mathbf{K}_J^{\text{enr}}\bar{\mathbf{J}}_g^{\text{enr}} = \mathbf{R}_J^{\text{enr}}, \tag{3.34}$$

$$\mathbf{K}_F^{\text{enr}}\bar{\mathbf{F}}_g^{\text{enr}} = \mathbf{R}_F^{\text{enr}}, \tag{3.35}$$

where $\bar{\mathbf{J}}_g^{\text{enr}}$ and $\bar{\mathbf{F}}_g^{\text{enr}}$ are the unknown global vectors of all enriched degrees of freedom; the global tangent matrices are given by,

$$\mathbf{K}_J^{\text{enr}} = \sum_e \int_{\Omega^e} (\mathbf{N}_J^{\text{enr}})^T \mathbf{N}_J^{\text{enr}} \, \mathrm{d}\Omega^e, \tag{3.36}$$

$$\mathbf{K}_F^{\text{enr}} = \sum_e \int_{\Omega^e} (\mathbf{N}_F^{\text{enr}})^T \mathbf{N}_F^{\text{enr}} \, \mathrm{d}\Omega^e; \tag{3.37}$$

tand the residuals matrices are given by,

$$\mathbf{R}_J^{\text{enr}} = \sum_e \int_{\Omega^e} (\mathbf{N}_J^{\text{enr}})^T \left(\tilde{J} - \mathbf{N}_J^{\text{reg}} \bar{J}^{\text{reg}} \right) \, \mathrm{d}\Omega^e, \tag{3.38}$$

$$\mathbf{R}_F^{\text{enr}} = \sum_e \int_{\Omega^e} (\mathbf{N}_F^{\text{enr}})^T \left(\tilde{\mathbf{F}} - \mathbf{N}_F^{\text{reg}} \bar{\mathbf{F}}^{\text{reg}} \right) \, \mathrm{d}\Omega^e. \tag{3.39}$$

In the above equations $\sum_e$ indicates the matrix assembly of the global system from the element matrices. Thus, the idea here is to simply calculate the enriched DOFs by performing the L^2 projections (3.32) and (3.33) [34] such that the deformation field quantities are accurately described in the elements cut by the interface.

3.6 Solution Algorithm

The numerical strategy progressively converges towards equilibrium by solving a series of pseudo steady states of flow until the velocity vanishes everywhere in the domain. The initially non-linear problem is decomposed in linear momentum and transport equation that are solved in a staggered way as follows:

1. At time $t = 0$, $\hat{\mathbf{F}}(0) = \mathbf{I}$ and $J(0) = 1$
2. In the elastic domain $\Omega^{\text{s}}(t)$, given $\hat{\mathbf{F}}^t$ and J^t, compute $\mathbf{v}^{t+dt}$ by solving (3.15).
3. Given $\mathbf{v}^{t+dt}$, update the position of Γ, which yields the new domains Ω^s_{t+dt}.
4. Given $\mathbf{v}^{t+dt}$ and $\Omega^{\text{s}}(t+dt)$ compute regular and enriched nodal degrees of freedom for $\hat{\mathbf{F}}^{t+dt}$ and J^{t+dt}.

5. if $||\hat{\mathbf{F}}^{t+dt} - \hat{\mathbf{F}}^{t}|| < \text{Tol}_F$ and $||J^{t+dt} - J^{t}|| < \text{Tol}_J$ and $||\mathbf{v}^{t+dt}|| < \text{Tol}_v$, end of computation. Else, set $t = t + dt$ and go to step 2.

3.7 Damage transport

Damage represents material defects, generally occurring as surface discontinuities (microcracks) or volume discontinuties (microvoids or cavities). When damage is present in a solid body, the effective area (excluding voids or cracks) is given by $\tilde{A}_{eff} = A - A_w$, A is the total area of solid body and A_w is the damaged area of the solid body. Effective stress in the domain $\tilde{\boldsymbol{\sigma}}_{eff} = \frac{\boldsymbol{\sigma}}{1-w}$ where $\boldsymbol{\sigma}$ is the force per unit damaged area (including cracks or voids) and w is the damage parameter. This damage parameter may have a value ranging from 0 meaning no damage in the body to 1 which represents entirely damaged material. The principle of strain equivalence states that "the strain associated with a damaged state under the applied stress is equivalent to the strain associated with its undamaged state under the effective stress" [35]. We use the Eulerian, updated Lagrangian and total Lagrangian descriptions for transporting damage in the solid domain. The basic difference between the updated and total Lagrangian descriptions is the choice of the reference configuration at each step of numerical simulation. In the updated Lagrangian, the current configuration becomes the reference configuration for the next iteration, that is, the reference domain is updated after each time step. In the total Lagrangian description, the initially defined domain is chosen as the reference configuration for all time steps.

3.7.1 Damage transport using Lagrangian formulation

In Lagrangian framework, assuming there is no damage growth, the damage transport equation is given by [36],

$$\dot{w}_{ij} = \frac{dw_{ij}}{dt} = 0. \tag{3.40}$$

The above equation states that the damage at material points does not change with time. In the updated Lagrangian description, for each node inside the solid domain, material

particle at its spatial domain $\mathbf{x}_i^{t+dt}$ is backtracked to earlier position $\mathbf{x}_i^t$. The damage value is interpolated at $\mathbf{x}_i^t$ using the nodal damage information for that element using

$$w(\mathbf{x},t) = \bar{\mathbf{N}}_w(\mathbf{x},t)\bar{w}(t). \tag{3.41}$$

For the damage transport, we consider the whole domain to be made by a single material without any sharp material interfaces, so the shape function matrix $\bar{\mathbf{N}}_w$ and element vector $\bar{w}$ will only have the terms corresponding to the standard DOFs (no enriched DOF) as :

$$\bar{w} = [\bar{w}^{\text{reg}}]_{4\times 1} \;\; \bar{\mathbf{N}}_w = \left[\begin{array}{c} \mathbf{N}_w^{\text{reg}} \end{array} \right]_{1\times 4} \tag{3.42}$$

The obtained damage value is then assigned to the material particle at $\mathbf{x}_i^{t+dt}$. A similar procedure is carried out in the total Lagrangian formulation, except that a material particle is backtracked to $\mathbf{x}_i^0$, position of the particle at initial configuration.

3.7.2 Damage transport using Eulerian formulation

In the Eulerian framework, the damage transport equation reads:

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{\partial w}{\partial t} + \mathbf{v}\cdot\nabla w = 0, \tag{3.43}$$

$$\frac{w(t+dt)}{\Delta t} + \mathbf{v}\cdot\nabla w(t+dt) = \frac{w(t)}{\Delta t}. \tag{3.44}$$

The initial condition of damage is known for the system and this damage is updated in each pseudo time using the above relation. Here we solve for $w(t+dt)$ by linearizing the above equation into $K_w \times w(t+dt) = F_w$ system where,

$$\begin{aligned} K_w &= \int_{\Omega^{\text{e}}} \left(\frac{\bar{\mathbf{N}}_w^T \bar{\mathbf{N}}_w}{\Delta t} + \bar{\mathbf{N}}_w^T \mathbf{v} \hat{\mathbf{B}}_w \right) \mathrm{d}\Omega \\ F_w &= \int_{\Omega^{\text{e}}} \frac{\bar{\mathbf{N}}_w^T \bar{w}}{\Delta t} \, \mathrm{d}\Omega \end{aligned} \tag{3.45}$$

where,

$$\begin{aligned}\hat{\mathbf{B}}_w &= \left[\mathbf{B}_w^1, \ldots, \mathbf{B}_w^4\right]_{2\times 4} \\ \mathbf{B}_w^I &= \begin{bmatrix} \frac{\partial \hat{N}^I}{\partial x_1} & \frac{\partial \hat{N}^I}{\partial x_2} \end{bmatrix}_{2\times 1}\end{aligned}$$

CHAPTER 4

Large deformation modeling in Abaqus

Most problems in Abaqus use a purely Lagrangian description. In a pure Lagrangian model, the mesh is attached with the material, so the mesh moves with the material. In many non-linear simulations, the material in the structure undergoes very large deformation which distorts the finite element mesh, often to the point where either the mesh is unable to provide accurate results or the analysis terminates for numerical reasons such as convergence problems, excessive distortions, etc. However, in the Lagrangian approach, it is easy to track free surfaces and apply boundary conditions in the problem domain; and it is also simpler and computationally less expensive.

4.1 Abaqus methodology

Abaqus is commonly used for modeling and solving large deformation problems in the real world. Abaqus, a software suite for finite element analysis and computer aided engineering, involves pre-processing or modelling, processing or finite element analysis and post-processing sequences. The first step of modelling includes defining the parts, materials and sections. While defining materials, the user can input different properties which include general property like density, mechanical properties like elasticity, plasticity, damage, viscosity, thermal properties like conductivity, specific heat, etc. Moreover, users can define their own material models using UMAT and VUMAT. This feature is very general and powerful and any mechanical constitutive model can be added. Different analysis steps are then defined which can be either statics, dynamics, heat transfer type of problem or coupled thermal-electric problems. After materials and analysis steps have been defined, loads and boundary conditions are assigned to the respective nodes or surfaces. The problem domain is then divided into small finite elements after meshing. Element type is assigned in the same step. This completes the modelling step and we move forward for submitting the

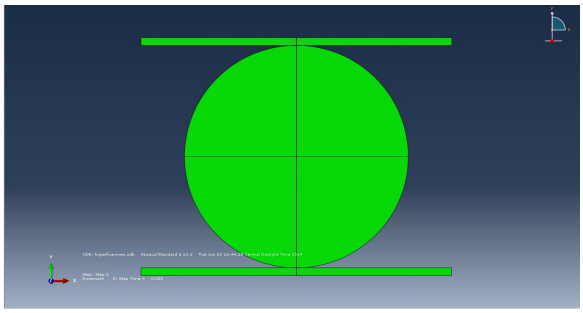
job and analyzing it. The submitted job can be monitored to know the progress and time taken in each step of the analysis. Also, it displays warnings and/or errors incurred during the analysis. Once the analysis is successfully completed, user can switch to visualization mode to get results of the analysis. Now, we further explain the steps in dealing with the modelling and analysis of hyperfoam and hyperelastic material within Abaqus.

4.2 Hyperfoam material

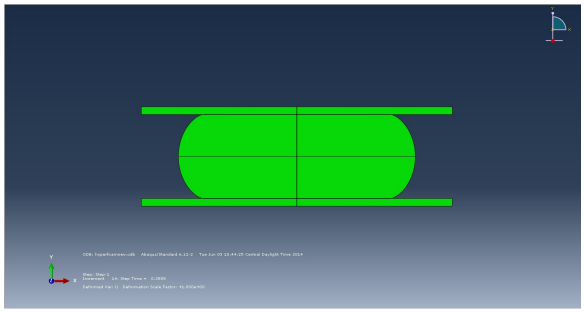
The hyperfoam material is based on the hyperelastic material theory. The difference between hyperfoam and hyperelastic material is that the former one is highly compressible. Hyperfoam material model is isotropic and nonlinear and is valid for large volumetric changes. This kind of material can undergo up to 90 % strain in compression requiring geometrical nonlinearity be accounted for. Abaqus has an in-built Hyperfoam material model. There are two approaches of defining this material model, the first one is by manually entering the material properties like μ and ν of the material and the second one is by entering the experimental test data available from uniaxial, biaxial, planar, simple shear or volumetric test. Here we will present a simple example problem of modeling hyperfoam material using Abaqus and show that such kind of material model can be used for large deformation simulations. Readers are referred to Abaqus analysis user's manual 22.5.2 for details about the mechanical behavior and strain energy potential of hyperfoam material.

