The University Of Sheffield.

GPU IMPLEMENTATION OF EXTENDED TOTAL LAGRANGIAN EXPLICIT (gpuXTLED) METHOD FOR SURGICAL INCISION APPLICATION

Author:

Idayat Salako (Miss)

(130263030)

Main Supervisor:

Dr Zeike Taylor

Co-supervisor:

Dr Jose Curiel-Sosa

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Philosophy

Department of Mechanical Engineering

The University of Sheffield

Nov. 8, 2020

Declaration

I, Idayat Salako, declare that I have produced this thesis entitled, " GPU Implementation of Extended Total Lagrangian Explicit (gpuXTLED) method for Surgical Incision Application" without the assistance of third parties and without making use of aids other than those specified. The thesis work was conducted from August 2014 to June 2019 under the supervision of Dr. Zeike Taylor and Dr. Jose Curiel-Sosa at the University of Sheffield (2014-2017)/University of Leeds (2018-2019) and Sheffield (2014-2019) respectively.

Sheffield, 8, Nov 2020

Acknowledgement

Without a doubt, it is not possible to journey through PhD without exposing all your weaknesses and strength (mental, emotional, physical, spiritual). But, for the one who reaches the first destination of submitting a thesis, whether you obtain a degree or not, congratulations for persevering: for not remaining sulking, wishing to be like successful predecessors before you. For if you have come this far, then know once and for all that you had all the abilities to acquire the necessary skills (perseverance, seeking, smart work), you just did not know it or believe in it.

So, all praise is to the One by whose decree a seed sprouts into a plant by the splitting of the earth. To my parents, family, sisters from another parent, friends, everyone that has contributed in one way or another to my pursuit of this research degree, and to my supervisors, Dr Zeike Taylor, Dr Jose Curiel-Sosa and Professor Alejandro Frangi for their support over the years. My gratitude also goes to FfWG for sponsoring my final year.

Abstract

An extended total Lagrangian explicit dynamic (XTLED) is presented as a potential numerical method for simulating interactive or physics-based surgical incisions of soft tissues. The simulation of surgical incision is vital to the integrity of virtual reality simulators that are used for immersive surgical training. However, most existing numerical methods either compromise on computational speed for accuracy or vice versa. This is due to the challenge of modelling nonlinear behaviour of soft tissues, incorporating incision and subsequently updating topology to account for the incision. To tackle these challenges, XTLED method which combines the extended finite element method (XFEM) using total Lagrangian formulation with explicit time integration method was developed. The algorithm was developed and deformations of 3D geometries under tension, were simulated. An attempt was made to validate the XTLED method using silicon samples with different incision configuration and a comparison was made between XTLED and FEM. Results show that XTLED could potentially be used to simulate interactive soft tissue incision. However, further quantitative verification and validation are required. In addition, numerical analyses conducted show that solutions may not be obtainable due to simulation errors. However, it is unclear whether these errors are inherent in the XTLED method or the algorithm created for the XTLED method in this thesis.

Contents

List of Figures

List of Tables

List of Abbreviation

VRS	Virtual Reality Simulator
MIS	Minimally Invasive Surgery
MSM	Mass Spring Model
FEM	Finite Element Method
TLED	Total Lagrangian Explicit Dynamics
XFEM	Extended Finite Element Method

1 INTRODUCTION

Simulators have been used for training in automobile, aerospace, and medicine. The use of simulators in these fields enables driving, flying, and surgery respectively, to be modelled as virtual tasks [1]. This computation alleviates cost, risk to life and it enables easier transition into real life training. However, the challenge to create a realistic training experience is still ongoing.

One of the applications of simulators in the surgical field is surgical training. Simulators known as virtual reality simulators (VRS) were developed due to the impact of minimally invasive surgery (MIS) on patient recovery. Unlike open surgery technique which could require larger incisions to physically access the area of operation, MIS technique only requires small incisions of about 0.5cm to 1.5 cm, through which surgical instruments are inserted [2], [3]. Due to the small incisions, patients are less likely to lose excess blood during operation or require blood transfusion, patient's hospital stay is reduced, surgical scars are smaller and aesthetically pleasing, and patient recovery is quick [2], [3].

However, MIS comes with its own challenges that surgeons must overcome. In MIS, a surgeon performs operation by manipulating long slender surgical tools while viewing the site of operation on a display screen. The viewing of the site of operation is made possible by endoscopes inserted into the patient's body via incision ports (trocars) (Figure 1-1). The lack of direct view and access to the area of operation, as present in open surgery, and constrained movement of the surgical tools inserted in trocars, results in restricted view of the operating area, and makes it less easy for the surgeons to manipulate surgical tools whilst coordinating both the hands and eyes during operation. Consequently, virtual reality simulators (VRS), similar in purpose

to flight simulators, were proposed to enable repetitive training and help surgeons master MIS skill by overcoming the challenge of coordinating the hands and eyes during surgery.

Figure 1-1: Minimally invasive surgical set up, showing a surgeon, surgical instruments inserted in incision ports (trocar), and display screen [4].

The VRS mainly consists of a haptic device, surgical instruments, a display screen and computational model of the operated organ or site of operation or patient (Figure 1-2). The haptic device makes the VRS especially unique, because it is a feedback system that allows a surgeon to provide input (touch, pull etc.) and receive output from the computational model. Prior to this technological advancement, surgeons would mainly train on cadavers, humans, animals, and other training alternatives like box trainers, torso, human-like mannequin. However, to reduce the risk to patient life, enable realistic and repetitive training, and due to the difference in anatomy between humans and animals, VRS was proposed. Consequently, commercial and non-commercial attempts have been made to develop virtual reality simulators for different surgeries. For example, the first commercial cholecystectomy simulator called MIST-VR (Minimally Invasive Surgery Trainer – Virtual Reality) was developed by Wolfson Centre for Minimally Invasive Therapy (Manchester) in

collaboration with the company called VR solution; and a cataract simulator was developed for removing eye cataract [1], [2]. Moreover, due to the benefits of MIS, surgical societies such as the Society of American Gastrointestinal and Endoscopic Surgeons (SAGE) have incorporated structured MIS training into the curricula at medical schools and some hospitals across Europe now have dedicated VRS training centres [1], [5]. The general feedback is that VRS can be used to improve basic MIS skills of surgeons, including suturing, cutting, probing, endoscope manipulation and can even be used to model anomalies [6].

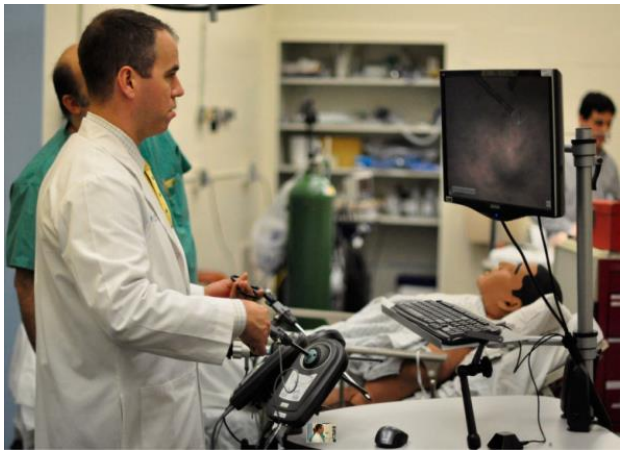

Figure 1-2: Surgical training using a virtual reality simulator [7]

Of the basic skills that could be acquired using VRS, cutting remains an area of active research. This challenge persists due to the need to simulate interactive and immersive surgical training in real time, using a realistic model. To achieve this using the haptic device, haptic feedback should occur at a frequency of at least 1kHz [8].

Physics-based or continuum approach (mesh and meshless numerical method) has been established as the ideal approach for computing the deformation of objects, and many linear and nonlinear deformation of soft tissues have been reported over the years [8], [9]. However, cutting adds another layer of complexity, since it involves the

separation of a surface, propagation of the cut and topological update of the computational and geometric model. This added complexity makes it difficult to achieve the required haptic feedback at 1kHz; in mesh based approaches, computational speed decreases with increased degree of freedom, as more elements are created due to cutting, and elements must be aligned to cut, whereas in meshless approaches which do not require elements, other challenges such as the creation of a cut surface arise [8], [9]. Furthermore, nonlinear analysis which captures soft tissue behaviour should be used to replicate real life incision-induced deformation of soft tissues, if VRS is to become a high-fidelity simulator in future.

These challenge and benefits of VRS is what motivated the investigation of the simulation of surgical cutting of soft tissues, using realistic soft tissue behaviour without compromising computational speed.

1.1 Aim and Objective

The aim of this research is to develop a numerical method suitable for realistic and real time surgical training. The objective to achieve this aim is to:

1. develop algorithms for the XTLED numerical method for arbitrary, cut-induced deformation of soft tissues, without manually conforming the incision to finite element faces.

The following chapters of this thesis are arranged as follows. A review of the state of the art in the simulation of surgical cutting is presented in chapter 2. Chapter 3 and 4 present background details on continuum mechanics and total Lagrangian explicit dynamics method. Chapter 5 presents the chosen numerical method, the extended total

Lagrangian explicit dynamics method (TLED), its verification and validation; and chapter 6 concludes this thesis.

2 SIMULATION OF SURGICAL INCISION

The main area of focus in this thesis is the simulation of the soft tissue cutting. To investigate this, existing work on soft tissue behaviour (simple, linear, nonlinear), authors' justifications for such choices and a brief review of the acquisition of soft tissue model will be presented. Secondly, various types of cut models according to physics and non-physics-based numerical approaches will be presented, along with a review of the representation of cuts for soft tissue cutting. Finally, existing approaches to accelerate numerical methods and achieve a haptic rate of 1kHz for cut-induced soft tissue deformation is reviewed. For a complete picture, a brief overview of VRS is presented first.

2.1 VRS Overview

In its simplest form, the VRS consists of a haptic device via which a user can interact with a computational tissue model, a display screen to visualise the interaction, and underlying software which run on a computing system (Figure 2-1).

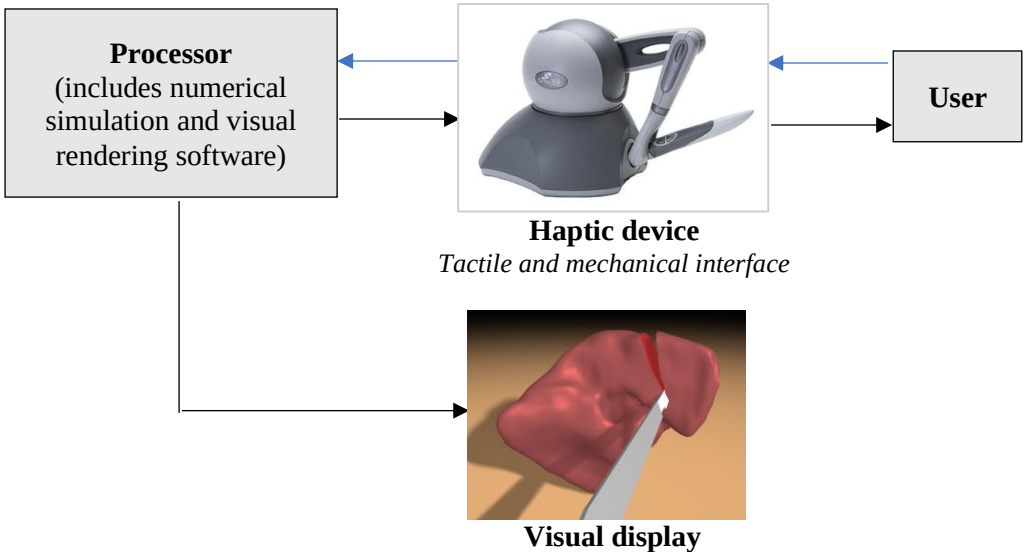

Figure 2-1: Components of a virtual reality simulator: the haptic device, display screen and underlying software [8], [10]

2.2 Soft Tissue Model for Realistic Simulation

2.2.1 Acquisition of soft tissue model

Accurate anatomic geometries of soft tissues are required for accurate simulation of soft tissues deformation. Such representation makes VRS useful for surgical planning and image guided therapy, besides surgical training, and are obtained from medical images via techniques such as X-ray, Computed tomography (CT) or Magnetic Resonance Imaging (MRI) [9], [11]. From these images, the desired tissue is segmented and spatially discretized into a grid-like structure called mesh [12], [13]. Once a geometry is obtained, realistic tissue model is required for tissue simulation, as is further discussed below.

2.2.2 Modelling Tissue Behaviour

There are three categories of soft tissues models in literature: heuristic, continuum, and data - driven models [6], [9]. Heuristic models include purely geometric representations (Figure 2-2), mass spring model (MSM) (Figure 2-3) and surface-based models (Figure 2-4). In geometric representation such as free form deformation, external control points placed on the geometry can be manipulated to give a desired deformation; MSM is a spring-based model where springs are arranged in a specific orientation to obtain a desired stiffness; and in surface based models such as Chainmail geometric restrictions dictates deformation.