We consider an elastic compressible cylinder of radius R=0.81 cm made up of hyperfoam ($E_Y = 15.0$ MPa and ν = 0). This cylinder is compressed between two rigid plates on the top and bottom. As the system has four-fold symmetry, we model only one quarter of the geometry. The interaction properties between the rigid plates and elastic cylinder is defined as hard contact and constraint enforcement method is selected as penalty approach. While defining the interaction, the slave surface is defined on the elastic cylinder (softer material) and master surface as the rigid plate. A displacement boundary condition is applied such that the plate moves to the final position as shown in Figure 4.1 (d). Figure

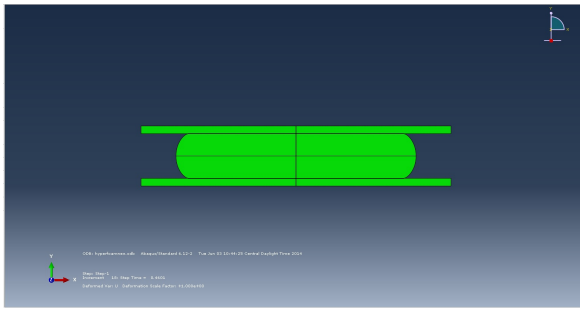
4.1 (a) is the initial geometry of the model showing the cylinder with the plate position and Figure 4.1 (b) and Figure 4.1 (c) are the intermediate stage during the compression process. Figure 4.1 (d) refers the final equilibrium stage showing the fully deformed elastic cylinder. This is an example of using in-built material model in Abaqus for large deformation modeling.



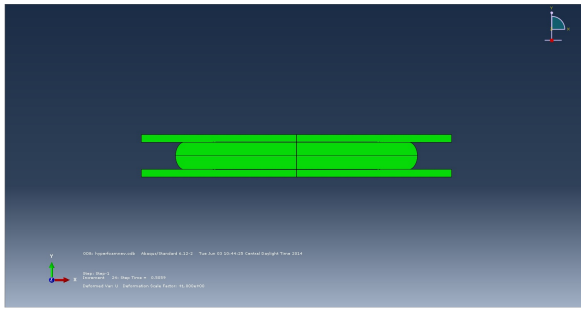
(a) Initial geometry of cylinder and plates



(b) Intermediate deformed shape of the cylinder



(c)



(d) Fully squeezed cylinder

Figure 4.1: Abaqus modeling of elastic circular cylinder made up of hyperfoam. The cylinder is located between two rigid plates on the top and bottom that moved towards each other. The plates are gradually moved to the final position shown in (d).

4.3 Hyperelastic material

The stress-strain relationship of hyperelastic material can be defined as non-linearly elastic, isotropic, incompressible and independent of strain rate. Different forms of hyperelastic material models are available in Abaqus such as Arruda-Boyce, Marlow, Mooney-Rivlin, Neo Hooke, Ogden and so on. Here, we use the user defined form of material model with the strain energy function as described in section 2.3. The number of property values that define the material model is specified while defining such material behaviors. The user

subroutine UHYPER must be specified in the edit job dialogue while running the model.

4.3.1 1-D modeling of rectangular bar subjected to tension

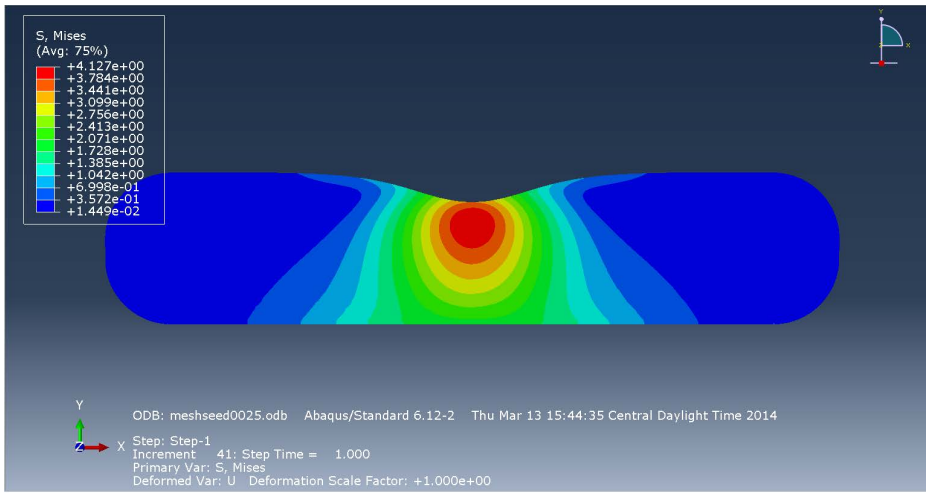
A rectangular domain of dimensions 0.2 m $\times$ 0.11 m is modelled in Abaqus (see Figure 5.1 for schematic diagram of the model). While defining the mechanical property, the material is selected as Hyperelastic material. Isotropic material type is selected and strain energy potential function is chosen to be user defined. This means a user defined subroutine UHYPER is to be supplied while submitting the job for analysis. The number of property values that is to be assigned to define the material is selected to be two which corresponds to the bulk modulus (κ =10 MPa) and shear modulus (μ = 6 MPa) values. Under the steps category, NIgeom is turned on. This controls the inclusion of nonlinear effects of large displacement. Under the boundary condition, rollers are provided in all except bottom side of the rectangular domain. The bottom side is subjected to a load of 2 MPa magnitude. Seed size is taken to be 0.01 m in both directions. Under the category of element type in mesh, plain strain is chosen as family so as to impose plane strain conditions. This defines a 4-node bilinear plane strain quadrilateral (CPE4) element type. We will later on compare the results obtained from this simulation with that obtained from CEL formulation along with analytical solution and is shown in Figure 5.4(c).

4.3.2 2-D modeling of a rounded rectangular solid

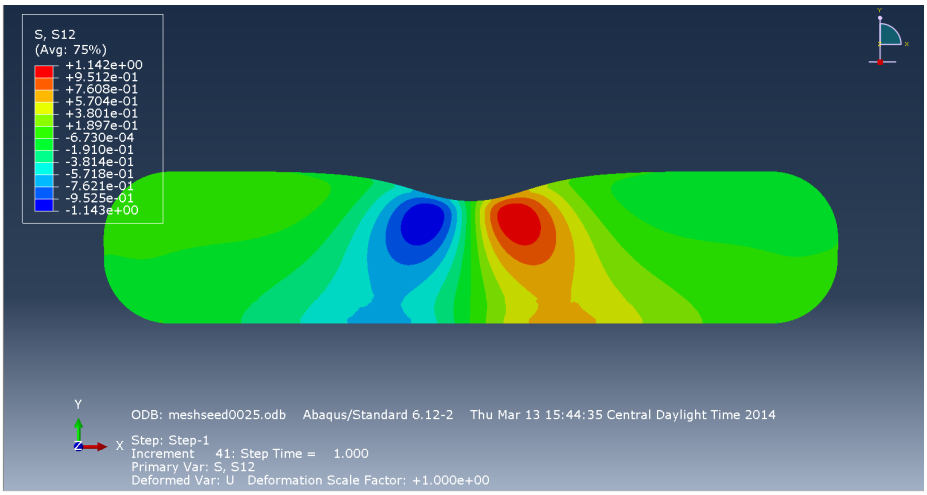
We consider a rounded rectangular solid made up of hyperelastic material ($E_Y = 15.0$ MPa and Poisson's ratio $\nu = 0.25$) . The dimensions of the straight portion of the rounded rectangle are 3.5 cm $\times$ 0.92 cm and the rounded edges are semicircles with radius 0.46 cm. The computational domain is discretized using an element size $h = 0.025$ cm. The solid is restrained in the x_1 and x_2 directions on the bottom surface.

4.3.2.1 Subjected to vertical pressure

Load is defined via subroutine DLOAD given in Appendix B. This load is assigned to the top surface of rectangular solid. The value $\alpha = 0.25$ cm in the DLOAD represents the standard deviation and is the spread of the applied pressure of magnitude 6 MPa around the central point. Load beyond 6 MPa could not be simulated for the given example problem as Abaqus showed convergence issues for higher load. This is due to the large deformation in the domain which causes excessive mesh distortion and this could not be handled by the Lagrangian way of problem solving as is done by Abaqus. Figure 4.2(a) shows the Von



(a) Von Mises Stress



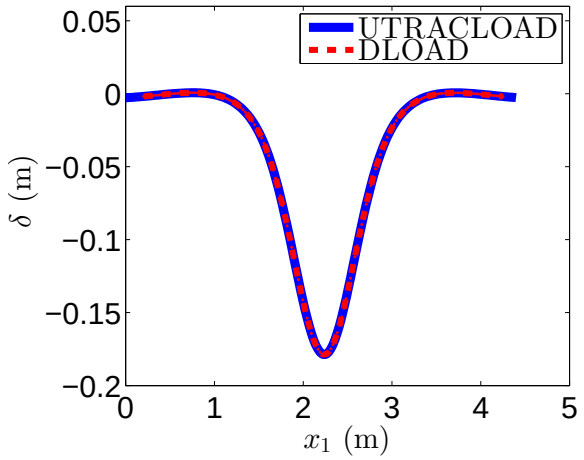
(b) Shear stress σ_{12}

Figure 4.2: Stress distribution in the hyperelastic rounded rectangular domain.

Mises stress in the rectangular domain with the maximum value of 4.127 MPa whereas Figure 4.2 (b) shows the shear stress with maximum of 1.142 MPa.

4.3.2.2 Subjected to surface traction

Another way of solving the same problem as in section 4.3.2.1 is by defining surface traction with the help of UTRACLOAD. In this case, instead of pressure, we define the surface traction and the direction in which the traction works. UTRACLOAD is shown in Appendix B. We compare the top surface displacement of the rectangular domain due to the applied DLOAD and UTRACLOAD in Figure 4.3 and find that the two plots exactly match each other . This is because the load of 6 MPa is too small to cause any major deformation of the top surface of the rounded rectangular solid that there is negligible change in normal direction of the surface.



(a)

Figure 4.3: Top surface displacement comparison for applied pressure and surface traction using subroutines DLOAD and UTRACLOAD respectively.

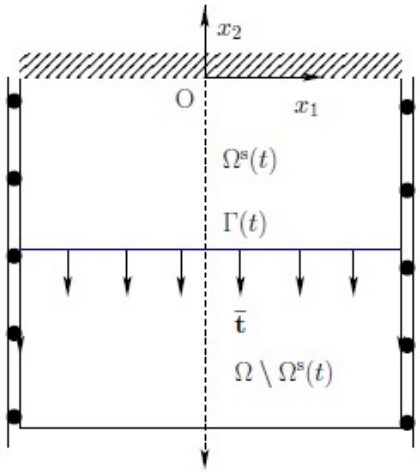
This example problem will be later used to benchmark the proposed CEL method and show that CEL method behaves as accurately as Abaqus for small load. Moreover, it will also be shown that CEL method is better for higher load where the Lagrangian way of solving the problem in Abaqus fails due to excessive mesh distortion.