Figure 2-2: Free form deformation (FFD) showing geometry with control points [9]

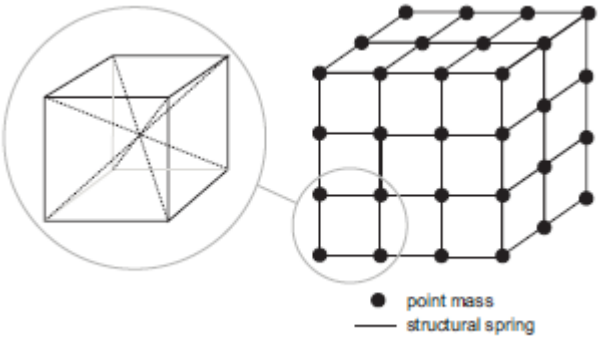

Figure 2-3: Mass-spring model showing point mass and springs [9]

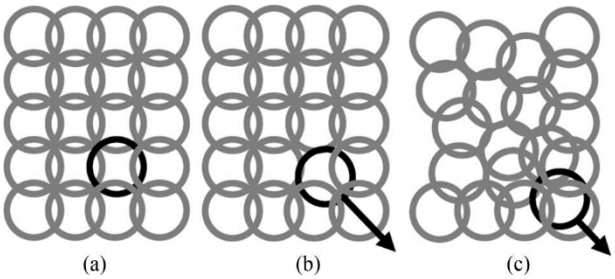

Figure 2-4: Chainmail model: (a) chainmail with chain elements (b) black chain element is moved along the black arrow (c) other chain elements bounded to the black element deform accordingly [9]

Researchers including Delingette et al., Bielser et al. and Misra et al. implemented heuristic models due to its simplicity and computational efficiency [14], [15], and [6]. However, such models do not realistically capture the behaviour of soft

tissues which consist of elastin, collagen fibres and water, and thus are nonlinear, anisotropic, nearly incompressible, time dependent, rate dependent, and an heterogenous material that undergo large deformation [6], [14],[16]. For instance, the structural behaviour of MSM, which has been extensively researched, is heavily dependent on the spring length, orientation, and stiffness. These dependencies make it difficult to relate constitutive laws to spring mass models and are unideal for topological changes. Thus, despite the simplicity of these methods, they cannot be used to depict real deformation of soft tissue.

A more realistic model is the continuum-based model, otherwise referred to in literature as physics-based models. This class of material models are mathematical models that characterise tissue behaviour based on the continuum mechanics theory. The theory assumes that a structure is a continuous medium and characterises its behaviour using parameters obtained from experimental analysis. For instance, linear structures are characterised using parameters such as Young's modulus and Poisson ratio [6].

The simplest continuum-based model is the linear elastic model. Like the heuristic model, linear elastic model has been used by many researchers for soft tissue deformation. However, due to its underlying principle that stress is computed directly from the strain and Young's modulus, the method is only ideal for small deformation analysis where elastic deformation is typically 1% - 2% of material or less than 10% of mesh size [17] .In other words, the deformation of the soft tissue should be between 1% - 2% of its dimension or less than 10% of the mesh dimension. Thus, linear elastic models cannot be used to capture the large, time dependent, and rate dependent deformations present in soft tissues [17].

Alternative continuum-based models are the hyperelastic and visco-elastic nonlinear models. Some researchers have adopted one of these models or a hybrid, to simulate the behaviour of soft tissues. For example, Ogden hyperelastic model was used to model incompressible nonlinear deformation; a hybrid of Mooney Rivlin hyperelastic model and 2-term Prony series viscoelastic model were used to capture porcine liver nonlinearity [8], [9]; and Mooney Rivlin model and exponential relaxation parameter were used to model non-linear and anisotropy behaviour of femur medial collateral ligament tibia [17]. These models fit differently to different soft tissues, and are ideal for large deformation, but are computationally expensive [6], [8], [9], [11], [17]. This is because unlike linear models, stresses for these models are obtained by differentiating energy functions [8], [16]. Thus, the widespread use of nonlinear models for soft tissue cutting in VRS application has been hindered by computational cost.

The final category which includes neural network, machine learning and shape modelling, consists of a set of methods that aim to attain computational speed or/and enable the use of patient specific data for real time applications [9], [18]. While these class of models may allow for differences in patient anatomy and physiology, the challenge of using such models lie in difficulty of obtaining patient specific data.

A summary of the classes of soft tissue material model is given in Table 2-1. Interested readers can refer to [9].

Table 2-1: Soft tissue material models for simulating cut

Tissue model	Pro	Con
Heuristic models		
• Free form deformation • Mass-spring model • Chainmail	• Simple • Easy computational implementation	• Discrete representation of a continuum model • Difficult to relate spring parameters to that of soft tissues (MSM) • Unsuitable for non-linear deformation due to difficulty of determining a suitable spring assembly (MSM)
Physics-based model (Constitutive models)		
• Linear	• Easy to implement	• Inadequate representation of soft tissue behaviour
• Nonlinear	• A better approximation of material behaviour compared to linear model • Accounts for creep and relaxation (viscoelastic)	• Does not account for time dependent deformation (hyperelastic) • Unideal for real time application due to effect on computational time
Other		
• Data-driven approach • Machine learning • Neural Network	• Encourages the use of patient specific • Real time computational advantage • Minimum error (machine learning)	• Valid data may be difficult to obtain (data-driven approach) • Machine learning depends on training data and learning algorithm

Another factor that influences the accuracy of the result obtained in soft tissue modelling, besides the geometry of a soft tissue and its behaviour, is the boundary conditions imposed by contiguous tissues [6], [19]. Moreover, to realistically simulate tissue deformation, the effect of neighbouring tissues should be considered. A research that has considered the effect of neighbouring tissues and friction is Courtecuisse et al. [19]. In the article, liver deformation was simulated with the effects of adjoining organs (stomach, colon, intestine and diaphragm). Very little research exists about the effect of neighbouring tissue and friction, due to the difficulty of obtaining in vivo data [6], [9].

2.3 Numerical Method for Simulating Surgical Incision

The type of problem under consideration dictates the equilibrium equation that is solved. For structural problems such as the deformation of a soft tissue, the mathematically model used is as elliptical equation written as partial differential equations (PDE) and governed by boundary conditions [20]. A numerical approach is required to solve or discretise this partial differential equation that governs the structural response of a deformable body (Figure 2-5). Traditionally, existing numerical methods for simulating soft tissue incisions are categorised as mesh and meshless methods, based on the technique employed to discretise the computational simulation domain [21]. Mesh methods allude to a grid-like spatial domain discretization where different elements may be used, while meshless methods describe a domain discretised with mass points (Figure 2-5). Different approaches are adopted for simulating soft tissue incision using either method or a hybrid. The following section looks at mesh-based and meshless method and the associated cutting techniques adopted for simulating surgical incision.

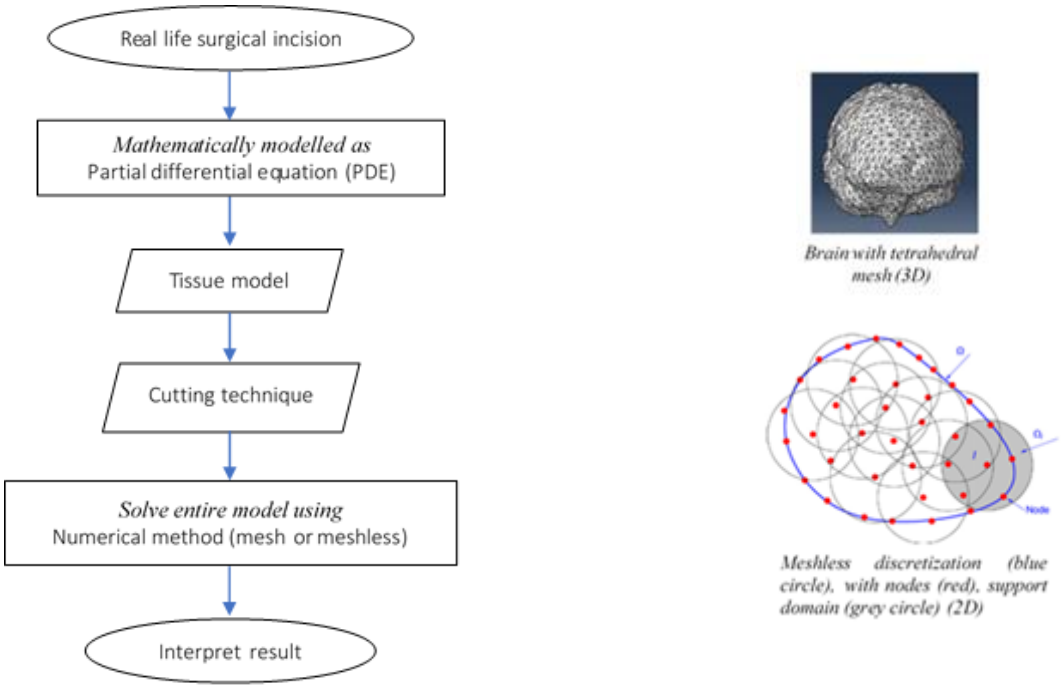

Figure 2-5: Stages of simulating incision induced soft tissue deformation (left); examples of three dimensional mesh-based discretization (right, top), and two-dimensional meshless discretization (right, bottom [22])

2.3.1 Mesh Based Methods

Of the mesh based methods, finite element method (FEM) is the most commonly used method in the field of surgical simulation, due to its accuracy in estimating structural response [8], [23]. Other methods such as boundary element method (BEM) are scarcely used since, for instance, BEM is only ideal for small deformation and its surface-only-discretization approach is unsuitable for volumetric topological changes, which exists during tissue incision [24], [25]. FEM is categorised into linear, corotational or nonlinear. Regardless of the type of analysis, the computational domain (geometry) is subdivided into a mesh of elements interconnected at nodes, and the deformation of the structure is computed from the

displacements at the nodes using shape functions. These shape functions could be polynomials of any order, depending on the dimension of the problem. An approximation of the unknown variable is subsequently obtained by refining the mesh whilst concurrently ensuring that the generated elements are of good quality, that continuity is maintained within elements and that there are no gaps between neighbouring elements during deformation. Thus, solution accuracy of FEM (and all mesh-based methods) is determined by the shape function (element type) and element quality.

To model interactive soft tissue incision using the FEM, cutting tool representation, cutting techniques and solution of deformation using all types of FEM analysis previously mentioned must be considered, and is presented next.

2.3.1.1 Cutting Tool Representation in FEM

Knowledge of cutting tool representation is required to simulate collision or tool-tissue interaction for any chosen cutting technique. This representation varies depending on the type of spatial discretization adopted for the computational domain of the deformable body. In mesh-based methods where tetrahedral, hexahedral or polyhedral are used, the cut is usually represented as triangles or links. For a tetrahedral or polyhedral mesh representation, the cut representation is easily extracted from the element surfaces of the deformable computational domain. On the other hand, surface reconstruction techniques like splitting cube or dual contouring algorithms are adopted to create a smooth cut representation due to the jagged surface effect of hexahedral mesh [8], [9]. Once this is determined, various individual or hybrid cutting techniques may be used.

2.3.1.2 Cutting Techniques in FEM

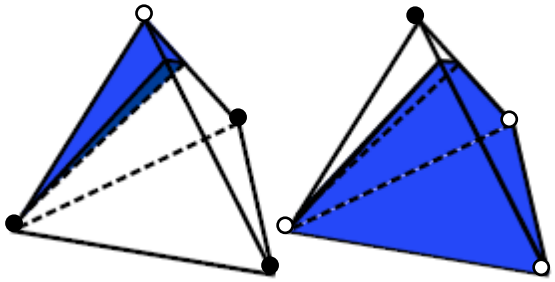

Figure 2-6: Partially empty replicas of a cut element; white circles indicate the virtual nodes [26]

Incisions have mainly been simulated with tetrahedral representations of a deformable body because it is easily adaptable to complex geometry due to its shape, in comparison to other elements. Consequently, a few methods have been proposed for soft tissue cutting using tetrahedral over the years [8], [27]. Of these cutting techniques, element duplication (or virtual node method) seems the most suitable for achieving arbitrary cutting, since it avoids numerical instability due to ill-shaped or silver elements and is applicable to different element types. Molino et al. originally proposed this method with tetrahedral and shell elements [26]. The idea was to create replicas of the cut element with the same nodal connectivity as the original element (Figure 2-6), compute deformation using the well-shaped replica, and to interpolate mid-nodes displacements from element nodes. The advantage of this method lies in the lack of remeshing, since remeshing could create of silver elements. Consequently, element duplication method eliminates instability issues since only well-shaped elements are used to compute the deformation response of the body. In other words, the resolution of the mesh and deformable geometry are separated. Furthermore, the researchers who proposed this method stated that the approach is suitable for multiple cuts and the algorithm can handle partially and fully cut elements. However, it is

unknowns whether the approach is similarly suited for the incision of an already cut elements, and whether cases of silver elements may arise. Furthermore, the method is limited by the fact that there must be at least at least one virtual node in the replica geometry. To eliminate this limitation, Sifakis et al. proposed a purely virtual node element algorithm to enable arbitrary number of element replicas [28] .

Prior to element duplication, other methods including element deletion, element refinement, splitting along existing element faces, node snapping, element refinement and node snapping had been proposed (Figure 2-7).