CHAPTER 5

Example problems using CEL

5.1 Uniaxial extension of rectangular bar

Let us consider a rectangular domain of dimensions 0.2 m (width x_1) $\times$ 0.25 m (depth x_2) containing a soft hyperelastic solid that is of dimensions 0.2 m $\times$ 0.11 m. We assume that the solid, characterized by a Young's modulus $E_Y = 15.0$ MPa and Poisson's ratio $\nu = 0.25$, is fixed at the top end and subjected to normal traction of $\bar{\mathbf{t}} = -2\hat{e}_2$ MPa at the bottom end (see Figure 5.1). The sides of the solid are constrained in the horizontal direction so that deformation gradient component $F_{11} = 1$ at all times. We discretize the domain using square (9-node and 4-node) finite elements of size $h = 0.0125$ m. We neglect the effect of gravity and assume zero body forces. The boundary and initial conditions of this simplified benchmark problem are:

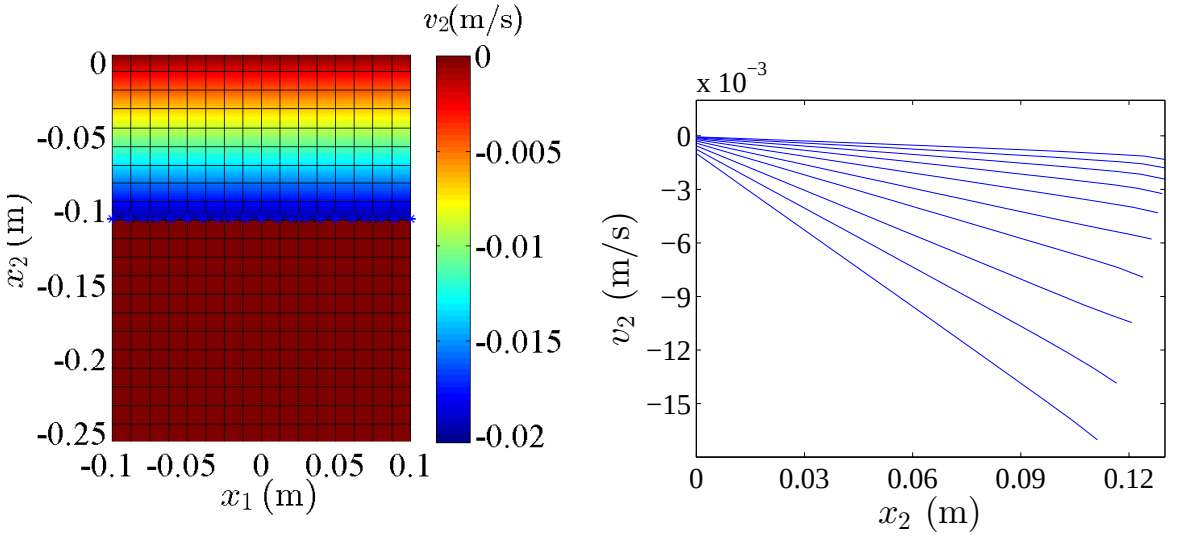


(a)

Figure 5.1: Schematic diagram of the uniaxial extension of a soft rectangular bar. A traction of $\bar{\mathbf{t}} = -2$ MPa is applied to the end of the bar to deform it elastically.

$$\left.\begin{aligned} \bar{\mathbf{t}} &= -2\hat{e}_2 \quad \text{on } \Gamma, \\ v_1(x_1 = -0.1,\, x_2) = v_1(x_1 = 0.1,\, x_2) &= 0, \\ \mathbf{v}(x_1,\, x_2 = 0) &= \mathbf{0}, \\ \mathbf{v}(\mathbf{x},\, t = 0) &= \mathbf{0}. \end{aligned}\right\} \tag{5.1}$$

As soon as the traction is applied at pseudo-time $t = 0$, the material in the solid domain moves downwards with a non-zero velocity $\mathbf{v}$ and consequently the solid elongates in the x_2 direction. The vertical component of the velocity field v_2 is negative (downward motion) and varies linearly in the x_2 direction as shown in Figure 5.2. With each pseudo-time step (or iteration) the component v_2 decreases and eventually the solid reaches its equilibrium state when $v_2 \to 0$.



(a) Initial velocity variation in the domain

(b) Velocity variation with depth after every 25 iterations

Figure 5.2: Numerical results from the CEL formulation for the downward flow of material under applied uniaxial tension.

Next, we investigate the performance of the mixed formulation for simulating compressible hyperelastic behavior for $\nu = 0$ and $\nu = 0.25$ using three different finite element (FE) interpolation strategies:

1. **Bilinear**: 4-node FE interpolation of $\mathbf{v}$, $\hat{\mathbf{F}}$ & J

2. **Biquadratic**: 9-node FE interpolation of $\mathbf{v}$, $\hat{\mathbf{F}}$ & J

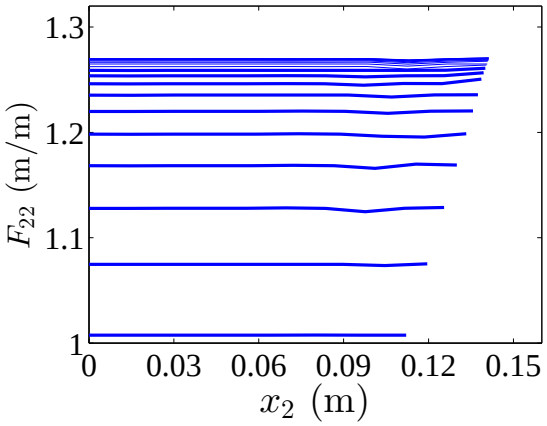
3. **Mixed**: 9-node FE interpolation of $\mathbf{v}$ and 4-node FE interpolation of $\hat{\mathbf{F}}$ & J

In the case of uniaxial extension in x_2, we have $J = F_{22} > 1$, since $F_{11} = F_{33} = 1$ and all other components of $\mathbf{F}$ vanish. Therefore, it is sufficient to only observe the behavior of F_{22} from $t = 0$ until equilibrium. In the following figures, we plot the variation of F_{22} in the x_2 direction at every 50 iterations. Note that the length of the solid increases and the change in F_{22} decreases with each iteration as we approach equilibrium. We can see from Figure 5.3 that for $\nu = 0$ the bilinear and mixed interpolation strategies work equally well, whereas the biquadratic interpolation strategy suffers from spurious oscillations close to the traction boundary. From Figure 5.3 we can observe that for $\nu = 0.25$ both the bilinear and biquadratic interpolation strategies suffer from spurious oscillations, whereas the mixed interpolation strategy is least affected. This study demonstrates that the mixed interpolation strategy leads to better accuracy and stability compared to the uniform interpolation strategies.

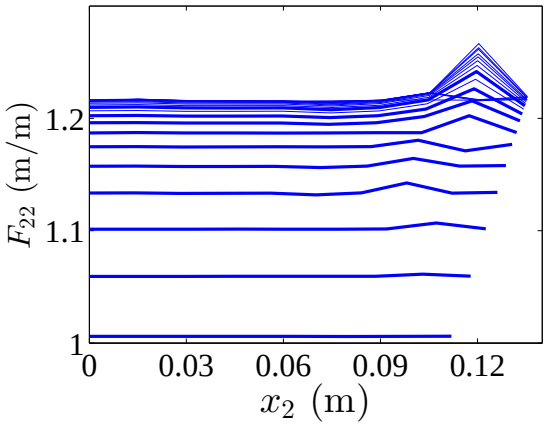
We next investigate the accuracy of the scheme by comparing the analytical and numerical equilibrium stress versus deformation curves. Using the constitutive law given in equation (2.16), we can derive the analytical expression for the Cauchy stress component σ_{22} as,

$$\sigma_{22} \;=\; \frac{1}{F_{22}}\left[\kappa \log(F_{22}) + \frac{2}{3}\mu F_{22}^{-2/3}(F_{22}^2 - 1)\right] \tag{5.2}$$

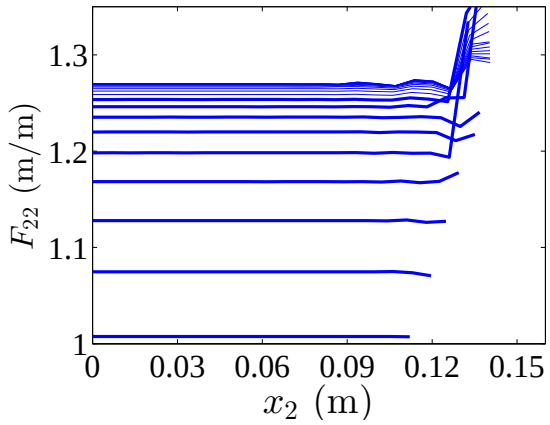
Now, for different values of applied normal traction $\bar{\mathbf{t}} \cdot \hat{e}_2 = \sigma_{22} \in [-4\ 4]$ MPa we numerically evaluate the equilibrium value of F_{22} for three different values of Poisson's ratio ν = 0, 0.25 and 0.45. These numerical results are then plotted as a scatter over the analytical solution (solid lines) given in (5.2). The excellent match of the numerical results with the analytical solution in Figure 5.4(a) illustrates the accuracy of the method. For the Poisson's ratio ν=0.25, we compare the result with that obtained using UHYPER as explained



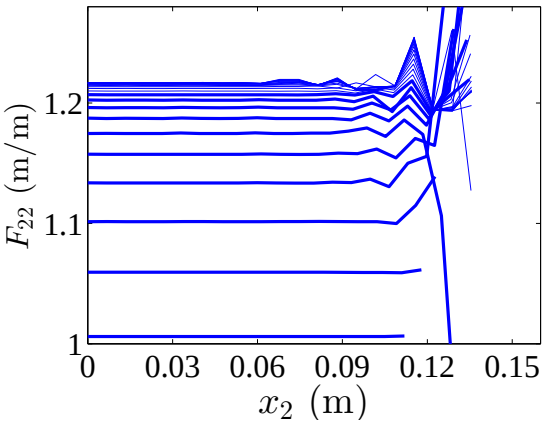
(a) Bilinear $\mathbf{v}$, $\hat{\mathbf{F}}$ and J; ν=0



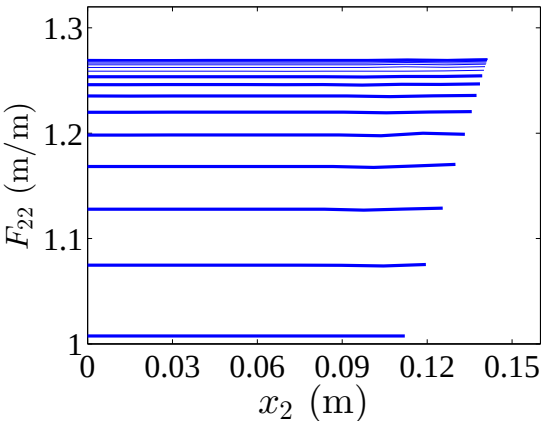
(b) Bilinear $\mathbf{v}$, $\hat{\mathbf{F}}$ and J; ν=0.25



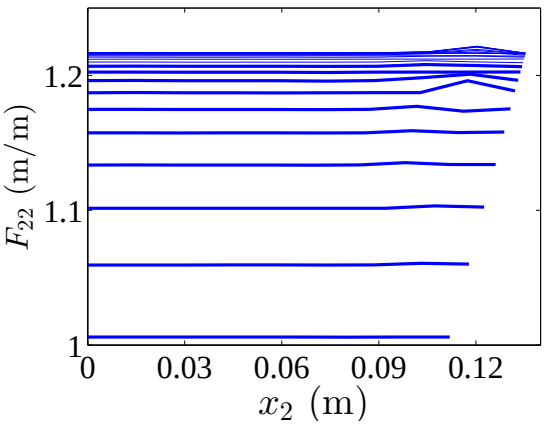
(c) Biquadratic $\mathbf{v}$, $\hat{\mathbf{F}}$ and J; ν=0



(d) Biquadratic $\mathbf{v}$, $\hat{\mathbf{F}}$ and J; ν=0.25



(e) Biquadratic $\mathbf{v}$, bilinear $\hat{\mathbf{F}}$ and J; ν=0



(f) Biquadratic $\mathbf{v}$, bilinear $\hat{\mathbf{F}}$ and J, ν=0.25

Figure 5.3: Performance of the mixed element formulation for uniaxial tension test. Variation of F_{22} along the length of domain is shown for bilinear, biquadratic and mixed formulation for two compressible materials with Poisson's ratio ν=0 (left column) and ν=0.25 (right column).

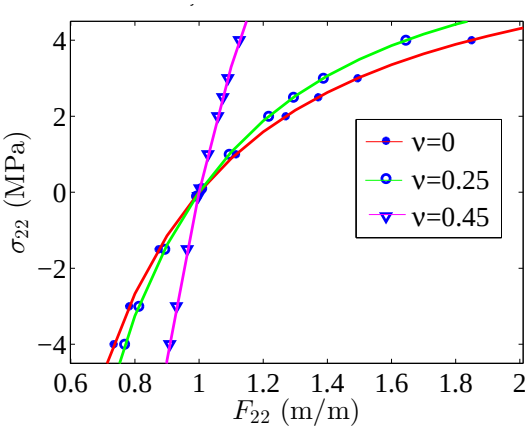
in section 4.3.1 and find that the result obtained from CEL, Abaqus using UHYPER and analytical solution as in equation (5.2) matches exactly and is shown in Figure 5.4(c). We also evaluate the variation of the error in the reference (initial) volume of the solid at each iteration to check for the conservation of mass. The initial volume $V^{(0)} = 0.044$ m^3 and at each iteration (i) we can calculate the percentage error as,

$$\varepsilon^{(i)} = \frac{V^{(0)} - V^{(i)}}{V^{(0)}} \times 100, \text{ where } V^{(i)} = \int_{\Omega^s} \frac{1}{J^{(i)}}\, dV. \tag{5.3}$$

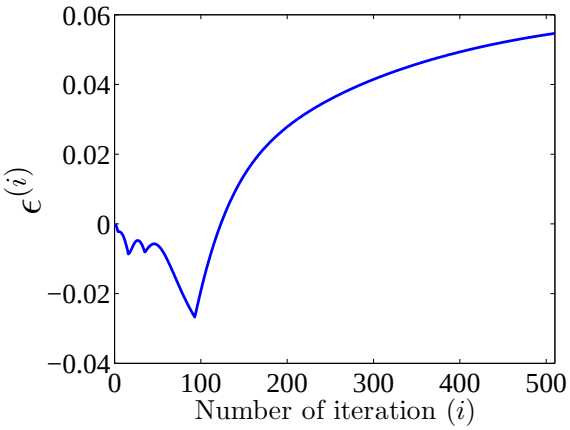
The variation of $\varepsilon^{(i)}$ with iterations is plotted in Figure 5.4(b). As we can see the error initially oscillates and after 500 iterations or so it gradually reaches a steady state. However, it is important to note that percentage error $\varepsilon^{(i)} < 0.06$ (i.e. error is 0.0006) indicating that the scheme is quite accurate in conserving the mass of the elastic solid. Since the volume error is so low at all times, the convergence criterion is based on the L^2 error in velocity or deformation gradient.

5.2 Simple shear of a rectangular block

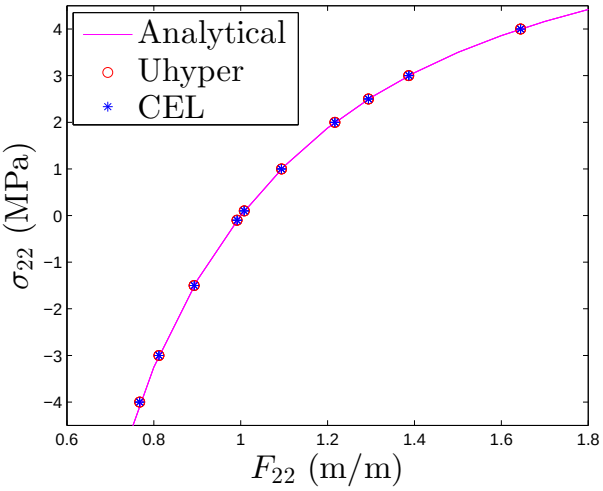
Let us now study the shear flow of a solid under applied shear traction. Once again, we consider a rectangular domain of dimensions 0.2 m $\times$ 0.25 m and solid domain of dimensions 0.2 m $\times$ 0.11 m. The domain is discretized with an element size $h = 0.0125$ m. We assume the Young's modulus $E_Y = 15.0$ MPa and Poisson's ratio $\nu = 0.25$. The solid is fixed at the top end and subjected to shear traction of $\bar{\mathbf{t}} = -0.4\hat{e}_1$ MPa at the bottom end. On the left and right boundaries, we impose zero velocity in the $\mathbf{e_2}$ direction to strictly prescribe horizontal shear flow. The boundary and initial conditions of this simplified benchmark



(a) Analytical (solid lines) and numerical curves (scattered points) for stress versus deformation gradient



(b) Percentage error in elastic body mass with pseudo time (iteration) steps



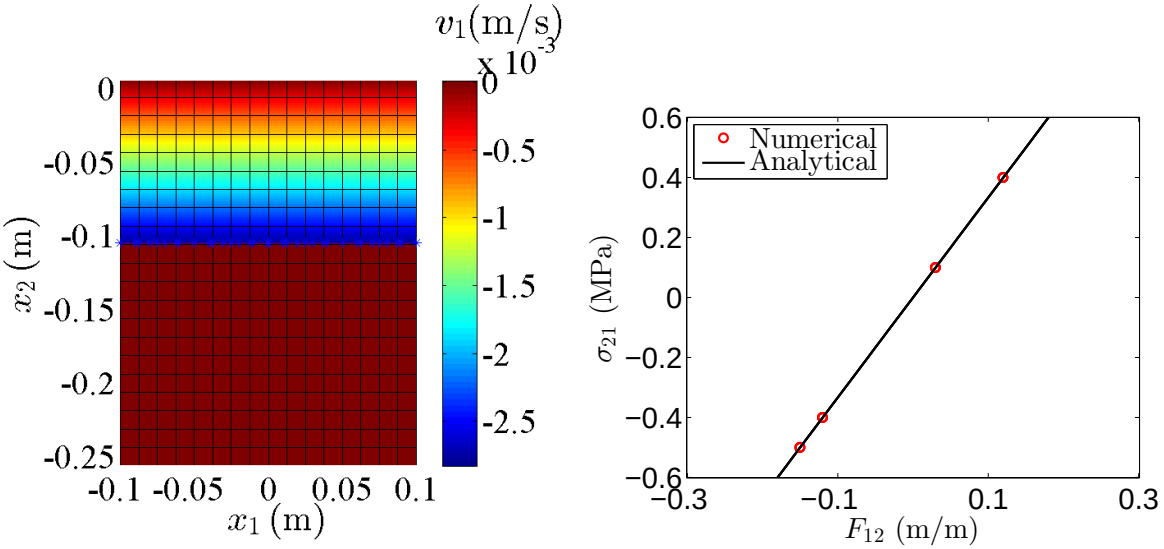
(c) Stress versus deformation gradient obtained via CEL, Abaqus and analytical solution for $\nu = 0.25$

Figure 5.4: Validation and error analysis of numerical results from the CEL formulation for uniaxial tension test

problem are,

$$\left.\begin{aligned} \bar{\mathbf{t}} &= -0.4\hat{e}_1 \quad \text{on } \Gamma, \\ \hat{\mathbf{F}}(x_1 = -0.1,\, x_2 \leq 0.11) = \hat{\mathbf{F}}(x_1 = 0.1,\, x_2) &= \mathbf{I}, \\ J(x_1 = -0.1,\, x_2) = J(x_1 = 0.1,\, x_2) &= 1, \\ \mathbf{v}(x_1,\, x_2 = 0) &= \mathbf{0}, \\ \mathbf{v}(\mathbf{x},\, t = 0) &= \mathbf{0}. \end{aligned}\right\} \tag{5.4}$$

We discretize the domain using the mixed interpolation strategy as discussed in the previous section with an element length of $h = 0.0125$ m in both x_1 and x_2 directions. Due to the applied shear, the material flows from right to left as shown in Figure 5.5a, so the velocity is negative. In the case of simple shear flow in x_1 direction, we have $F_{12} > 0$, $F_{22} = F_{11} = F_{33} = 1$ and all other components of $\mathbf{F}$ are zero. Therefore, it is sufficient to only observe the behavior of F_{12} from $t = 0$ until equilibrium. We next plot the match



(a) Initial velocity variation in the domain

(b) Analytical and numerical curves for stress versus deformation gradient

Figure 5.5: Numerical results from the CEL formulation for the shear flow of material under applied shear traction. The results are in agreement with theory, thus, validating our scheme.

between the analytical and numerical equilibrium stress versus deformation curves. From

the constitutive relation in equation (2.16), we can write the analytical expression for the Cauchy stress component $\sigma_{12} = \mu F_{12}$. For four different values of applied shear stress, we plot the numerical results (scatter) against the analytical solution (solid line) in Figure 5.5(b). We observe an excellent agreement between theory and simulation with a linear response in the applied stress range. Since shear flow is isochoric, the error in volumetric deformation identically vanishes.

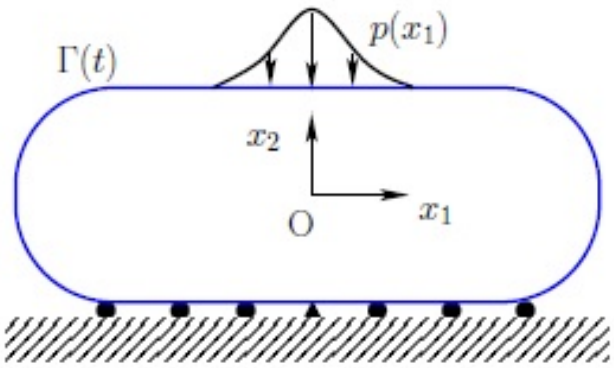
5.3 Indentation of a rounded rectangular solid

Let us consider a rounded rectangular solid made up of the same soft material as in the previous example ($E_Y = 15.0$ MPa and $\nu = 0.25$). The dimensions of the straight portion of the rounded rectangle are 3.5 cm $\times$ 0.92 cm and the rounded edges are semicircles with radius 0.46 cm. The solid domain and test configuration are chosen to mimic a hydrogel placed onto a relatively rigid substratum, typically seen in tissue printing. The total computational domain is 5.2 cm $\times$ 1.2 cm and is discretized using an element size $h = 0.1$ cm. A Gaussian-shaped vertical pressure field with amplitude p (MPa) is prescribed on the top surface centered at mid-span as follows:

$$p(x_1) \quad = \quad p_0 \exp\left(-x_1^2/\alpha^2\right) \tag{5.5}$$

where $\alpha = 0.25$ cm is the standard deviation and represents the spread of the applied pressure around the central point. The solid is restrained in the x_1 and x_2 directions on the bottom surface. It is important to note that the bottom surface is restrained in the x_1 and x_2 directions by enforcing these Dirichlet conditions on the underlying grid nodes that are closest to the interface, and not on the interface itself. In order to limit the error created in doing so, we position the rounded rectangle such that its bottom interface remains very close to the nodes of the underlying mesh. Alternatively, Dirichlet boundary conditions can easily be enforced directly on the interface with the use of Lagrange multipliers. The geometry and the boundary conditions are illustrated in the Figure 5.6. The initial unde-

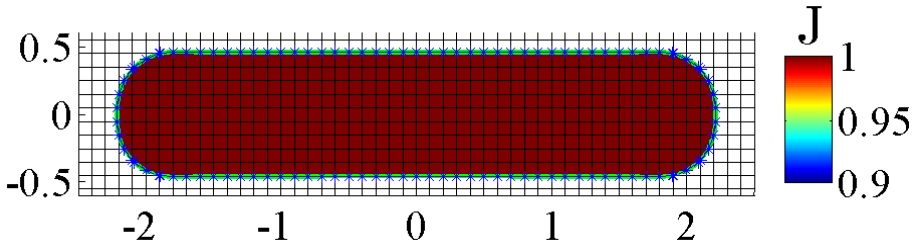
formed shape and the final or equilibrium deformed shape of the solid under an applied pressure amplitude of $p_0 = 6$ MPa are shown in Figures 5.7a and 5.7b. The surface plot of the Jacobian determinant J in Figure 5.7b shows that the material experiences compression at the center ($J < 1$) and some tension as we move towards the ends; however, far away from the center the material is unstressed $J = 1$.



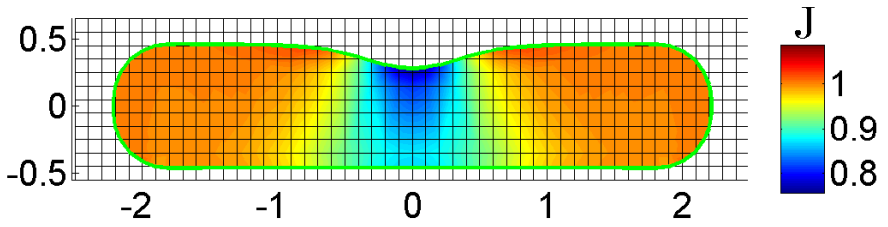
(a)

Figure 5.6: Schematic diagram of the indentation of soft solid. A Gaussian type pressure load is applied to simulate the contact between a rigid indenter and the solid. At the bottom the solid is allowed to slip, however, due to symmetry the center node is pinned.