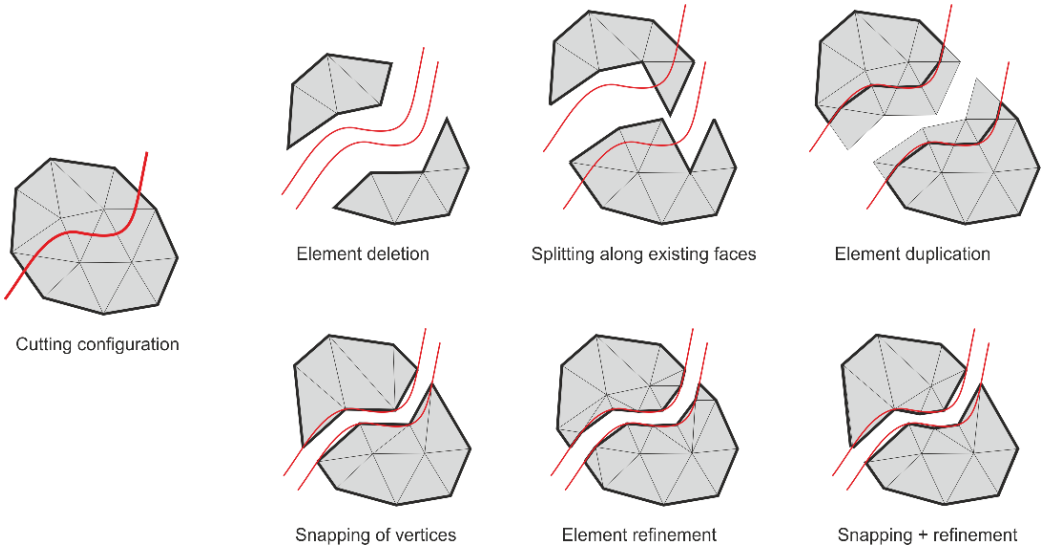

Figure 2-7: Cutting techniques presented in 2D [8]

Element deletion was one of the first known cutting techniques proposed and adopted for simulating incisions [27], [29]. In this method, elements intersected by a cutting tool are deleted. While this method is simple and does not lead to the creation of new elements, it violates mass conservation law and is an inaccurate approximation. Similar in consequence to element deletion is the technique of cutting along existing element face. Here, because the original number of elements prior to cutting is maintained, the technique hinders arbitrary cutting and is only ideal if the cut path is

known prior to simulation. A fourth technique is element remeshing. Similar to element deletion, element remeshing had been used by many researchers at least between 1999 and 2001 [27]. However, while element remeshing eliminates the inaccuracy due to loss of mass and volume, the number of elements generated due to this technique increases the computational time for computing deformation. Hence, achieving real time simulation via element remeshing seems farfetched, and there is also the possibility of generating ill-shaped elements.

A common hinderance in the above cutting techniques is the creation of ill shaped elements. Researchers such as Bielser et al. proposed strategies for tetrahedron decomposition to prevent creating poor quality elements [15], [30]. Five topological configurations were created as a result; two complete partition of a tetrahedron (III, IV) and three partial partitions (Figure 2-8, left). Steinemann et al. enhanced the method by providing an algorithm that enabled cutting along node, adding three extra configurations to those of Bielser et al. [31] (Figure 2-8, right). Other attempts to reduce the number of elements generated as a result of element refinement include progressive cutting and multiresolution approach [32], [33]. While all these methods do work, the creation of new elements does not eliminate the possibility of creating ill shaped elements.

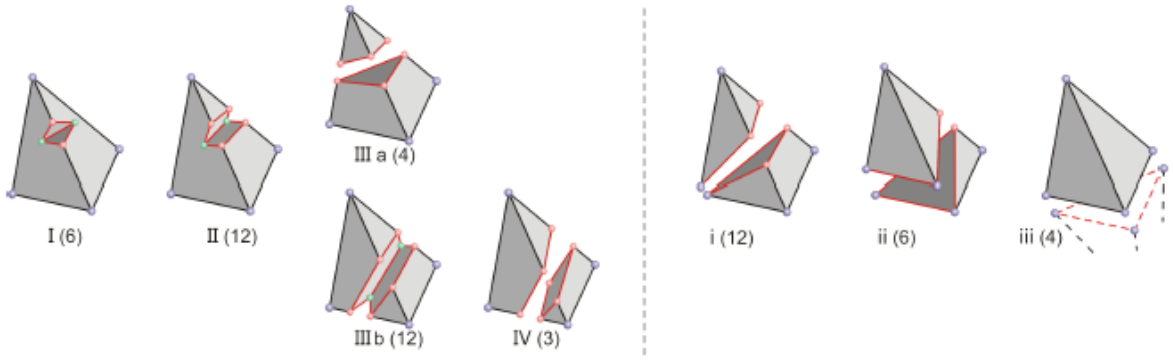

Figure 2-8: Partial and full topological configuration as proposed by [15], [30]

A final category of cutting technique aimed at enabling arbitrary cut paths and eliminating the creation of poor-quality elements are the node snapping and a hybrid of node snapping and element refinement [34]. In the former method, nodes are brought onto the cutting path if it is considered too close to any element node based on a distance criterion. Consequently, this method reduces the number of new elements generated. However, there is still a possibility of creating ill-shaped elements which could lead to the removal of such element, thus, leading to an inaccurate solution [35]. A summary of all cutting techniques is given in Table 2-2 below.

Table 2-2: Comparison between cutting techniques

	Element deletion	Element refinement	Node snapping	Element duplication
Pro	No creation of extra elements during cutting	Enables arbitrary cutting	Enables arbitrary cutting	Enables arbitrary cutting
Cons	Does not fulfil mass conservation law	Creation of new elements after cutting	Creation of ill-shaped elements	Unknown if ill-shaped elements are generated

In hexahedral mesh, linked volume representation and adaptive octree grid were used to simulate cuts [8]. Although, the hexahedral is not a simple shaped element like the tetrahedral, it is a higher order element, and this allows for faster

convergence [8]. However, both tetrahedral and hexahedral elements behave overly stiff, for nearly incompressible material such as a soft tissues, when the elements are fully integrated – this can be overcome using under-integrated elements, where one Gauss point is used for integration [36], [37]. This phenomenon of over stiffening of fully integrated elements is referred to as volumetric locking.

Linked representation of cut was proposed by Frisken-Gibson [38]. Here, the deformable body is discretized into uniform hexahedral grid with adjacent element faces connected by links: six links per hexahedral. These links are disconnected when intersected by the virtual cutting tool and information of the point of intersection on the link and the normal to the tool is stored to create surface vertices on the affected cells (that is the cells containing either ends of the link). Subsequently, a surface is created using two surface patches: 2 by 2 triangles per each cut link.

An alternative cutting technique in hexahedral mesh is the use of adaptive octree grid proposed by Wu et al. [8]. Here, the majorly coarse hexahedral elements are further refined along the cut, whilst maintaining a smooth transition between elements that share edges, vertices or faces.

Polyhedral mesh has also been used for numerical discretization by Martin et al. [39]. Polyhedral allows the flexibility of generating varied number of n-sides elements after cutting, unlike the tetrahedral and hexahedral mesh where same-sided elements must always be present in the mesh. As a result, polyhedral cutting creates reduces number of new elements after incision, compared to previously mentioned methods. However, polyhedral discretization is affected by two drawbacks. These drawbacks are lack of clarity about element quality of polyhedral, and the possibility

of creating of ill shaped elements. As with any mesh-based method, element quality is essential for computational accuracy.

A comparison of the tetrahedral, hexahedral and polyhedral discretization elements is given in Table 2-3.

Table 2-3: Comparisons of incisions in tetrahedral, hexahedral and polyhedral mesh

	Tetrahedral	Hexahedral	Polyhedral
Pros	• Ideal for discretizing complex geometries. • Cutting tool surface is easily represented by inspecting the 3D geometry that is to be cut.	• A higher order element than the tetrahedron element.	• Higher order element than a tetrahedron or hexahedron. • Less elements are generated. • Easy generation of cutting tool surface. • No restriction of elements generated during cutting.
Cons	• Ill shaped elements lead to numerical instability	• Computationally more expensive due to more nodes. • Requires a technique to generate smooth cutting tool surfaces	• May create ill shaped elements which affects stability

2.3.1.3 FEM Simulation of Surgical Incision

A time-dependent partial differential equation or governing equilibrium equation must be solved to simulate incision (Figure 2-5). The discretization of this equation in FEM, using shapes functions (which translates to elements type used for spatial discretization) and time integration methods leads to a system of equations which is be solved at every time step. The numerical finite element discretization could be linear, co-rotational or nonlinear FEM. As for the time integration, this could be implicit or explicit time integration methods.

The simplest FEM is the linear elastic model which assumes that the material undergoes small displacement and strain, and for which the linearized Green strain

(Cauchy strain) is used. Many researchers have implemented the linear elastic FEM due to its simplicity and computational efficiency. For example, Song and Reddy simulated the cutting of 2D object using linear elastic FEM [40]. Cotin et al. used 3D quasi-linear elastic FEM model of a liver and also incorporated force feedback and contact algorithm [41]. Update frequencies of 500Hz and 30Hz were reported for force and visual updates, respectively. However, linear FEM cannot accurately model non-linear behaviour and a consequence of such simplification is the ghost forces phenomenon, which is due to the use of rotation variant Cauchy strain that results in a structure expanding unreasonable during large deformation (Figure 2-9).

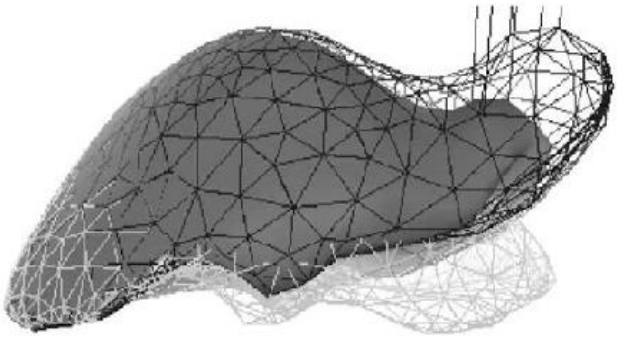

Figure 2-9: Deformation of a liver using linear elastic model (wire fire) and nonlinear elastic model (solid) [17]

However, the co-rotational method proposed by Müller et al. solves the issue by combining linear elastic FEM with Cauchy strain [42]. Here, the reaction force is computed with the elements aligned to their reference (undeformed) configuration and then rotated back into the deformed configuration. Hence, the choice of the rotation technique heavily affects the solution accuracy.

Non-linear behaviour in FEM is categorised into contact, material and geometric nonlinearity. In the context of surgical simulation, accurate real time tool-

tissue interaction falls under contact while complex or irregular shape of the soft tissues fall under material and geometric non-linearity. Some attempts to model non-linear tissue behaviour include the work of Taylor et al., where Neo-Hookean hyperelastic model with explicit time integration method were implemented to model brain shift [43]; Ghali and Sirouspour implemented Ogden's constitutive model for material and geometry non-linearity [44]; Picinbono et al. simulated surgical gestures in laparoscopy using anisotropic, non-linear St. Venant-Kirchoff constitutive model of liver in FEM with explicit dynamic method [17]. An updating frequency of 20Hz was reported in the latter.

2.3.2 Meshfree Methods

Meshless methods have been adopted to avoid the dependency of solution accuracy on quality element, as present in mesh-based methods. Even after generating quality mesh for a complex soft tissue, it is still possible to produce highly distorted elements (i.e. zero or negative Jacobian) during simulations, due to large deformation of soft tissues, thus invalidating the solution [9].

In meshless methods, the problem domain is discretized by disconnected arbitrary nodes and with the displacement at any point of interest is approximated using interpolation functions over a support domain consisting of neighbouring nodes, (Figure 2-5) [8], [9], [22]. This support domain is usually rectangular or circular. The choice of interpolation function in meshless method influences correct and easy imposition of boundary conditions, which are essential to creating incision (Table 2-4). Unlike finite elements, whose shape function possesses all qualities stated in Table 2-4, interpolation functions used for some meshless methods do not possess all these qualities. For example, the moving least squares interpolation function (MLS)

used in Element free Galerkin method (EFG) does not possess the Kronecker delta property necessary for easy implementation of boundary conditions. Furthermore, specifically for simulating incisions, the benefits of arbitrary nodes with no explicit connectivity comes with the drawback that node-to-node adjacency must be computed, stored and updated for every simulation step, thereby increasing computational time [8].

Some of the most prominent methods in virtual surgical simulation are hereby presented with the cutting techniques used for such methods.

Table 2-4: Shape function properties and implications

Shape function properties	Description and implications
Consistency or reproducibility	This is the ability to reproduce a certain order of polynomial in all support domains across the problem domain. Required for passing the standard patch test
Compatibility	Ensures the solution of adjacent elements are the same and hence there are no false gaps
Kronecker delta	$u(x) = u_i$ Ensure correct imposition of boundary constraint. That is the displacement constraint applied at any point of interest equals the nodal displacement at that point. Implies that the former can be specified by fixing the latter. All this can only happen when the sum of shape function at a node of interest equals 1
Partition of unity (PU)	The sum of all nodal shape functions within a support domain, evaluated at any point of interest within the same domain equals unity.
Compact	Enables sparse and banded discretised system equation

2.3.2.1 Cutting Techniques in Meshless Method

There are four ways to simulate incision in a meshless domain [8], [22] (Figure 2-10). Visibility criterion is a technique where the shape function of a point that is perceived visible from a simulation node is assigned a zero value, creating an artificial discontinuity. The issue with this method is the definition of the visibility criterion can leave some nodes wrongly diagnosed as being on either side of a cut. The transparency

and diffraction ray methods were proposed to alleviate the wrong diagnosis in visibility method. But unlike the visibility criterion, these methods are for 2D simulation of a discontinuity. Transparency method creates discontinuity based on a distance criterion between the simulation node x_i and any other arbitrary node x. On the other hand, the diffraction ray method is similar to the transparency method, but here, the distance between a simulation point x_i and an arbitrary point x is weighted by the distance of x from the discontinuity tip. This was extended to 3D by Steinemann et al. who proposed the use of visibility graph [31].