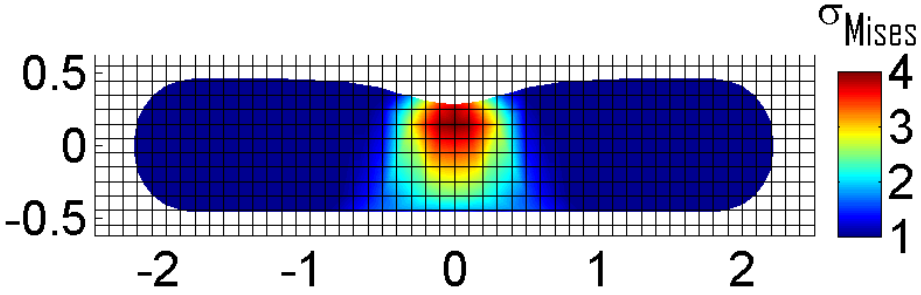
To benchmark our simulation, we analyze the problem with a fully Lagrangian finite element formulation (using the software Abaqus with the UHYPER subroutine). For $p_0 = 6$ MPa, the deformed shape of the top surface of the solid and the variation of J obtained from Abaqus and our coupled Eulerian-Lagrangian (CEL) formulation are plotted against each other in Figure 5.8a and Figure 5.8b, respectively. The maximum error in the displacement of the top surface δ is 1.28 % and the maximum error in Jacobian J along the free surface is 0.39 %. Next, we check the mass conservation behavior of the CEL implementation by plotting the error in mass $\varepsilon_{\text{mass}}$ with iterations or pseudo-time steps, as given in Figure 5.8c. The error increases initially, reaches a maximum around 100^{th} iteration and then decreases to reach a steady state value as the equilibrium is attained. We now find the error in Von Mises stress and pressure between Abaqus and CEL formulation. For this, we take all nodes in Abaqus and then interpolate the stress value for these nodes in different



(a) Initial undeformed



(b) Equilibrium deformed reached after 200 psuedo-time steps (iterations)



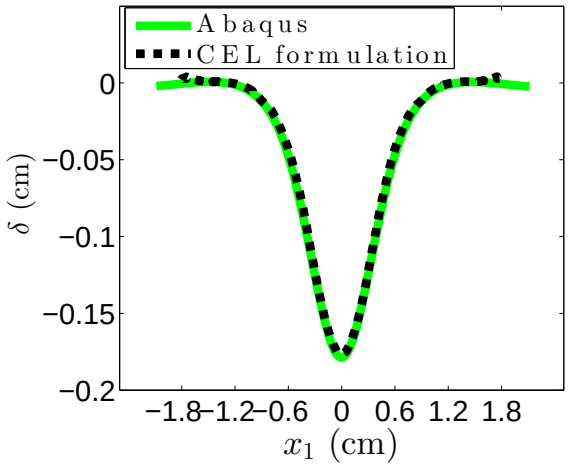
(c) Von Mises stress distribution in the domain at equilibrium

Figure 5.7: Numerical results showing the Jacobian determinant of the deformation, Von Mises stress and the shape evolution of a soft rounded rectangular solid for $p_0 = 6$ MPa during indentation.

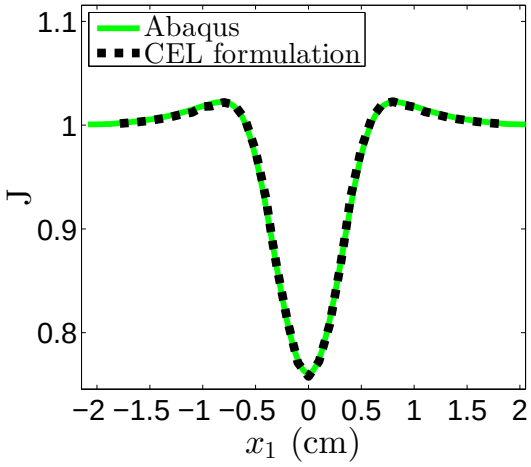
mesh sizes in CEL formulation. We find that the L^2 norm of error in Von Mises stress and pressure decreases very rapidly with element size h, as shown in Figure 5.8d. Next, to demonstrate the robustness and viability of the approach, we apply a larger pressure amplitude of $p_0 = 40$ MPa so as to simulate large material distortions. The final equilibrium shape of the solid is shown in Figure 5.9a where we note that material below the load undergoes large compressive strains with $J = 0.5$ (that is, the material is confined to half it original volume). The evolution of the solid boundary with pseudo-time is then shown in Figure 5.9b. As we can see that the interface initially moves with a high velocity and eventually reaches its final equilibrium shape after about 150 iterations. It is to be noted that at even moderate load of $p_0 = 10$ MPa, the UHPYER implementation in Abaqus crashed just after a few iterations due to convergence issues. For higher loads, ALE algorithms can work; however, CEL formulations have the advantage that they can be used to simulate extreme deformations without requiring mesh moving or remeshing schemes.

5.4 Lateral compression of a cylinder

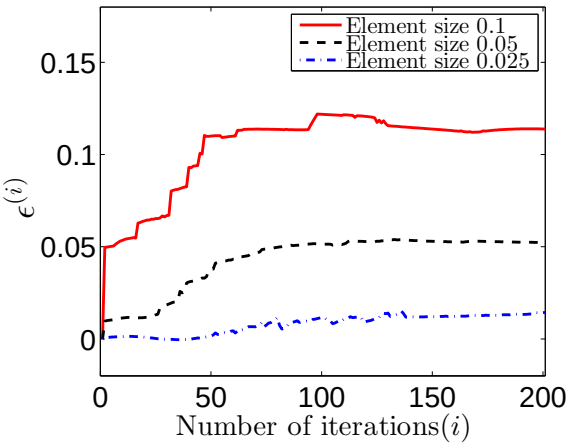
In the previous two benchmark examples, the interface remained flat at all times. Herein, we shall consider an example problem with a curved interface and demonstrate the ability of our formulation to handle its evolution as the solid undergoes very large deformation. Let us consider an elastic compressible cylinder of radius $R = 0.81$ cm, with $E_Y = 15.0$ MPa and $\nu = 0.25$, which is compressed between two planes on the top and bottom. The total computational domain is 3.2 cm $\times$ 2.4 cm and is discretized using an element size $h = 0.08$ cm. Plane strain conditions apply and body forces are neglected. We set up the problem with four-fold symmetry about the origin. The boundary and initial conditions for



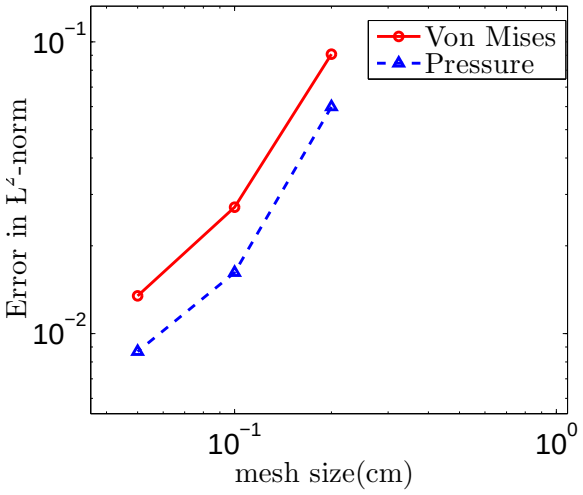
(a) Displacement of top surface at equilibrium



(b) Jacobian determinant along the top surface at equilibrium

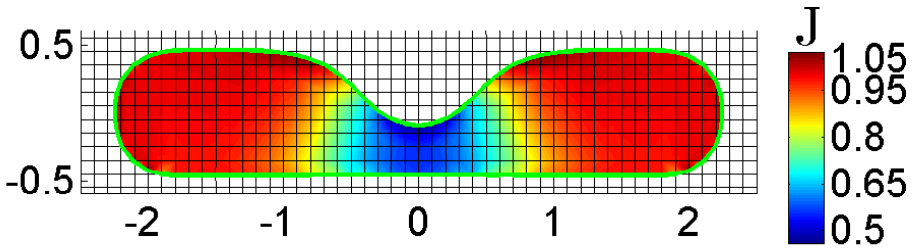


(c) Percentage error in elastic body mass for different meshes

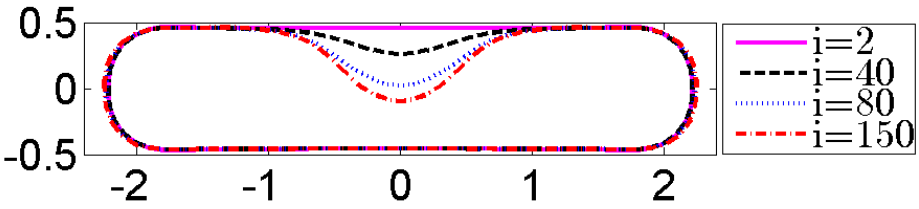


(d) L^2 error in invariants at equlibrium

Figure 5.8: Validation of numerical results from the CEL formulation with the standard Lagrangian formulation in the commercial software Abaqus for the indentation of a rounded rectangular solid. The L^2 error is calculated by taking the Abaqus solution from a very fine mesh as the exact solution.



(a) Equilibrium deformed



(b) Interface locations

Figure 5.9: Numerical results showing the Jacobian determinant of the deformation and the shape evolution of a soft rounded rectangular solid for $p_0 = 40$ MPa during indentation. Abaqus UHYPER subroutine did not converge for this high load case, which demonstrates the robustness of the current CEL formulation.

this problem are,

$$\left.\begin{aligned} v_2(x_1,\, x_2 = 0) = v_1(x_1 = 0,\, x_2) &= \mathbf{0}, \\ \mathbf{v}(\mathbf{x},\, t = 0) &= \mathbf{0}, \\ \hat{\mathbf{F}}(\mathbf{x},\, t = 0) &= \mathbf{I}, \\ J(\mathbf{x},\, t = 0) &= 0. \end{aligned}\right\} \tag{5.6}$$

We define a vertical force that is applied on the portion of interface Γ that is within a certain distance $d_0 =$ from either of the planes. This force function is defined as an exponential repulsive force to avoid penetration between the cylinder and the two compressive planes:

$$\begin{cases} \bar{\mathbf{t}}(\mathbf{x}) = (\phi(\mathbf{x}) - d_0)\exp((d_0 - \phi(\mathbf{x}))\mathbf{e}_2 & \text{if } \phi(\mathbf{x}) \le d_0 \\ \bar{\mathbf{t}}(\mathbf{x}) = 0 & \text{if } \phi(\mathbf{x}) > d_0 \end{cases} \tag{5.7}$$

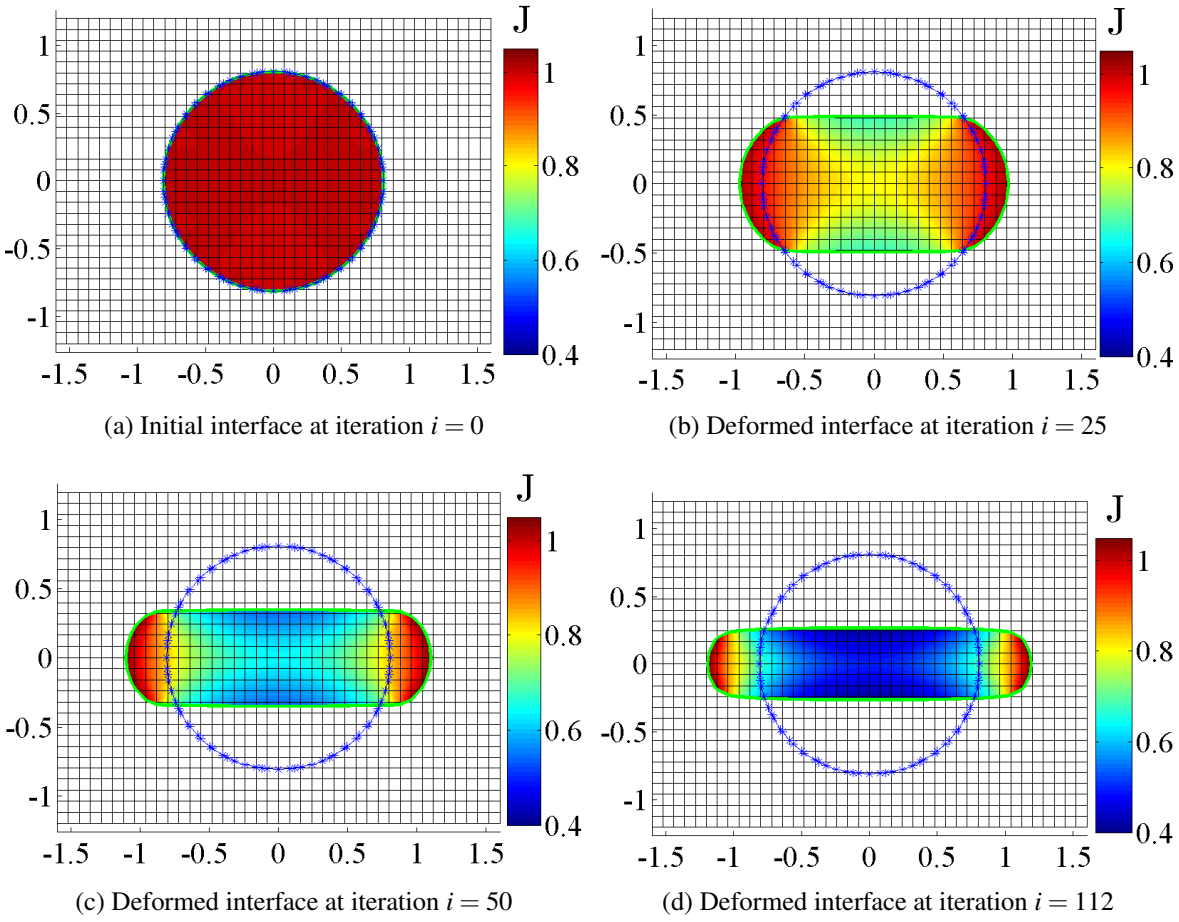
where d_0 represent a cut-off distance over which the repulsive force is applied, and is taken here to be 1% of the radius of the cylinder. As the planes move closer, only a portion of the solid cylinder is subjected to compression, which is clear from the contour plot of J in Figure 5.10. For example, in Figure 5.10(b) we can see that the material in the center is compressed (i.e. $J < 1$), whereas the material on the sides is not (i.e. $J \approx 1$). As the planes move even closer the solid deforms into an elongated shape as shown in Figure 5.10(c) and (d), when the material at certain points is compressed to less than half its initial volume ($J \approx 0.4$). Note that the planes are gradually moved to the final position shown in Figure 5.10(d) until iteration $i = 80$ and then held in position. At iteration $i = 112$, the velocity in the domain vanishes (less than tolerance), so the stress in the solid is at static equilibrium. As opposed to the presented method, a Lagrangian finite element formulation would suffer from large mesh distortion in this deformation regime. To check whether the numerical implementation conserves mass, we consider three mesh sizes as shown in Table 5.1 and calculate the % error in V_0 after each iteration (pseudo-time step). As expected the coarsest mesh has the highest % error of 1.6 and with the refinement of mesh, the % error reduces to as low as 0.16.

Element size	Element in X-dir	Element in Y-dir	% Error
0.16	40	30	1.6
0.08	80	60	0.28
0.04	160	120	0.16

Table 5.1: Percentage error in elastic body mass for different finite element mesh sizes for the deforming cylinder under lateral compression at equilibrium.

5.5 Damage Transport

We consider a rectangular domain with the same dimensions as shown in section 5.1 and simulate damage transport using the updated Lagrangian, total Lagrangian and Eulerian framework. Unlike in section 5.1, the whole rectangular domain contains a soft hyperelastic solid, so for these simulations, there is no discontinuty in the elements. We discretize the



(a) Initial interface at iteration $i = 0$

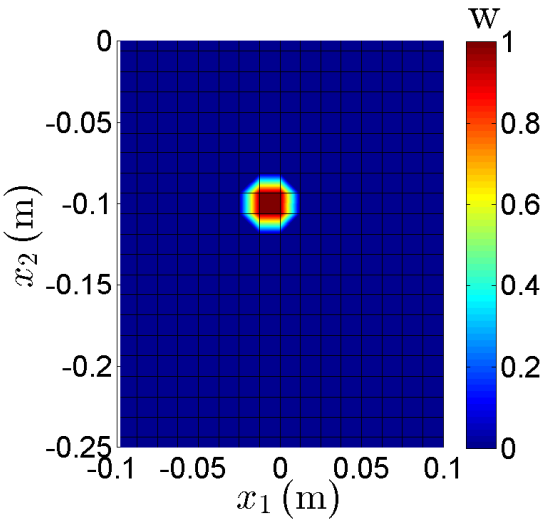
(b) Deformed interface at iteration $i = 25$

(c) Deformed interface at iteration $i = 50$

(d) Deformed interface at iteration $i = 112$

Figure 5.10: Numerical results from the CEL formulation for lateral compression of a circular cylinder. The cylinder is located between two rigid planes on the top and bottom that moved towards each other so that four fold symmetry is maintained. The planes are gradually moved to the final position shown in (d) until iteration $i = 80$ and then held in position. At iteration $i = 112$, the velocity in the domain vanishes (less than tolerance) so that the stress in the solid is at static equilibrium.

domain using finite element mesh size h = 0.0125 m on both directions. We prescribe initial damage at four nodes with $w = 0.99$ whereas the remaining nodes of the mesh have $w = 0$, as shown in Figure 5.11. Due to the applied velocity, $\mathbf{v} = 0.015\hat{e}_2$, the damage starts to flow in the -ve x_2 direction (i.e. downwards). Figure 5.12 shows damage being transported



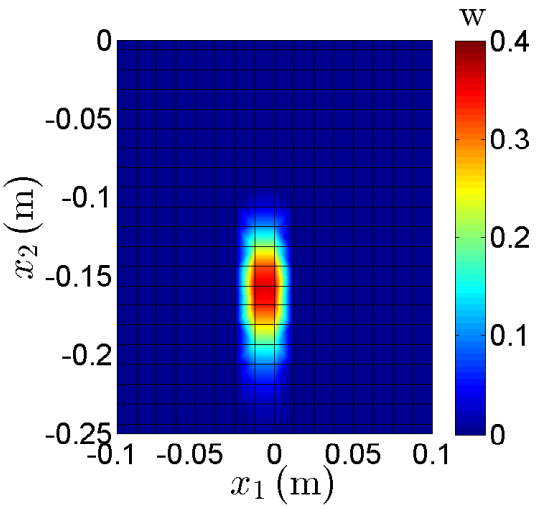
(a)

Figure 5.11: Hyperelastic solid with one completely damage element

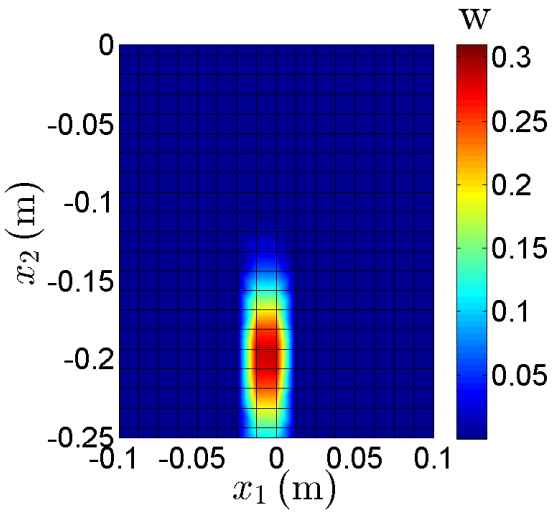
in the domain using the updated Lagrangian formulation. We observe the spreading of damage during the transport process. At time t=0, the damage is associated only with four nodes of a damage element but on subsequent iterations, when the damage flows through the Eulerian mesh, this damage spreads over more elements and thus the associated nodes. In updated Lagrangian framework, we make the current configuration of simulation as reference configuration for next iteration, this seems to cause the damage to keep spreading with the iteration until damage starts to flow out of the domain. It can be verified that the damage is spreading through artificial diffusion until it reaches the outlet (bottom edge of the domain) by calculating total damage in the solid at each iteration using

$$w = \int_{\Omega} w \, dV. \tag{5.8}$$

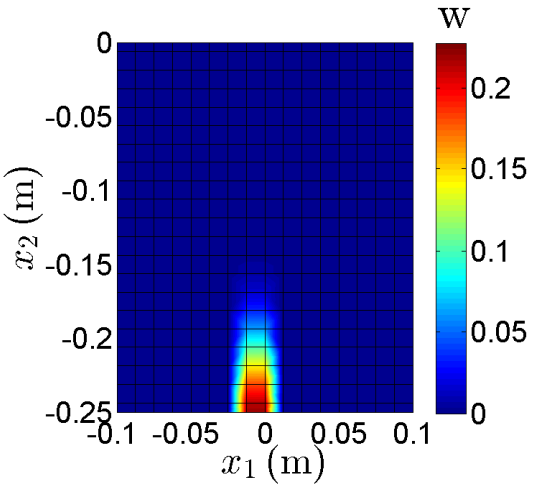
The total damage (w) in the solid remains constant until 75 iterations (see Figure 5.13)



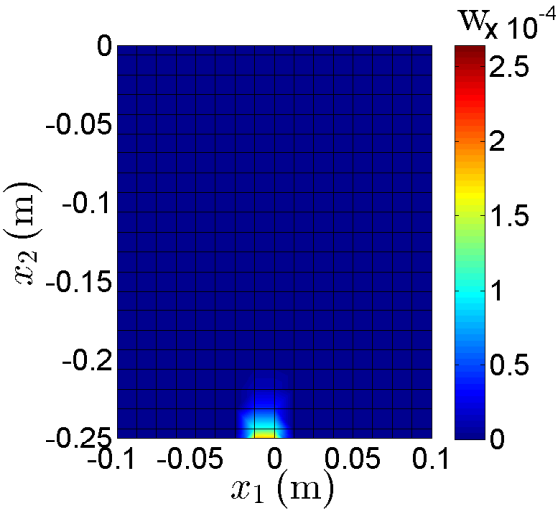
(a) Damage in the domain at iteration $i = 40$



(b) Damage in the domain at iteration $i = 60$

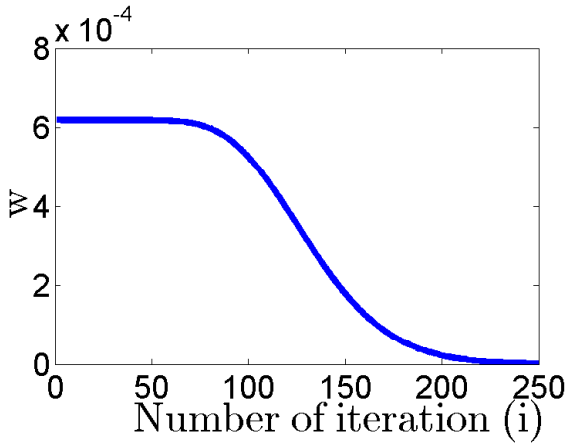


(c) Damage in the domain at iteration $i = 90$



(d) Damage in the domain at iteration $i = 190$, damage leaves the solid domain

Figure 5.12: Damage transport in the hyperelastic solid using the updated Lagrangian framework.