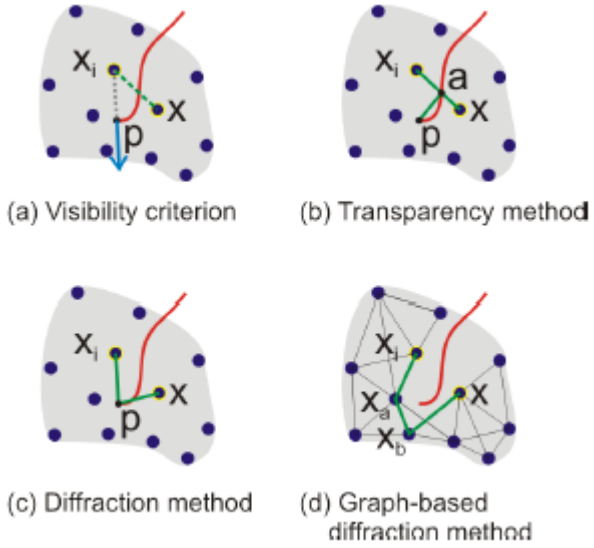

Figure 2-10: Techniques for simulating discontinuity in meshless methods[8]

2.3.2.2 Meshless Simulation of Surgical Incision

There is a long list of meshless methods as a choice of numerical method. A few of these methods have been adopted for surgical simulation and others have been modified for the same purpose.

One of the first methods in meshless methods is the smoothed particle hydrodynamics (SPH). Although originally created for fluid analysis, SPH was

adopted for surgical simulation. In this method, responses including displacements, and forces are approximated at any point of interest using kernel shape functions are used over a smoothing distance (rather than a well-defined domain). A few modifications of SPH including the predictive corrective incompressible SPH, normalised total Lagrangian SPH were proposed by Palyanov et al. and Rausch et al. respectively [45],[46]. Point Collocation-based method of Finite Spheres (PCMFS) was proposed by De et al. for simulating surgical incision [47]. These researchers stated that PCMFS is suitable for simulating real time simulation without the need for pre-computation. The points stated by the authors was that solution accuracy was dependent on the nodal distribution within the domain and the resulting stiffness matrix is non-symmetric. The method was compared to the standard FEM using linear elasticity theory with the hope of using a non-linear elastic model using multiresolution approach in future research. In 2010, the method was extended to the Point-Associated Finite Approach [6].

In 2010, Horton et al. proposed the Meshless Total Lagrangian Explicit Dynamics method (MTLED) [48]. In this method, the problem geometry or domain is discretised into hexahedral that do not conform to the domain and are only used for numerical integration. However, this non-conformity results in volume inaccuracy during integration when intersected with the domain boundary. The shape function used for estimating deformation are non-polynomial functions and its support domain spans outside of a single hexahedron. This non-polynomial shape function is ideal for approximation rather than interpolation since the evaluation of a function at a node using this shape function does not give an exact solution (i.e., lacks Kronecker delta property). In addition, MTLED method seems ideal for average or overall reaction force and displacement since it does not perform well when nodal estimates of the

same variables are required. Nonetheless, similar to TLED from which it was coined, the advantage of MTLED is its total Lagrangian finite element formulation which enables the computation of derivatives with respect to initial or reference configurations and with the use of deformation gradient the system matrix is collated per time step. It was reported to be twice as fast as it TLED counterpart using hexahedral mesh and thrice as fast using tetrahedral mesh [49]. In trying to solve the volume inaccuracy and drawback of inability to easily implement displacement boundary conditions present in MTLED, Zhang et al. proposed the use of tetrahedral and the combination of MLS with polynomial FEM shape function [50]. The latter, however, requires the creation of elements along the domain boundaries. Thus, this leads to the same concern of quality mesh present in mesh-based method. Furthermore, in 2017, Jin et al. proposed a displacement correction to alleviate the implementation of boundary condition issue present in MTLED [51]. While this approach alleviates the issue, it is assumed that it leads to increased computational time, since these are all corrective measures. Two other improvements were suggested by Zou et al. and Chowdhury et al. to enable imposition of boundary condition and obtain a more accurate solution as a result [52], [53]. The latter developed a modified MLS (MMLS) shape function while the former used radial basis polynomial shape function (RPIM).

Similar to these are the smoothed point interpolation methods (SPIM). Though not yet adopted for any surgical simulation, this class of method which is distinguished by the use of radial or point interpolation shape functions and the use of integration cells which are only used for integration. An example is the point interpolation method (PIM) proposed by Wang et al. [54]. The shape functions possess the Kronecker delta property and enables the use of higher order shape functions.

Other meshless methods that have been adopted include the analytical meshless method by Aras et al. [55], and the most recent attempt specifically for virtual simulator proposed by Cheng et al. [56]. In the latter a viscoelastic material model is used and cutting force are considered for tool-tissue interaction.

The main issues with meshless method are the imposition of essential boundary conditions and handling moving discontinuities or the generation of computational surfaces to simulate tool-tissue collision, the creation of new surfaces due to cut, and the dependency of accuracy on effective distribution of nodes [22]. While these problems have solutions, there is an increased computational cost associated to them. For example the issue of collision handling was investigated by Jung and Lee using dynamic bounding volume hierarchy [57]. Many other proposed solutions can be found in the review article by Nguyen et al. [22].

2.3.3 Other Numerical Methods

Another class of method between the mesh-based and meshless method is the extended finite element method (XFEM). The principle is based on decoupling the spatial discretization of the cut from that of the geometry into which it is introduced. This distinction of spatial discretization alludes to the possibility of reaching a balance between accuracy and speed.

In XFEM, the cut is modelled into the problem domain by discretizing the solution space (as given in FEM but) with an additional discontinuous shape function and which amounts to added degree of freedom. The displacement field of the problem is thus computed as the summation of a continuous and discontinuous displacement fields. The most common choice of the enrichment function which satisfies the

Kronecker delta property is the shifted Heaviside function. It preserves the enrichment function within the cut elements but vanishes (equals zero) at the element borders and beyond thus simplifying the implementation. Authors who have used the method include Martin et al. who proposed harmonic enrichment approach that employs one type of enrichment function type for partial, multiple, progressive and complete cuts [39]. Jeřábková et al. also presented an implementation of the XFEM method with total Lagrangian formulation of the FEM, without validation [58].

2.4 Techniques for Reduced Computational Time

Reducing computational time is usually done via three ways: using basic soft tissue models such as the heuristic or linear elastic model (as described in section 2.3.1.3), employing means of reducing the size of the stiffness matrix when implicit time integration method is used or using a GPU implementation of a numerical algorithm.

One of the earliest and notable attempts to reduce computational time for real time soft tissue deformation is the condensation technique proposed by Bro-Nielsen [23] and implemented by Kühnapfel et al [59]. The technique which uses volumetric mesh, is based on the idea of only calculating the displacement of the nodes in the vicinity of the deforming force or tool. Consequently, the global stiffness matrix is only populated with local stiffness matrices of the surface nodes. In the author's work, a linear elastic 3D model of the lower leg was analysed without any force feedback but with a visual update record of 30Hz. A similar concept is utilised in the multiresolution or hierarchical approach where information from the local region under deformation is

used to solve the system equation [60]. The difference with the condensation method is the insertion of new nodes in the region of deformation.

Reduced order method (ROM) was originally proposed by Krysl et al. [61] and implemented by Taylor et al. [37] This method involves approximating the full displacement response of dynamic equilibrium equation using a generalised displacement of much lower dimension [43]. Thus, it reduces the size of the dynamic system equation that must be solved. The aim of using this method was to alleviate the drawback of conditional stability in explicit time integration method. The drawback highlighted in the research is the fact that the ROM is only suited for homogeneous boundary conditions.

An alternative optimization method is computing the bulk of the calculation at the pre-processing stage (prior to the time stepping) as seen in Cotin et al. [41] wherein a 3D hybrid model of a liver was used for verification. This method is very efficient and can be used with any optimization technique were appropriate. The use of the major or significant vibration modes to obtain the deformation of a dynamic problem is another way to reduce the computation time. This technique was implemented by Basdogan in 2001 [62]. In the same year, Wu et al. proposed the idea of a progressive mesh to reduce computational time in FEM and enable real time deformation [63].

Another important alternative for reducing computational time is the type of system equation solver used. For example Cholesky decomposition is faster than LU while LU decomposition is said to be slower than direct solvers or methods that require inversing the stiffness matrix or factorization [6], [64]. Direct solvers do not consider system invariance and they also require large memory for storage. Iterative methods on the other hand could be combined with preconditioners to speed up convergence

and hence computation time. For example, in the research of Courtecuisse et al., conjugate gradient method using an asynchronous preconditioner update at low frequency was said to reduce the number of calculations [19].

The use of a parallel processor also reduces the computational time. Originally created to process graphics, GPU has evolved into a programmable graphics processing unit that can now be used to accelerate algorithms of different applications including engineering problems. This is because GPU's can now perform arithmetic operations; supports higher precision floating point values (32 bits); has increased memory bandwidth; and is faster due to the dedication of its transistors to computing rather than extracting instruction-level parallelism as is the case with CPU [65], [66].

To be able to use GPU, the algorithm must be such that it can be parallelized. In other words, the algorithm must have sections that are independent and can be run simultaneously. Such parts of the algorithm are usually identified as loops [67].

Several reports exist about accelerating algorithms using GPU. In a 2017 report by Mariappan et al., a software developed to predict features of a hepatic lesion could compute lesions in 3mins within a 26mins ablation procedure. While this time is significantly below the requirement for surgical cutting, the report shows that GPU implementations of algorithms can improve performance time in important research fields such as cancer treatment [68]. Possibly the first implementation of a FEM is the 2008 report by Taylor et al. [69]. In the publication, Taylor et al. presented a GPU implementation of a nonlinear finite element algorithm for brain deformation - this was when the programmability of GPU tool was still in its infancy - and an increase in performance in comparison to CPU implementation of the algorithm was reported. Similarly in 2010, Zhang et al. reported an increase in speed for a GPU implementation

of a spring model algorithm used for endoscope image cutting simulation. The GPU implementation was reported to be 27.57 times faster per iteration, than its CPU counterpart, for a 70 x 70 mesh grid resolution [70]. Intuitively, a significant increase in performance is expected with increasing number of iterations for the GPU implementation. Similar to Taylor et al.'s 2008 publication, Mafi and Sirouspour, developed a GPU implementation of a nonlinear finite element and implicit time integration method in 2014. The result obtained confirmed that the GPU implementation of the proposed preconditioned gradient (PCG) method could potentially be used to achieve a haptic feedback rate of 10-1000Hz for soft tissue simulation [71]. In 2015, in the neurosurgical field, Sase et al. compared the computational speed for simulating the opening of a brain fissure via blunt dissection and brain retraction procedures, in real time, using a CPU, multicore CPU (with parallelization) and GPU implementation of nonlinear corotational finite element method [72]. The focus of the research was on time consuming aspects of FEM: matrix assembly and rearrangement. The computational time reported were 20ms and 80ms for blunt dissection and brain retraction respectively. Clearly, from these reports, GPU implementation of parallelized algorithms is faster than CPU, and thus GPU accelerates computation time for parallelizable algorithms.

A current trend in research is geared towards a combination of CPU and GPU. This is reported in many researches [67], [73]–[75]. The article written by Joldes et al. [73] reported 20 times increase in the computation speed for the hybrid method.

2.5 Summary of Review

Many approaches have been adopted by researchers to achieve realistic soft tissue deformation induced by cutting.

Different soft tissue models have been used with numerical method for solving soft tissue deformation. The most suitable soft tissue model to achieve realism in surgical training application is the nonlinear model, since it better captures soft tissue behaviour. In addition, patient specific nonlinear model would be an ideal model, and methods such as machine learning are currently geared towards achieving this.

However, the implication of using a nonlinear soft tissue model is its impact on computational time of the numerical solver for solving the structural problem. The choice of using a nonlinear soft tissue model further compounds the problem of achieving realistic simulation, since displacements are computed from the constitutive equations. Thus, nonlinear models are computationally more expensive in comparison to linear or spring mass models

The numerical method used for computing soft tissue deformation puts another constraint on computational time. Existing mesh-based and meshless numerical methods have inherent limitations. Mesh based methods such as BEM are limited by factors such as suitability for large deformation, incorporation of topological changes, suitability for volumetric model (BEM). The most promising mesh-based method, FEM is affected by the existence of numerical instabilities due to poor mesh quality/ill conditioned elements which invalidates solution. Furthermore, there is the challenge of non-automated mesh generation of certain mesh types like hexahedral and polyhedral mesh which are of better quality than tetrahedral mesh. Furthermore, while a tetrahedron is a more appropriate choice for discretizing irregular shapes such as soft tissues, it exhibits the locking phenomena which affects solution accuracy. However,

existing remedies such as the average nodal pressure tetrahedral elements (ANP) proposed by Joldes et al. [36], seem very promising to combat this issue. Hexahedral on the other hand are high quality elements but produce jagged problem geometry and cut surfaces.