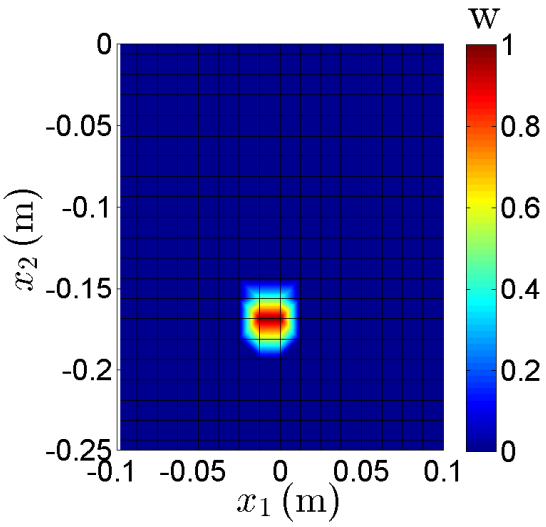


(a)

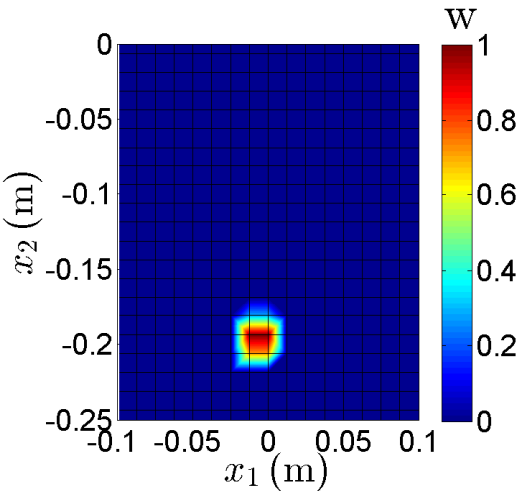
Figure 5.13: Total damage in the domain.

meaning that damage is being transported without any loss. Then after, damage starts moving out of the domain and the total damage value decreases until 200 iterations and ultimately comes to zero in about 240 iterations. At this point, all damage is transported out of the domain.

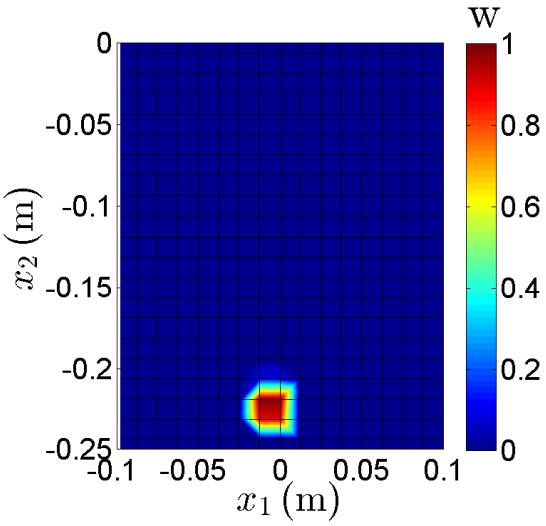
We carry out the same transport process using the total Lagrangian formulation and find that this formulation has less artificial diffusion of damage because the initial configuration is used as the reference configuration and at each pseudo time step or iteration, we back-track the position of particle to its initial configuration ($t = 0$) where only 4 nodes were damaged. Damage in total Lagrangian formulation spreads only within the peripherial 9 elements, which have at least one damage node. Next, we employ the Eulerian formulation as explained in section 3.7.2 for handling the same transport problem and find that it works better than the updated Lagrangian in checking the spread of damage. During the transport process, a thin trail is left as shown in Figure 5.15 but there is not much of spreading of damage. Our preliminary conclusion from this study is that the total Lagrangian mapping scheme is better in combination with the Eulerian solid mechanics formulation because artificial diffusion of damage is minimal.



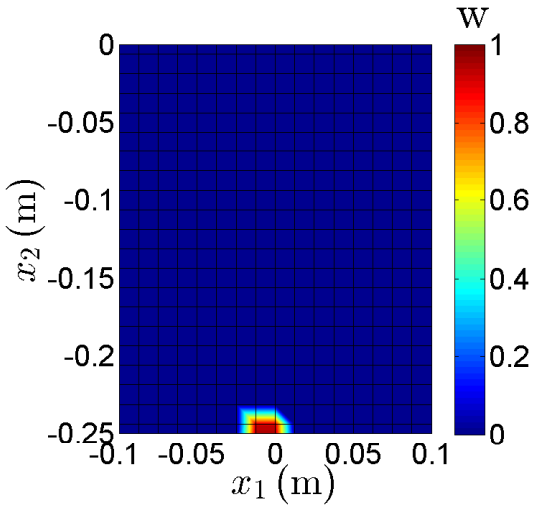
(a) Damage transported after iteration $i = 60$



(b) Damage transported after iteration $i = 80$

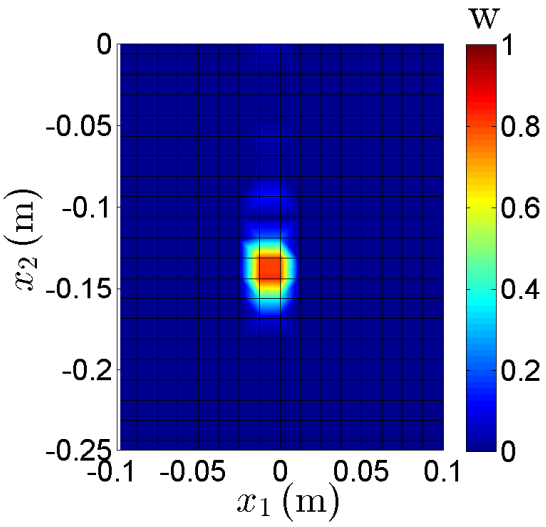


(c) Damage transported after iteration $i = 100$

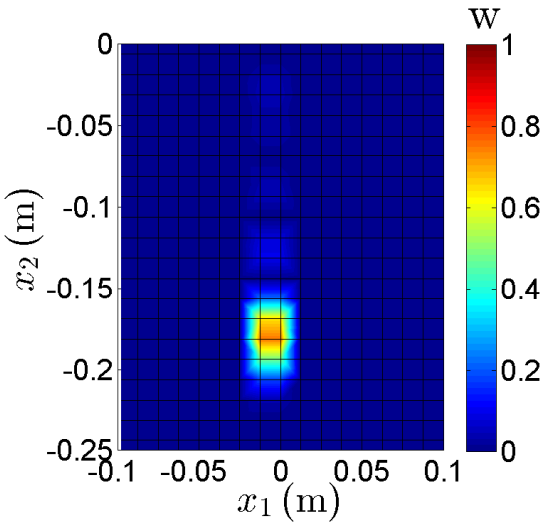


(d) Damage transported after iteration $i = 120$

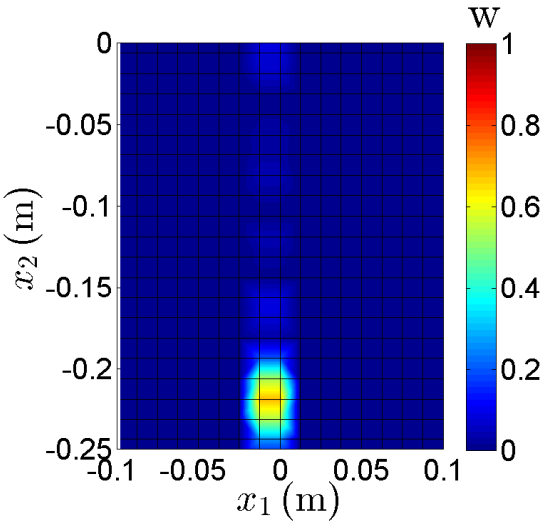
Figure 5.14: Damage transport using total Lagrangian framework



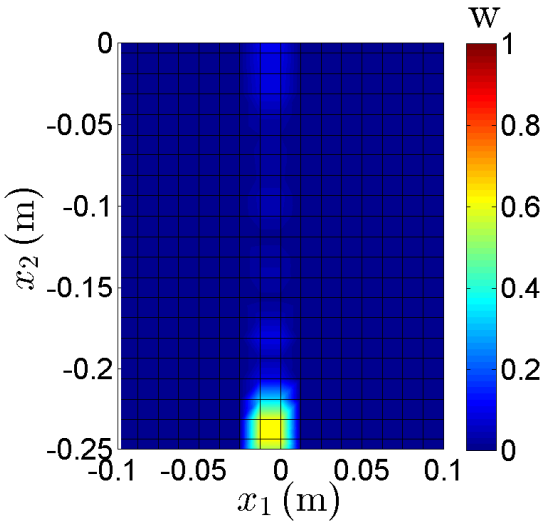
(a) Initial damage in the domain $i = 30$



(b) Deformed interface at iteration $i = 50$



(c) Deformed interface at iteration $i = 70$



(d) Deformed interface at iteration $i = 80$

Figure 5.15: Damage transport using the Eulerian framework

CHAPTER 6

Conclusion

With the given numerical examples, it can be concluded that coupled Eulerian-Lagrangian (CEL) formulation is a stable and convergent way for modeling large deformations of soft compressible hyperelastic materials and transporting damage within such materials . In the CEL formulation, the equilibrium equations are solved in an Eulerian framework and the transport equations of deformation gradient and Jacobian determinant are solved in an updated Lagrangian framework; thus, the strategy is opposite of that employed in an arbitrary Lagrangian-Eulerian (ALE) formulation, wherein the equilibrium equations are solved in a Lagrangian framework and the mesh or field variables are transported in an Eulerian framework. The mixed element formulation, although originally proposed in [2] for handling incompressibility, is observed to improve the accuracy of the numerical scheme even in the case of compressible media. The numerical results of uniaxial tension and simple shear studies agree well with theory indicating the accuracy and feasibility of the approach. The numerical study of indentation of a rounded rectangular block demonstrates the robustness of the implementation as compared to standard Lagrangian finite element implementation in Abaqus. Our preliminary results for damage transport indicate that the total Lagrangian formulation works better than the updated Lagrangian or the Eulerian formulation. The work presented in the thesis suggests the presented CEL formulation can be an attractive numerical approach when materials undergo extreme deformation and distortions such as that observed in very soft and visco-elastic media. The proposed approach can also be an interesting strategy for modeling fluid-structure interactions using a fully Eulerian framework for both fluid and solid mechanics; thus, it can be ideal for applications in biology [37, 38, 39, 40] (e.g. in cell mechanics and growth) or in studying the mechanics of soft-matter [41, 42, 43]. Moveover, damage transport scheme presented herein indicates the

viability of the CEL formulation for studying moving interface problems in hyperelastic materials with damage evolution.

As stated earlier in section 1.2.1, the shortcoming of the present Eulerian approach is the difficulty in applying the boundary conditions as the interface is not aligned with the mesh nodes. As a part of future work, we intend to employ Lagrange multipliers to apply the boundary condition on the interface rather than applying it at the nearest node. The damage transport schemes needs to be extended to model growth due to mechanical and/or chemical processes. Further development of this model can enable the modeling of propagation and advective transport of fractures in polar ice sheets, which is a direction for future work.

Appendix A

UHPYER source file

```
      subroutine uhyper(bi1,bi2,aj,u,ui1,ui2,ui3,temp,noel,
     .                  cmname,incmpflag,numstatev,statev,
     .                  numfieldv,fieldv,fieldvinc,numprops,
                        props)

     include 'aba_param.inc'

      character*8 cmname
      dimension ui1(3),ui2(6),ui3(6),statev(*),fieldv(*),
     .          fieldvinc(*),props(*)
c
      c10 = props(1)
      c01 = props(2)
      d1  = props(3)
c
      statev(1) = bi1
      statev(2) = bi2

      if (aj == 0.) then
      aj = 1.
      end if

      if (statev(1) == 0.) then
```

```
      statev(1) = 3.
      end if
c
      u = c10*(statev(1)-3.) + d1*log(aj)**2.
      ui1(1) = c10
      ui1(2) = 0. !c01
      ui1(3) = 2.*d1/aj*log(aj) ! 2./d1*(aj-1.)
      ui2(1) = 0.
      ui2(2) = 0.
      ui2(3) = 2.*d1/aj**2.*(1.-log(aj)) ! 2./d1
      ui2(4) = 0.
      ui2(5) = 0.
      ui2(6) = 0.
      ui3(1) = 0.
      ui3(2) = 0.
      ui3(3) = 0.
      ui3(4) = 0.
      ui3(5) = 0.
      ui3(6) = 2.*d1/aj**3.*(-3.+2.*log(aj)) ! 0.
c     Cauchy stress S22
      statev(3) = 1./aj*(2.*d1*log(aj)+4./3.*c10*statev(1)
     .      -4.*c10/aj**(2./3.))

c     Deformation gradient F22
      statev(4) = aj
      return
      end
```

Appendix B

Defining DLOAD and UTRACLOAD in Abaqus

```
      SUBROUTINE DLOAD(F,KSTEP,KINC,TIME,NOEL,NPT,LAYER
     .                 ,KSPT,COORDS, JLTYP,SNAME)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION TIME(2),COORDS(3)
      CHARACTER*80 SNAME
C
       F = 3*EXP(-(COORDS(1))**2/(2*0.25**2));
      RETURN
      END




      SUBROUTINE UTRACLOAD (ALPHA,T_USER, KSTEP, KINC,
     1                 TIME, NOEL,  NPT, COORDS, DIRCOS,
     1                 JLTYP, SNAME)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION T_USER(3), TIME(2), COORDS(3), DIRCOS(3,3)
```

```
      CHARACTER*80 SNAME
C
       ALPHA = 6*EXP(-(COORDS(1))**2/(2*0.25**2));
       T_USER(1)=0.0;
       T_USER(2)=-1.0;
       T_USER(3)=0.0;
      RETURN
      END
```

Appendix C

Input file 2-d rectangular solid

```
*Heading
** Job name: 2drectangularmodel Model name: Model-1
** Generated by: Abaqus/CAE 6.12-2
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Part-1
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Part-1-1, part=Part-1
*Node
      1,  -1.70000005,           0.
      2,  -1.49011612e-08,           0.
      3,  -2.98023224e-08,  0.800000012

  ......
```

```
    659,  -1.36568213, -0.530021012
    660, -0.753666282, -0.353020668
    661,  -1.09292388, -0.428057164
*Element, type=CPE4H
  1,  23,  24, 140, 171
  2, 157, 204, 205, 175
  3, 141,  56,   6, 157

  .....

612, 661, 626, 628, 629
613, 655, 127,  17, 545
614, 659, 601, 625, 647
*Nset, nset=Set-1, generate
   1,  661,    1
*Elset, elset=Set-1, generate
   1,  614,    1
** Section: Section-1
*Solid Section, elset=Set-1, material=Material-1
,
*End Instance
**
*Nset, nset=Set-5, instance=Part-1-1
   2,   3,  11,  34,  35,  36,  37,  38,  39,  40,  95,
     96,  97,  98, 99, 100,  101,
*Elset, elset=Set-5, instance=Part-1-1
   9,  10,  11,  25,  27,  49,  64,  66, 156, 185,  186,
```

```
  188, 190, 191, 203, 230, 330, 331, 333, 335, 365, 370,
  371, 375, 463, 492, 495,  502, 533, 535, 537, 539
*Nset, nset=Set-6, instance=Part-1-1
  1,  2,  7, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,  28,
  29, 30, 31, 32, 33, 60, 61, 62, 63, 64, 65, 66, 67, 68,
  69, 70, 71, 72, 73, 74, 75
*Elset, elset=Set-6, instance=Part-1-1
  1,    9,  16,  19,  20,  22,   23,  29, 31, 34, 38, 40,
  41,  50, 53,  76,   81, 161, 163, 164, 169, 170,   174,
  183, 186, 193, 194, 196, 201, 205, 211, 219, 238,  311,
  321, 322, 323, 324, 329, 330, 348, 349, 350, 358,  361,
  363, 364, 368, 466, 469, 470, 473, 479, 481, 482,  485,
  498, 500, 502, 503, 508, 509, 515, 521
*Elset, elset=_Surf-1_S2, internal, instance=Part-1-1
  14,  52,  54,  55,  68,  69, 155, 168, 173, 178,  195,
*Elset, elset=_Surf-1_S4, internal, instance=Part-1-1
  27,  28,  32,  67, 171, 181, 184
*Elset, elset=_Surf-1_S3, internal, instance=Part-1-1
 185,
*Surface, type=ELEMENT, name=Surf-1
_Surf-1_S2, S2
_Surf-1_S4, S4
_Surf-1_S3, S3
*End Assembly
**
** MATERIALS
**
```

```
*Material, name=Material-1
*Depvar
      4,
*Hyperelastic, user, type=COMPRESSIBLE, properties=3

2.8846, 0.0,  6.25

** ----------------------------------------------------------
**
** STEP: Step-1
**
*Step, name=Step-1, nlgeom=YES,  inc=10000
*Static
0.0001, 1., 1e-09, 0.1

**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation
*Boundary
Set-5, 1, 1
** Name: BC-2 Type: Displacement/Rotation
*Boundary
Set-6, 2, 2
**
** LOADS
**
```

```
** Name: Load-1   Type: Pressure
*Dsload
Surf-1, PNU, 3
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=ALL
SD3,SD4
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
```

BIBLIOGRAPHY

[1] D. Demarco and E. N. Dvorkin. An Eulerian finite element formulation for modelling stationary finite strain elastic deformation processes. *Intl. J. Numer. Methods in Engrg.*, 62(8):1038–1063, 2005.

[2] R. Duddu, L. L. Lavier, T. J. R. Hughes, and V. M. Calo. A finite strain Eulerian formulation for compressible and nearly incompressible hyper-elasticity using high-order B-spline finite elements. *Intl. J. Numer. Methods in Engrg.*, 89(6):762–785, 2012.

[3] S. Leung and H. Zhao. A grid based particle method for moving interface problems. *J. Comp. Physics*, 228(8):2993 – 3024, 2009.

[4] Erwin Stein, Rene de Borst, and T.J.R. Hughes. *Encyclopedia of Computational Mechanics*, volume 1. John Wiley and Sons, 2004.

[5] D.J. Benson. An efficient, accurate, simple ALE method for nonlinear finite element programs. *Comput. Methods Appl. Mech. Engrg.*, 72(3):305 – 350, 1989.

[6] P.A. Cundall. Numerical experiments on localization in frictional materials. *Ingenieur Archiv.*, 59(2):148–159, 1989.

[7] L.L. Lavier and W.R. Buck. Half graben versus large-offset low-angle normal fault: Importance of keeping cool during normal faulting. *Journal of Geophysical Research - Solid Earth*, 107(B6):2122, 2002.

[8] E.S. Choi, L. Lavier, and M. Gurnis. Thermomechanics of mid-ocean ridge segmentation. *Physics of Earth and Planetary Interiors*, 171(1-4):374–386, 2008.

[9] T.E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces - The deforming-spatial-domain/space-

time procedure: I. The concept and the preliminary numerical tests. *Comput. Methods Appl. Mech. Engrg.*, 94(3):339 – 351, 1992.

[10] T.E. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces - The deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Comput. Methods Appl. Mech. Engrg.*, 94(3):353 – 371, 1992.

[11] T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Comput. Methods Appl. Mech. Engrg.*, 29(3):329 – 349, 1981.

[12] R.B. Haber. A mixed Eulerian-Lagrangian displacement model for large-deformation analysis in solid mechanics. *Comput. Methods Appl. Mech. Engrg.*, 43(3):277 – 292, 1984.

[13] W.K. Liu, C. Herman, C. Jiun-Shyan, and T. Belytschko. Arbitrary Lagrangian-Eulerian Petrov-Galerkin finite elements for nonlinear continua. *Comput. Methods Appl. Mech. Engrg.*, 68(3):259 – 310, 1988.

[14] D.J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comput. Methods Appl. Mech. Engrg.*, 99(2-3):235 – 394, 1992.

[15] T. Yamada and F. Kikuchi. An arbitrary Lagrangian-Eulerian finite element method for incompressible hyperelasticity. *Comput. Methods Appl. Mech. Engrg.*, 102(2):149 – 177, 1993.

[16] A.A. Johnson and T.E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput. Methods Appl. Mech. Engrg.*, 119(1-2):73 – 94, 1994.

[17] Y. Bazilevs, K. Takizawa, and T.E. Tezduyar. *ALE and Space-Time Methods for Moving Boundaries and Interfaces*, pages 83–109. John Wiley and Sons, Ltd, 2013.

[18] K. Stein, T.E. Tezduyar, and R. Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *Journal of Applied Mechanics*, 70(1):58–63, 2003.

[19] S. Okazawa, K. Kashiyama, and Y. Kaneko. Eulerian formulation using stabilized finite element method for large deformation solid dynamics. *Intl. J. Numer. Methods in Engrg.*, 72(13):1544–1559, 2007.

[20] D.J. Hill, D. Pullin, M. Ortiz, and D. Meiron. An Eulerian hybrid WENO centered-difference solver for elastic-plastic solids. *J. Comp. Physics*, 229(24):9053 – 9072, 2010.

[21] T. Richter. A fully Eulerian formulation for fluid-structure-interaction problems. *J. Comp. Physics*, 233:227 – 240, 2013.

[22] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comp. Physics*, 79(1):12–49, November 1988.

[23] J. A. Sethian. A marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[24] E. Béchet, N. Moës, and B. Wohlmuth. A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method. *International Journal for Numerical Methods in Engineering*, 78(8):931–954, 2009.

[25] T.-Y. Kim, J. Dolbow, and T. Laursen. A mortared finite element method for frictional contact on arbitrary interfaces. *Comput. Mech.*, 39(3):223–235, 2007.

[26] J. Dolbow and I. Harari. An efficient finite element method for embedded interface problems. *Intl. J. Numer. Methods in Engrg.*, 78(2):229–252, 2009.

[27] C. Annavarapu, M. Hautefeuille, and J.E. Dolbow. A robust Nitsche's formulation for interface problems. *Computer Methods in Applied Mechanics and Engineering*, 225-228:44 – 54, 2012.

[28] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1969.

[29] J.C. Simo, R.L. Taylor, and K.S. Pister. Variational and projection methods for the volume constraint in finite deformation elasto-plasticity. *Comput. Methods Appl. Mech. Engrg.*, 51(1-3):177–208, 1985.

[30] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Springer, Berlin, 1998.

[31] S. Doll and K. Schweizerhof. On the development of volumetric strain energy functions. *Journal of Applied Mechanics*, 67(1):17–21, 2000.

[32] J. A. Sethian. *Level Set Methods & Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, Cambridge, UK, 1999.

[33] J. Glimm, J. W. Grove, X. L. Li, and D. C. Tan. Robust computational algorithms for dynamic interface tracking in three dimensions. *SIAM J. Sci. Comput.*, 21(6):2240–2256, 1999.

[34] R. Merle and J. Dolbow. Solving thermal and phase change problems with the extended finite element method. *Comp. Mech.*, 28(5):339–350, 2002.

[35] J. Lemaitre and J.L. Chaboche. Solving thermal and phase change problems with the extended finite element method. *Journal De Mechanique Appliquee*, 2(3):317–365, 1978.

[36] R. Duddu and H. Waisman. A temperature dependent creep damage model for polycrystalline ice. *Mechanics of Materials*, 46:23–41, 2012.

[37] F. J Vernerey and M. Farsad. A constrained mixture approach to mechano-sensing and force generation in contractile cells. *Journal of the mechanical behavior of biomedical materials*, 4(8):1683–99, November 2011.

[38] L. C Foucard and F. J Vernerey. Particle-based Moving Interface Method for the study of immersed soft vesicles. *IJNME (under review)*, pages 1–31, 2014.

[39] L. Foucard and F. J Vernerey. A thermodynamical model for stress-fiber organization in contractile cells. *Applied Physics Letters*, 100(1):13702–137024, 2012.

[40] F. J. Vernerey and M. Farsad. A mathematical model of the coupled mechanisms of cell adhesion, contraction and spreading. *Journal of Mathematical Biology*, 68(4):989–1022, 2014.

[41] P.-G. de Gennes, F. Brochard-Wyart, and David Quere. *Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves*. New York, Springer, 2004.

[42] W. Zhu, M. Shelley, and P. Palffy-Muhoray. Modeling and simulation of liquid-crystal elastomers. *Physical Review E*, 83(5):1–11, May 2011.

[43] F M Leslie. Continuum Mechanics Thermodynamlcs Continuum theory for nematic liquid crystals. *Continuum*, 4:167–175, 1992.