As for meshless methods, the main limitation is the additional computational cost of creating, storing and updating node to node adjacency and imposition of boundary conditions. A possible alternative for simulating arbitrary soft tissue incision is the use of a semi-meshless method such as SPIM, where imposition of boundary conditions can easily be implemented without any treatment due to the use of a polynomial shape function. A more promising and possible the most suitable numerical method for simulating discontinuity is the XFEM. Here discontinuities are created as an enrichment of the basis function and thus alleviates solution instability due to the creation of new and ill-shaped elements as a result of incorporating a cut.

Harnessing this benefit of XFEM and coupling XFEM method with a time integration method such as the explicit time integration method, could enable nonlinear simulation soft tissue incision at interactive rates of 1kHz for haptic rendering.

3 CONTINUUM MECHANICS

The XFEM method, an extension of the finite element method, is based on variational approach which is built upon continuum mechanics theory. The continuum mechanics theory assumes that a structure is approximated as a continuous medium made of continuum particles. However, in reality, any structure is a composition of discrete atoms and subatomic components. Nonetheless, depending on the application - here for the purpose of simulating soft tissue deformation due to incision - it is sufficient, to predict the behaviour of a structure with continuous functions of variables such that the smoothness and continuity of the material holds at any structural level (i.e. micro or macro), and the existence of a granular level of discrete particles (atoms) is ignored.

Continuum mechanics covers three main topics: kinematics (motion without considering its cause, includes the study of deformation and strain); stress in a continuum structure; and the mathematical representation of the physics governing the behaviour of a structure. This section introduces the fundamental and relevant components of continuum mechanics as a prelude to the variational approach and XFEM method presented in chapter 4 and 5. Interested readers can refer to [16], [76] for further details.

3.1 Motion, Deformation and Strain Measure

Kinematics deals with the motion of a body without reference to its mass or the influencing force [76]. Motion is that which causes a change in shape (deformation), position (translation) or orientation (rotation) of a body. A measure of motion, deformation gradient, is central to kinematics and enables the development of

various strain measures. Also, since soft tissue deformation by incision is a dynamic process, velocity, acceleration and the rate of change of deformation and strain measures are briefly described in the following section.

3.1.1 Characterisation of Motion

As a body traverses through space it assumes different configurations over a time t. Thus, continuum points in the body at the start of the motion will have corresponding unique points at other times during the motion. Figure 3-1 shows a body within a Cartesian coordinate system with origin O and coordinate axes defined by unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$, that traverses from an initial (reference) configuration at time $t = t_0$, to new configurations (current configuration) after a time t in space. The body also occupies a region or volume in space (configuration) denoted by Ω and bounded by a surface area denoted by Γ. The motion of such a body is mathematically defined as the vector field $\boldsymbol{\chi}$ that maps a continuum point in the reference configuration, defined by a position vector $\boldsymbol{X}$**,** to a unique point in the new configuration, defined by position $\boldsymbol{x}$ eq. (1)**,** where $\boldsymbol{X} = X_a\boldsymbol{e}_a$, , $\boldsymbol{x} = x_a\boldsymbol{e}_a$, a =1 or 2 or 3 coordinate directions.

$$\boldsymbol{x} = \boldsymbol{\chi}(\boldsymbol{X}, t) \tag{1}$$

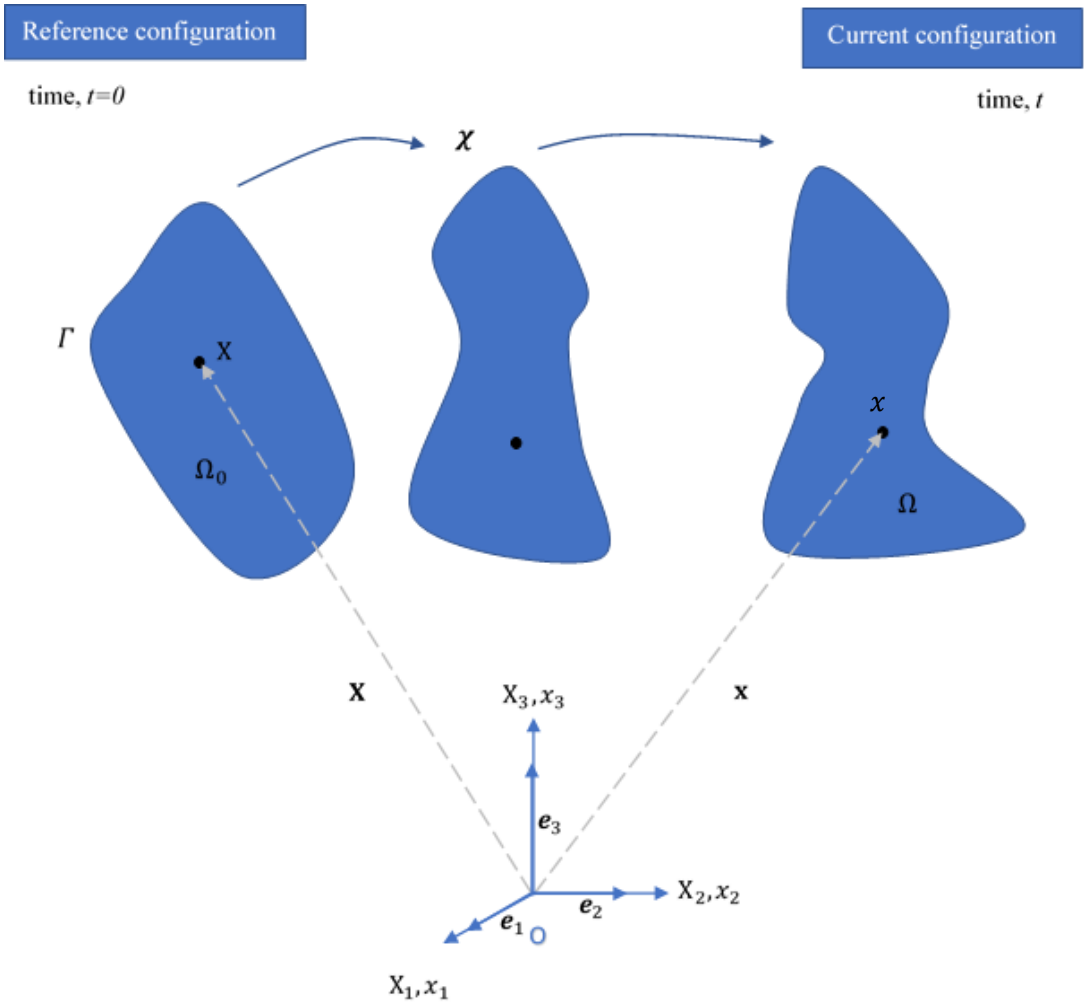

Figure 3-1: Motion of a continuum body

For a valid motion, conditions that must be fulfilled are summarised below [16], [77]:

1. the motion maps a point in the reference configuration to a unique point in the current configuration - that is, the motion must result in a 1-to-1 relation between these points,
2. the motion must be invertible: that is, the position of a particle in the reference configuration must be obtainable from $\boldsymbol{X} = \boldsymbol{\chi}^{-1}(\boldsymbol{x}, t)$, and
3. the motion should be continuously differential to enable the definition of deformation.

Based on the different configurations, it is possible to characterise motion with respect to either the reference (Lagrangian) or current configuration (Eulerian). The approach adopted is dictated by computational convenience of the branch of continuum mechanics (i.e. fluid or solid mechanics), although, either approach is applicable within all continuum branches. The Lagrangian approach is adopted when the focus is the particle (body), whereas Eulerian approach is adopted when the focus is on a specific region in space through which the body traverses.

Displacement, velocity and acceleration

For a body that undergoes motion, displacement, velocity and acceleration can be defined as follows. Displacement is a vector field which describes the change in position of a continuum particle during motion and is defined as:

$$\boldsymbol{U}(\boldsymbol{X},t) = \boldsymbol{x}(\boldsymbol{X},t) - \boldsymbol{X} = \boldsymbol{\chi}(\boldsymbol{X},t) - \boldsymbol{X} \tag{2}$$

It relates the reference and current positions of the particle. The corresponding velocity and acceleration obtained as first and second time derivatives of the displacement field and their corresponding spatial forms are given in the table below:

Table 3-1: Material and Spatial forms of displacement, velocity and acceleration

Quantities	Material form	Spatial form
Displacement	$\boldsymbol{U}(\boldsymbol{X},t) = \boldsymbol{x}(\boldsymbol{X},t) - \boldsymbol{X}$	$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{x} - \boldsymbol{X}(\boldsymbol{x},t)$
Velocity	$\boldsymbol{V}(\boldsymbol{X},t) = \frac{\partial \boldsymbol{U}(\boldsymbol{X},t)}{\partial t}$	$\boldsymbol{v}(\boldsymbol{x},t) = \frac{\partial \boldsymbol{u}(\boldsymbol{x},t)}{\partial t}$
Acceleration	$\boldsymbol{A}(\boldsymbol{X},t) = \frac{\partial^2 \boldsymbol{U}(\boldsymbol{X},t)}{\partial t^2}$	$\boldsymbol{a}(\boldsymbol{x},t) = \frac{\partial^2 \boldsymbol{u}(\boldsymbol{x},t)}{\partial t^2}$

3.1.2 Measures of Deformation

Measure of deformation and strain are crucial to nonlinear continuum mechanics, since motion causes deformation, that is change in the shape and size of a body. We begin with the deformation gradient which is important for the derivation of strain measures. Rate of deformation is also introduced to explore the variation of deformation with time and this is later used for the derivation of balance principles [16].

Deformation gradient

Similar to the displacement which maps a point in the reference configuration relates/maps to its corresponding unique point in the current (deformed) configuration, the deformation gradient describes deformation of line segment (material fibres) within a continuum, that is how a tangent vectors to a curve deform. Given a curve with a line segment $d\boldsymbol{X}$ in the reference configuration and which maps to $d\boldsymbol{x}$ in the current configuration, the deformation gradient which relates both line segments is given as:

$$\boldsymbol{F}(\boldsymbol{X},t) = \frac{\partial \boldsymbol{\chi}(\boldsymbol{X},t)}{\partial \boldsymbol{X}} = Grad\ \boldsymbol{x}(\boldsymbol{X},t). \tag{3}$$

The deformation gradient is also referred to as the Jacobian matrix of motion and it is a 2nd order tensor with 9 components given in matrix form as:

$$\boldsymbol{F} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix} \tag{4}$$

To ensure compatibility (i.e. no gaps or overlaps) in the deformable body, the motion of a body must have an invertible deformation gradient (eq. 5), whose determinant ($det\,\boldsymbol{F}$) must not be singular, that is $det\boldsymbol{F} > 0$ [16].

$$\boldsymbol{F}^{-1}(\boldsymbol{x},t) = \frac{\partial \boldsymbol{\chi}^{-1}(\boldsymbol{x},t)}{\partial \boldsymbol{x}} = Grad\,\boldsymbol{X}(\boldsymbol{x},t), \tag{5}$$

where a component of the tensor is $F_{Aa}^{-1} = \frac{\partial \chi_A^{-1}}{\partial x_a} = Grad_{x_a} X_A$. x_a is the component of $\boldsymbol{x}$ and X_A is the component of $\boldsymbol{X}$.

The gradient of the displacement field, displacement gradient tensor could be defined in relation to the deformation gradient and displacement as:

$$\begin{gathered} Grad\boldsymbol{U}(\boldsymbol{X},t) = Grad\boldsymbol{x}(\boldsymbol{X},t) - Grad\boldsymbol{X} \\ Grad\boldsymbol{U} = \boldsymbol{F}(\boldsymbol{X},t) - \boldsymbol{I} = \quad or \quad \frac{\partial U_a}{\partial X_A} = F_{aA} - \delta_{aA} \end{gathered} \tag{6}$$

where $\boldsymbol{I}$ is the identity tensor and δ_{aA} is the Kronecker delta. The corresponding spatial form is

$$\begin{gathered} grad\boldsymbol{u}(\boldsymbol{x},t) = grad\boldsymbol{x} - grad\boldsymbol{X}(\boldsymbol{x},t) \\ grad\boldsymbol{u} = \boldsymbol{I} - \boldsymbol{F}^{-1}(\boldsymbol{x},t) \quad or \quad \frac{\partial u_A}{\partial x_a} = \delta_{Aa} - F_{Aa}^{-1} \end{gathered} \tag{7}$$

Strain Measures

The strain measure describes the change in length of two neighbouring points as the body undergoes motion. Various strain measures exist in literature. Here, the most common material and spatial strain tensors in literature will be presented, that is right Cauchy Green, left Cauchy Green, Piola deformation and Euler-Almansi tensor. It

should be noted that although there are many strain tensors, anyone of them could be used. The variety is probably due to extensive research on the topic.

The right and left Cauchy Green tensor is defined as eq. (8) . It is obtained from the square of a vector quantity known as stretch.

$$\boldsymbol{\lambda_{a_0}} \cdot \boldsymbol{\lambda_{a_0}} = \boldsymbol{F a_0} \cdot \boldsymbol{F a_0} = \boldsymbol{a_0 F^T F a_0} = \boldsymbol{a_0 C a_0}$$

$$\boldsymbol{C} = \boldsymbol{F^T F} \tag{8}$$

Stretch is defined with respect to the deformation gradient and $\boldsymbol{a_0}$ as

$$\boldsymbol{\lambda_{a_0}}(\boldsymbol{X}, t) = \boldsymbol{F}(\boldsymbol{X}, t)\boldsymbol{a_0}, \tag{9}$$

where $\boldsymbol{a_0}$ is a unit vector defined with respect to the material line $d\boldsymbol{X}$ (Figure 3-2) of two points defined by vector $\boldsymbol{X}$ **and** $\boldsymbol{Y}$ in a body as given by eq. (10)

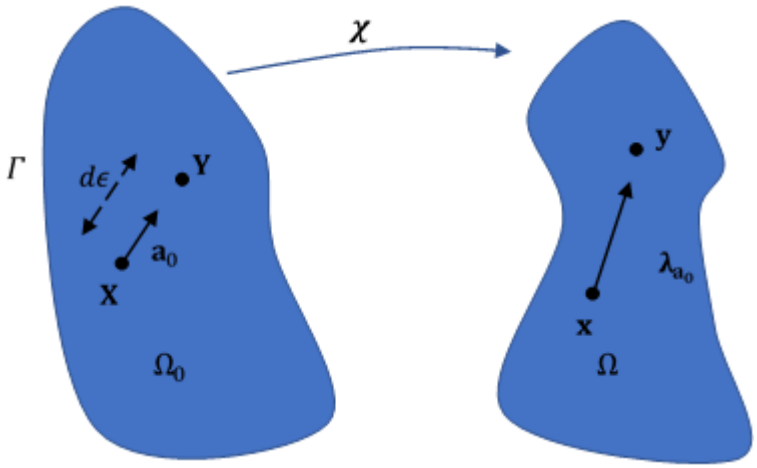

Figure 3-2: Transformation of a material line from reference to current configuration.

$$\boldsymbol{a_0} = \frac{\boldsymbol{Y} - \boldsymbol{X}}{|\boldsymbol{Y} - \boldsymbol{X}|} \tag{10}$$

$$d\boldsymbol{X} = \boldsymbol{Y} - \boldsymbol{X} = \boldsymbol{a_0} d\varepsilon \; ; \quad \frac{d\varepsilon}{|\boldsymbol{X}|} \ll 1; \tag{11}$$

$$\begin{aligned} d\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{x} \; &= \; \boldsymbol{\chi}(\boldsymbol{Y}, t) - \; \boldsymbol{\chi}(\boldsymbol{X}, t); \\ &= \boldsymbol{\chi}(\boldsymbol{X} + d\varepsilon \boldsymbol{a_0}, t) - \; \boldsymbol{\chi}(\boldsymbol{X}, t); \end{aligned} \tag{12}$$

Using Taylor series expansion, total differential, and the distributive property of $\boldsymbol{\chi}$, gives

$$\boldsymbol{y} - \boldsymbol{x} = \boldsymbol{F}(\boldsymbol{X}, t)(\boldsymbol{Y} - \boldsymbol{X}) + o(\boldsymbol{Y} - \boldsymbol{X}); \tag{13}$$

(10) implies that $\boldsymbol{F}(\boldsymbol{X}, t)$ linearly maps the material line from the undeformed to the deformed configuration, and o is the Landau order symbol. Note that dependencies on position and time are omitted for ease of reading. Other material and spatial strain tensors are summarised in Table 3-2 below.

Table 3-2: Material and Spatial strain tensors

Material strain measures		Spatial strain measures	
Piola deformation	$\boldsymbol{B} = \boldsymbol{C^{-1}}$ $= \boldsymbol{F^{-1}F^{-T}}$	Left Cauchy-Green	$\boldsymbol{b} = \boldsymbol{FF^T}$
Green Lagrange	$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$	Euler-Almansi	$\boldsymbol{e} = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{F^{-T}F^{-1}})$

It noteworthy that spatial strain tensors could be derived by resolving vectors along triads of basis vectors as shown above. Also, a spatial tensor could be obtained from its material counterpart via the push-forward and pull-back operations. The former transforms a material strain tensor to a spatial strain tensor and the latter operator does the opposite [16]. Similarly, local motion and strain may be written in terms of pure stretch or pure rotation by decomposing the deformation gradient [16].

3.2 Stress Measure

A body will undergo stress when strained, due to deformation. When a body is acted upon by an external force such that it deforms and or moves through space to a new configuration, reaction or internal forces will be generated. If it is considered that an infinitesimal resultant force ($d\boldsymbol{f}$) (traction force) acts on an infinitesimal spatial surface ds, then it is defined with respect to a spatial traction vector $\boldsymbol{t}$ (or $\boldsymbol{T}$ in the undeformed configuration) as given in (21) and shown in Figure 3-3.

$$d\boldsymbol{f} = \boldsymbol{t}ds = \boldsymbol{T}d\boldsymbol{S}; \qquad \boldsymbol{t}(\boldsymbol{or}\ \boldsymbol{T}) = \boldsymbol{t}(\boldsymbol{x}, t, \boldsymbol{n}) \tag{14}$$

$\boldsymbol{t}$ is termed Cauchy or true traction while $\boldsymbol{T}$ is the first Piola Kirchhoff or nominal traction vector. Both are measured as force per unit surface area**.**

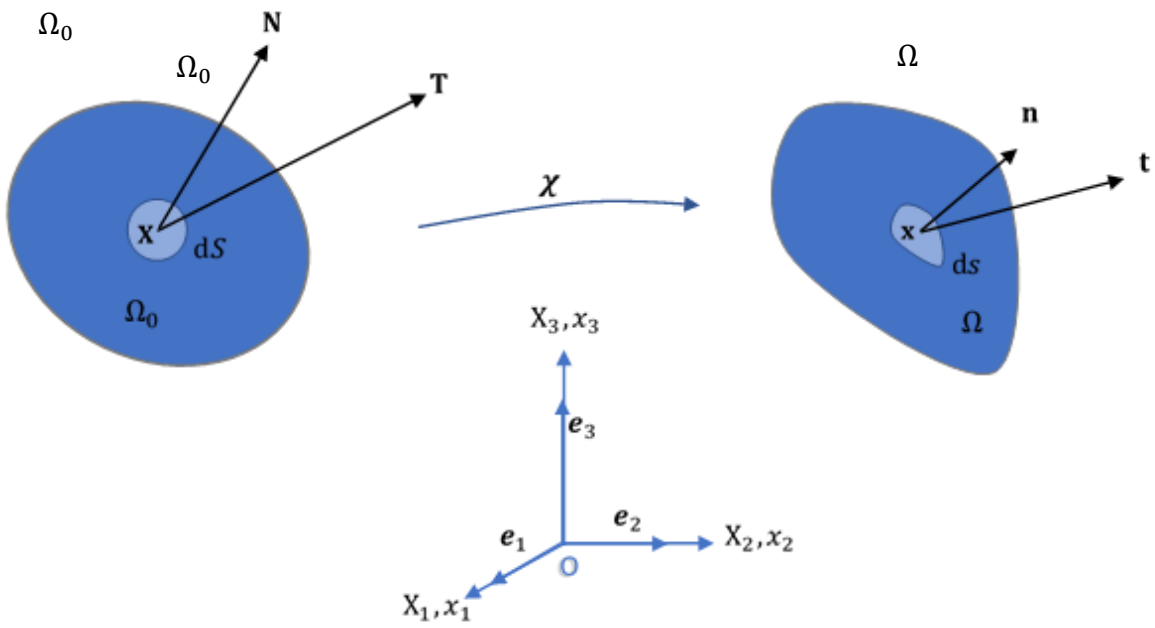

Figure 3-3: Two-dimensional view of an internal force acting on an infinitesimal surface element in a continuum body

The resulting stress due to this internal traction is a second order spatial stress tensor $\boldsymbol{\sigma}$ defined by (22). (23) shows a similar equation for $\boldsymbol{P}$, the first Piola Kirchhoff or nominal stress tensor.

$$\boldsymbol{t}(\boldsymbol{x}, t, \boldsymbol{n}) = \boldsymbol{\sigma}(\boldsymbol{x}, t)\boldsymbol{n}, \tag{15}$$

$$\boldsymbol{T}(\boldsymbol{X}, t, \boldsymbol{N}) = \boldsymbol{P}(\boldsymbol{X}, t)\boldsymbol{N}, \tag{16}$$

$\boldsymbol{n}$ and $\boldsymbol{N}$ are unit outward normal in the spatial and material configurations respectively, $\boldsymbol{\sigma}$ is known as the True Cauchy stress. (22) and (23) are known as the Cauchy stress theorem. It states that a traction force which depends on a unit outward normal must be linear in $\boldsymbol{n}$ and $\boldsymbol{N}$**.** Using relations (15), **(**23) and Nanson's formula, (23) may be related to (16) and vice versa.

$$\begin{aligned} \boldsymbol{P}(\boldsymbol{X}, t) &= J\boldsymbol{\sigma}\boldsymbol{F}^{-T}\ ; \\ \boldsymbol{\sigma} &= J^{-1}\boldsymbol{P}\boldsymbol{F}^{T} = \boldsymbol{\sigma}^{T} \end{aligned} \tag{17}$$

It should be noted that the material form of stress should not be confused to mean that stress exists in the undeformed configuration at time $t = 0$; rather these tensors express stress computed with respect to the material configuration.

Stress, like any tensor, can be expressed in terms of its components. The components of the true stress tensor are given by eq. (18), where the diagonal components correspond to normal stresses and the off-diagonal terms are shear stresses.

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{18}$$

3.3 Balance Principles

The third fundamental aspect of continuum mechanics formulations are the balance principles, also called conservation laws: Conservation of Mass (CoMs), Conservation of Momentum (CoMm) and Conservation of Energy (CoE). CoMm and CoE are considered for a purely mechanical system: other types of energy besides mechanical energy are ignored, for example, the effects of temperature.

3.3.1 Conservation of Mass (CoMs)

Under the assumption that matter cannot be created or destroyed, the law states that mass cannot be zero and it is conserved during motion: mass of a body in the reference configuration is the same as in the current configuration (26):

$$m_{\Omega_0} = m_{\Omega} \tag{19}$$

An equation that exists from CoMs is the continuity equation given as:

$$\rho_0 = \rho J \tag{20}$$

3.3.2 Conservation of Momentum (CoMm)

Newton's second law states that the rate of change of momentum equals total force acting on a body. The conservation of momentum expressed as the material time derivative of linear (and angular) momentum with respect to the material configuration (material time derivative of momentum) results in a total force (or moment) acting on a continuum body:

$$\dot{\boldsymbol{\mathcal{L}}}(t) = \frac{D}{Dt}\int_{\Omega_0} \rho_0 \boldsymbol{V} dV = \int_{\Omega_0} \rho_0 \frac{D\boldsymbol{V}}{Dt} dV = \frac{D}{Dt}\int_{\Omega} \rho \boldsymbol{v} dv = \boldsymbol{F_f}(t) \tag{21}$$

The force $\boldsymbol{F_f}(t)$ can be split into body force and traction force integrated over the reference domain (Ω_0) and reference boundary ($\partial\Omega$) respectively, as eq. (22)

$$\boldsymbol{F_f}(t) = \int_{\Omega_0} \rho \dot{\boldsymbol{v}} dV = \int_{\partial\Omega} \boldsymbol{t} ds + \int_{\Omega} \boldsymbol{b} dv \tag{22}$$

The equivalent material form of eq. (23) is:

$$\boldsymbol{F_f}(t) = \int_{\Omega_0} \rho_0 \dot{\boldsymbol{V}} dV = \int_{\partial\Omega} \boldsymbol{T} dS + \int_{\Omega} \boldsymbol{B} dV \tag{23}$$

Body forces include weight while traction are surfaces forces such as frictional or contact forces. Although there is angular momentum, for the purposes of this thesis only linear momentum will be presented. Using the definition of material time derivative of an integral, Cauchy's stress theorem, and the divergence theorem[1], the traction term in eq. (22) may be written as a volume integral:

$$\int_{\partial\Omega} \boldsymbol{t}(\boldsymbol{x}, t, \boldsymbol{n}) ds = \int_{\partial\Omega} \boldsymbol{\sigma}(\boldsymbol{x}, t) \boldsymbol{n} ds = \int_{\Omega} div\, \boldsymbol{\sigma}(\boldsymbol{x}, t) dv. \tag{24}$$

Cauchy's equation of equilibrium given in the spatial form as:

$$\rho \dot{\boldsymbol{v}} = div\, \boldsymbol{\sigma}(\boldsymbol{x}, t) + \boldsymbol{b} \;\; or \; \rho \dot{v}_a = \frac{\partial \sigma_{ab}}{\partial x_{ab}} + b_a \tag{25}$$

Its material form is obtained based on the polar identity $Div(J\boldsymbol{F}^{-T}) = \boldsymbol{o}$. Where $\boldsymbol{o}$ is a zero vector and $Div\, \boldsymbol{P}(\boldsymbol{x}, t) = JGrad\boldsymbol{\sigma}: \boldsymbol{F}^{-T} + \boldsymbol{\sigma}(\boldsymbol{x}, t) Div(J\boldsymbol{F}^{-T}) = J(Div\boldsymbol{\sigma})\boldsymbol{F}^{-T} = Jdiv\boldsymbol{\sigma}$ as:

$$Div\, \boldsymbol{P}(\boldsymbol{x}, t) + \boldsymbol{B} = \rho_0 \dot{\boldsymbol{V}} \tag{26}$$

This statement $\int_{\Omega} (div\, \boldsymbol{\sigma}(\boldsymbol{x}, t) + \boldsymbol{b}) dv = 0$ holds for any arbitrary volume v, hence, the integral has been removed.

[1] Also called integration by parts[80] or Gauss theorem [77]

3.3.3 Conservation of Energy (CoE)

Conservation of energy is also called theorem of power expended. It states that the rate of change of kinetic energy of a continuum body ($\frac{D}{Dt}K(t)$) and the rate of work done by internal stress $P_{int}(t)$ equals the rate of external mechanical work done on the system $P_{ext}(t)$. Mathematically written as:

$$\frac{D}{Dt}K(t) + P_{int}(t) = P_{ext}(t)\,, \tag{27}$$

The equivalent of this statement in spatial and material forms are

$$\frac{D}{Dt}\int_{\Omega}\frac{1}{2}\rho\boldsymbol{v}^2 dv + \int_{\Omega}\boldsymbol{\sigma}:\boldsymbol{d}\,dv = \int_{\partial\Omega}\boldsymbol{t}\cdot\boldsymbol{v}\,ds + \int_{\Omega}\boldsymbol{b}\cdot\boldsymbol{v}dv,$$

$$\frac{D}{Dt}\int_{\Omega}\frac{1}{2}\rho_0\boldsymbol{V}^2 dV + \int_{\Omega}\boldsymbol{P}:\dot{\boldsymbol{F}}\,dV = \int_{\partial\Omega}\boldsymbol{T}\cdot\boldsymbol{V}\,dS + \int_{\Omega}\boldsymbol{B}\cdot\boldsymbol{V}dV$$

respectively, where $\boldsymbol{d}$ is the rate of deformation tensor, and $\dot{\boldsymbol{F}}$ is the rate of deformation gradient tensor.

From this principle, specifically from the (alternative) expression of the internal power, stress-deformation pairs referred to as work conjugates are obtained. These includes $J\boldsymbol{\sigma}/\boldsymbol{d}$, $\boldsymbol{P}/\dot{\boldsymbol{F}}$, $\boldsymbol{S}/\dot{\boldsymbol{E}}$ etc (see [16], [76]). $\boldsymbol{S}$ is the 2nd Piola Kirchhoff stress. Thus, it follows that a system whose sum of external (Π_{ext}) and internal (Π_{int}) potential energy and kinetic energy is unchanged is said to have its energy conserved:

$$\Pi_{ext}(t) + \Pi_{int}(t) + K(t) = constant \tag{28}$$

3.4 Hyperelastic Material Models

Further to the continuum concept presented earlier for the study of the physical behaviour of a body, a constitutive model is required to describe the material's stress-

strain behaviour. This relation is partly obtained based on the phenomenological approach, where a mathematical model (equation) is fitted to experimental data. Here, hyperelastic material is considered because it has been presented in many literature as the simplest and sufficient model for characterising soft tissue behaviour which are known to exhibit large deformation or finite strain [17], [78].

Hyperelastic material is a class of nonlinear elastic material characterised by a stored potential energy W, and the work done by its stress field is dependent only on its initial and final configuration rather than work path. The stored energy function is a scalar-valued function more generally referred to as Helmholtz free-energy function, measured per unit volume. The constitutive equation is given by

$$\boldsymbol{P} = \frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}}, \qquad \boldsymbol{\sigma} = J^{-1}\frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}}\boldsymbol{F}^{T}. \tag{29}$$

Although (36) shows the strain energy function written as a function of $\boldsymbol{F}$, it could also be written in terms of $\boldsymbol{C}$ or $\boldsymbol{E}$ and are all equivalent: $W(\boldsymbol{F}) = W(\boldsymbol{C}) = W(\boldsymbol{E})$, which will produce different stress measures accordingly. The strain energy is said to be objective: it is independent of the observer irrespective of translation or rotation in space [16]. When written in terms of $W(\boldsymbol{C})$ *or* $W(\boldsymbol{E})$ it is referred to as a reduced form of constitutive equation and mathematically expressed as:

$$\boldsymbol{P} = 2\boldsymbol{F}\frac{\partial W(\boldsymbol{C})}{\partial \boldsymbol{C}}; \qquad \boldsymbol{S} = 2\frac{\partial W(\boldsymbol{C})}{\partial \boldsymbol{C}} = \frac{\partial W(\boldsymbol{E})}{\partial \boldsymbol{E}} \tag{30}$$

A fundamental characteristic of the hyperelastic material is path independency, is mathematically expressed as

$$\int_{t_1}^{t_2} \boldsymbol{P}: \dot{\boldsymbol{F}}\, dt = \int_{t_1}^{t_2} \frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}}: \dot{\boldsymbol{F}}\, dt = \int_{t_1}^{t_2} \frac{DW(\boldsymbol{F})}{Dt}\, dt = W(\boldsymbol{F}_2) - \Psi(\boldsymbol{F}_1) = 0 \quad (31)$$

where $\boldsymbol{F}_2 = \boldsymbol{F}_1$

3.4.1 Isotropic hyperelastic material

A material is said to be isotropic if its behaviour is independent of direction and therefore (39) holds for a description with respect to the reference configuration. This expression implies that the store energy in a body is not affected by orthogonal rotation $\boldsymbol{Q}$.

$$W(\boldsymbol{F}) = W(\boldsymbol{F}^*)$$
$$\boldsymbol{F} = \boldsymbol{F}^* \boldsymbol{Q}^T \quad (32)$$

For an isotropic hyperelastic material, it is also possible to write the stored energy function in terms of principal invariants I_i of $\mathbf{C}$ (or $\mathbf{b}$); referred to as the representation theorem of invariants (40) and Table 2-1.

$$W = W[I_1(\boldsymbol{C}), I_2(\boldsymbol{C}), I_3(\boldsymbol{C})] = W[I_1(\boldsymbol{b}), I_2(\boldsymbol{b}), I_3(\boldsymbol{b})] \quad (33)$$

Table 3-3:Expression of Invariants of **C** or **b** for an isotropic hyperelastic material in terms of principal stretches (λ_i) in all directions and trace (tr) and determinant (det) operators

Invariants	In terms of $\boldsymbol{b}$ or $\boldsymbol{C}$	Differential
$\boldsymbol{I_1}$	$tr\boldsymbol{b} = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$	$\frac{\partial I_1}{\partial \boldsymbol{b}} = \frac{\partial (tr\boldsymbol{b})}{\partial \boldsymbol{b}} = \frac{\partial (\boldsymbol{I}:\boldsymbol{b})}{\partial \boldsymbol{b}} = \boldsymbol{I}$
$\boldsymbol{I_2}$	$\frac{1}{2}\left[(tr\boldsymbol{b}) - tr(\boldsymbol{b^2})\right] = tr\boldsymbol{b}^{-1} det\boldsymbol{b}$ $= \lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_3^2$	$\frac{\partial I_2}{\partial \boldsymbol{b}} = \frac{\partial}{\partial \boldsymbol{b}}\left[\frac{1}{2}(tr\boldsymbol{b}) - \frac{1}{2}tr(\boldsymbol{b^2})\right] = I_1\boldsymbol{I} - \boldsymbol{b}$
$\boldsymbol{I_3}$	$det\boldsymbol{b} = J^2 = \lambda_1^2\lambda_2^2\lambda_3^2$	$\frac{\partial I_3}{\partial \boldsymbol{b}} = I_3\boldsymbol{C}^{-1}$

Correspondingly, strain energy derivatives can be expressed as in (41), where **a** represents any appropriate strain measure. The partial derivatives of invariants with respect to such tensors are given in Table 3-3 above.

$$\frac{\partial W(\boldsymbol{a})}{\partial \boldsymbol{a}} = \sum_i^3 \frac{\partial W(\boldsymbol{a})}{\partial I_i}\frac{\partial I_i}{\partial \boldsymbol{a}} \tag{34}$$

For example, the second Piola-Kirchhoff stress could be re-expressed in terms of the principal invariants as

$$\boldsymbol{S} = 2\frac{\partial W(\boldsymbol{C})}{\partial \boldsymbol{C}} = 2\left[\left(\frac{\partial W}{\partial I_1} + I_1\frac{\partial W}{\partial I_2}\right)\boldsymbol{I} - \frac{\partial W}{\partial I_2}\boldsymbol{C} + I_3\frac{\partial W}{\partial I_3}\boldsymbol{C}^{-1}\right] \tag{35}$$

A material is said to be incompressible when it experiences insignificant volume change under deformation. For such material $J = 1$, and Poisson ratio $\nu = 0.5$. The strain energy function could be expressed in such cases as:

$$W = W(\boldsymbol{F}) - p(J-1), \tag{36}$$

where $J = det\boldsymbol{F} = 1$, the scalar p is a Lagrange multiplier, identifiable as the hydrostatic pressure. It is a workless reaction to the kinematic constraint of the deformation field, and it is only obtainable from the equilibrium equation and boundary conditions. Differentiating eq. (44) and adopting the necessary transformation to covert from first Piola-Kirchhoff stress to other stress measures gives:

$$\boldsymbol{P} = -p\boldsymbol{F}^{-T} + 2\frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}}$$

$$\boldsymbol{S} = -p\boldsymbol{F}^{-1}\boldsymbol{F}^{-T} + \boldsymbol{F}^{-1}\frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}} = -p\boldsymbol{C}^{-1} + 2\frac{\partial W(\boldsymbol{C})}{\partial \boldsymbol{C}} \tag{37}$$

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}}\boldsymbol{F}^{T} = -p\boldsymbol{I} + \boldsymbol{F}\left(\frac{\partial W(\boldsymbol{F})}{\partial \boldsymbol{F}}\right)^{T}$$

For an isotropic incompressible hyperelastic material, $I_3 = 1$. Thus, the strain energy function and stress are expressible in terms of the other principal invariants:

$$W = W[I_1(\boldsymbol{C}), I_2(\boldsymbol{C})] - \frac{1}{2}p(I_3 - 1)$$

$$\boldsymbol{S} = -p\boldsymbol{C}^{-1} + 2\left(\frac{\partial W}{\partial I_1} + I_1\frac{\partial W}{\partial I_2}\right)\boldsymbol{I} + 2\frac{\partial W}{\partial I_2}\boldsymbol{C} \tag{38}$$

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\frac{\partial W}{\partial I_1}\boldsymbol{C} - 2\frac{\partial W}{\partial I_1}\boldsymbol{C}^{-1}$$

Nearly incompressible hyperelastic have been used to model soft tissue behaviour in literature, as they greatly simplify the numerical treatment of the problem; for such materials, the Poisson's ratio is somewhat less than 0.5, with values of around 0.49 commonly used [16]. An example of Neo-Hookean strain energy function for nearly incompressible isotropic hyperelastic material is

Neo-Hookean: $$\Psi(\boldsymbol{C}) = \frac{1}{2}\lambda_0(ln\,J)^2 - \mu_0\,ln\,J + \frac{1}{2}\mu_0(tr\boldsymbol{C} - 3)$$

3.5 Conclusion

The equilibrium equation presented in this chapter is based on the theory of continuum mechanics, which is most ideal for computing realistic, nonlinear deformation of soft tissues. The continuum mechanics equations presented here will be used in the following chapter to derive a weak form of the equation which is pertinent to the FEM and upon which the XFEM in chapter 5 will be based.

4 VARIATIONAL APPROACH & FINITE ELEMENT METHODS

The continuum mechanics theory presented in chapter 3 will be used here to develop the mathematical models which predict the mechanical behaviour of a system. The resulting differential equilibrium equations previously obtained could be solved directly without transformation into integral (weak) forms only in special cases, that is when the solution domain and boundary conditions are "simple" enough to obtain a result. However, integral forms are preferred for solid mechanics problems presented here, since the differential equation is satisfied in an integral sense rather than at every infinitesimal point within the domain. This section looks at the solution of governing equilibrium equation using the variational method.

4.1 Variational Approach

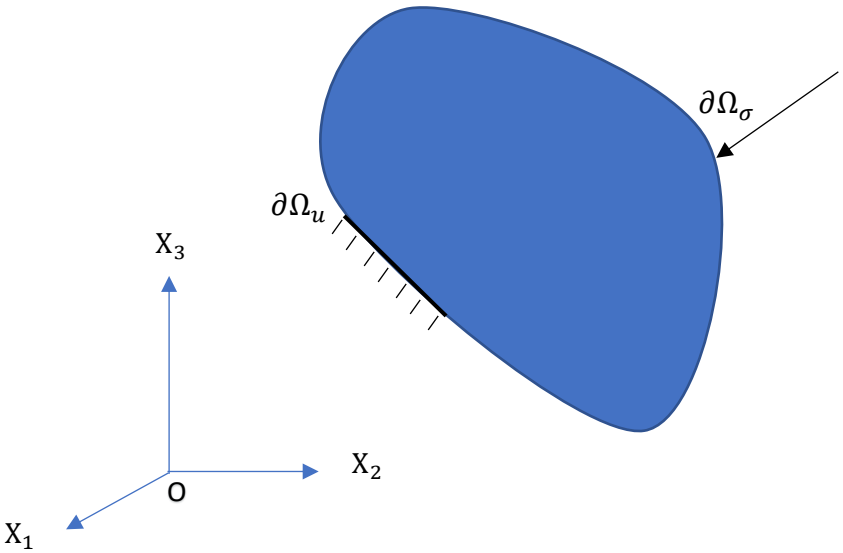

Figure 4-1: Boundary-valued problem showing a domain in the cartesian coordinate with imposed displacement $\partial\Omega_u$ and force $\partial\Omega_\sigma$ constraint

The weak form of the differential equilibrium equations of a system, in terms of its state variable, is obtainable via the variational method. For a boundary valued problem (Figure 4-1), the governing mathematical model (46) which accounts for equilibrium, constitutive relations and boundary constraints $\boldsymbol{u}$ and $\boldsymbol{t}$ are:

$$
\begin{aligned}
&div\,\boldsymbol{\sigma}(\boldsymbol{x},t)+\boldsymbol{b}=\rho\ddot{\boldsymbol{u}},\\
&\boldsymbol{u}=\bar{\boldsymbol{u}}\ on\ \partial\Omega_u\ ,\\
&\boldsymbol{t}=\bar{\boldsymbol{t}}\ \ on\ \partial\Omega_\sigma\ ,\\
&\boldsymbol{u}(\boldsymbol{x},t)|_{t=0}=\boldsymbol{u}_0(\boldsymbol{X}),\\
&\dot{\boldsymbol{u}}(\boldsymbol{x},t)|_{t=0}=\dot{\boldsymbol{u}}_0(\boldsymbol{X}),
\end{aligned}
\tag{39}
$$

where $div\,\boldsymbol{\sigma}(\boldsymbol{x},t)$ is the divergence of the internal stress with respect to the current configuration and is a function of deformation gradient (37), $\boldsymbol{b}$ is the body force measured per unit current volume, $\bar{\boldsymbol{u}}$ is the prescribed displacement boundary condition on the displacement boundary $\partial\Omega_u$, $\bar{\boldsymbol{t}}$ is the prescribed Cauchy traction vector on the traction boundary $\partial\Omega_\sigma$ measured per unit surface area, while $\boldsymbol{u}_0(\boldsymbol{X})$ and $\dot{\boldsymbol{u}}_0(\boldsymbol{X})$ are initial displacement and velocity respectively. The weak formulation simply means an integral form of the governing differential equilibrium equations, and unlike (46) which must be continuously differentiable twice (C^1), its variational (integral) form only needs to be continuously differentiable once. The integral form is the basis for developing finite element equations for any given structural problem and under any given constraint.

The procedure for obtaining the variational equilibrium equation of a system is as follows:

1. Choose a functional, Π, that is a scalar field, for instance, potential energy.
2. Invoke stationarity of the variation of the functional: $\delta\Pi=0$

3. Obtain the governing equilibrium equation and boundary constraints by evaluating $\delta\Pi = 0$ using divergence theorem (integration by parts) and the variation condition [79] [80]

If m is the highest order derivative of the test function within a functional, then the variation condition states that the variation of the variable and its derivatives up to order $(m-1)$ must be zero at all essential (or displacement) boundary conditions, except at prescribed displacement boundaries where it is equal to the prescribed value. In other words, for a functional problem of order C^{m-1}, the highest derivative in the essential boundary condition must be order $m-1$. Similarly, the highest derivative in the natural boundary condition must be $2m-1$.

4.1.1 Principle of Virtual Work (or virtual displacement)

There is no unique functional for a problem type. For instance, in structural mechanics, one could adopt minimum potential energy, displacement-based variational formulations, Hu-Washizu, Hellinger-Reissner principle etc. Here, the principle of virtual work is presented as the simplest variational principle and it is equivalent to the principle of minimizing the total potential energy of a system [80].

The principle of virtual work states that for any compatible or kinematically admissible virtual (unreal) displacement applied to a body in equilibrium, that is, the displacement is continuous and satisfies the variation condition [80], the external and internal virtual work must be equal. The above statement is written mathematically as

$u \in V$ such that $a(u,v) = (f,v) \qquad \forall v \in V$ (40)

where space V is defined as: $V = \left\{ v | v \in L^2(L), \frac{dv}{dx} \in L^2(L), v|_{x=0} = 0 \right\}$

and $L^2(L)$ is a square integral space over length $0 \leq x \leq L$ written as

$$L^2(L) = \left\{ w \middle| w \; is \; defined \; over \; 0 \leq x \leq L \; and \; \int_0^l w^2 dx = \|w\|_{L^2}^2 < \infty \right\}$$

where $a(u,v)$ represents the internal work done and (f,v) represents the external work done.

Following the procedure outlined in section 4.1 above, the functional is obtained from the dot product of an arbitrary virtual displacement field $\delta\boldsymbol{u}$ and linear momentum equation (28). Fulfilling the variational condition that a variation and its derivatives are zero at boundaries or equal to the prescribed boundary conditions, using integration by parts, and rearranging the resulting equation results in

$$f(\boldsymbol{u}, \delta\boldsymbol{u}) = \int_\Omega \boldsymbol{\sigma} : grad(\delta\boldsymbol{u})\, dv = \int_\Omega [(\boldsymbol{b} - \rho\ddot{\boldsymbol{u}}) \cdot \delta\boldsymbol{u}]\, dv + \int_{\partial\Omega} \boldsymbol{t} \cdot \delta\boldsymbol{u}\, ds = 0,$$

$$\int_\Omega \boldsymbol{u}|_{t=0} \cdot \delta\boldsymbol{u}\, dv = \int_\Omega \boldsymbol{u}_0(\boldsymbol{X}) \cdot \delta\boldsymbol{u}\, dv \tag{41}$$

$$\int_\Omega \dot{\boldsymbol{u}}|_{t=0} \cdot \delta\boldsymbol{u}\, dv = \int_\Omega \dot{\boldsymbol{u}}_0(\boldsymbol{X}) \cdot \delta\boldsymbol{u}\, dv$$

(41) is what is referred to as a variational equation, and in this case is a spatial expression of the problem. The equivalent material form according to the conversion $\int_\Omega \boldsymbol{\sigma} : \delta\boldsymbol{e}\, dv = \int_\Omega \boldsymbol{\sigma} : grad(\delta\boldsymbol{u})\, dv = \int_\Omega J\boldsymbol{\sigma} : Grad(\delta\boldsymbol{u})\boldsymbol{F}^{-1}\, dV = \int_{\Omega_0} \boldsymbol{P} : Grad(\delta\boldsymbol{u})\, dV$, is

$$\int_{\Omega_0} \boldsymbol{P} : Grad\delta\boldsymbol{u}\, dV = \int_{\Omega_0} \boldsymbol{B} \cdot \delta\boldsymbol{u}\, dV + \int_{\Omega_0} \rho\ddot{\boldsymbol{u}} \cdot \delta\boldsymbol{u}\, dV + \int_{\partial\Omega_{0,\sigma}} \boldsymbol{T} \cdot \delta\boldsymbol{u}\, dS \tag{42}$$

As alluded to earlier, adopting the variational equation reduces the smoothness requirement of the governing equilibrium equation by virtue of integration by parts.

By using this integration method, the derivative of the stress (i.e. $div\,\boldsymbol{\sigma}(\boldsymbol{x},t)$) is eliminated, and the smoothness requirement of the stress field is correspondingly reduced to C^{-1} (since the highest derivative of stress is 0). The trial function (displacement field), in turn, by virtue of the constitutive relation, only needs to be C^0 continuous. Similarly, the use of Cauchy's equation of motion results in the traction boundary constraint being inherent in the governing integral equation (see [20], [80])

4.2 Finite Element method (FEM)

Approximation techniques are employed to solve equations for which closed forms or exact solutions are not attainable. The finite element method (FEM) is one such numerical method that could be used to solve the variational equations (48) or (49). It is derived by combining the benefits of the weighted residual Galerkin procedure and Ritz methods [80]. In other words, the Galerkin method is written in a form that allows the use of polynomial trial functions used in Ritz and thus combines the benefits of symmetry, positive definiteness, differentiability of the trial function, and fulfilment of boundary constraints. FEM is sometimes referred to as Galerkin FEM in literature.

4.2.1 Discretization of the Weak Form

In FEM, the solution of the equilibrium equations is obtained by discretizing (subdividing) the solution space and system geometry into an assembly of discrete non-overlapping sub-bodies (elements), with associated shape functions describing the spatial variation of solution variables, interconnected at joints (nodes) (Figure 4-2).

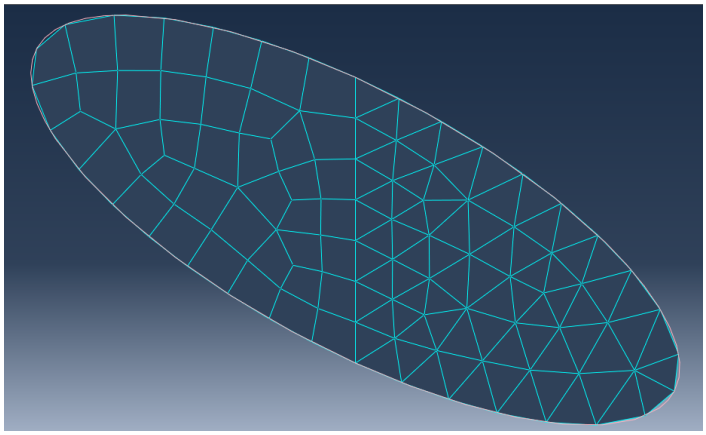

Figure 4-2: Finite element mesh using triangular and quadrilateral element

The displacements within each element (or local displacement) is then approximated from the nodes using the function:

$$\boldsymbol{u}_h(x) = \sum_{i=1}^{n} N_i \boldsymbol{u}_i = \boldsymbol{N}(\boldsymbol{x})\boldsymbol{u}_{(e)}, \tag{43}$$

where $\boldsymbol{u}^h(\boldsymbol{x}) = [u(x) \quad v(x) \quad w(x)]^T$ is the approximated displacement at position $\boldsymbol{x} = [x \quad y \quad z]^T$. $\boldsymbol{N}(\boldsymbol{x}) = [N_1(\boldsymbol{x})\boldsymbol{I} \quad N_2(\boldsymbol{x})\boldsymbol{I} \quad \cdots \quad N_n(\boldsymbol{x})\boldsymbol{I}]$ is a matrix of scalar valued shape functions $N_i(\boldsymbol{x})$, $\boldsymbol{u}_i = [u_i \quad v_i \quad w_i]^T$ is a vector of the displacements of the node i and is commonly represented using polynomial, $\boldsymbol{u}_{(e)} = [\boldsymbol{u}_1^T \quad \boldsymbol{u}_2^T \quad \cdots \quad \boldsymbol{u}_n^T]^T$ is the vector of displacements for all nodes of element e, and n is the number of nodes per element. $\boldsymbol{I}$ is the identity matrix of order D, for a D-dimensional model.

To obtain the discretized weak formulation or finite element equations, finite element interpolants are substituted into the weak formulation (44) as trial functions:

$$\int_{\Omega} \boldsymbol{\sigma} : (\delta \boldsymbol{u} \otimes \boldsymbol{\nabla} N_i)\, dv - \int_{\Omega} \boldsymbol{b} \cdot \delta \left(\sum_{i=1}^{n} N_i \boldsymbol{u}_i \right) dv - \int_{\Omega} \rho \ddot{\boldsymbol{u}} \cdot \delta \left(\sum_{i=1}^{n} N_i \boldsymbol{u}_i \right) dv = \int_{\partial\Omega} \boldsymbol{t} \cdot \delta \left(\sum_{i=1}^{n} N_i \boldsymbol{u}_i \right) ds \tag{44}$$

Given that the nodal displacement is independent of the integration and using the contraction property $\boldsymbol{\sigma} : (u \otimes v) = u.\boldsymbol{\sigma} v$ gives

$$\delta \boldsymbol{u} \cdot \left(\int_{\Omega} \boldsymbol{\sigma} \boldsymbol{\nabla} N_i \, dv - \int_{\Omega} N_i \boldsymbol{b} \, dv - \int_{\Omega} \rho N_i \sum_{j}^{n} N_j \ddot{\boldsymbol{u}} \, dv - \int_{\partial\Omega} N_i \boldsymbol{t} ds \right) = 0 \tag{45}$$

Since the $\delta \boldsymbol{u}$ is arbitrary, it follows that

$$\int_{\Omega} \boldsymbol{\sigma} \boldsymbol{\nabla} N_i \, dv - \int_{\Omega} N_i \boldsymbol{b} \, dv - \int_{\Omega} \rho N_i \sum_{j}^{n} N_j \ddot{\boldsymbol{u}} \, dv - \int_{\partial\Omega} N_i \boldsymbol{t} ds = 0 \tag{46}$$

$$\int_{\Omega} \boldsymbol{\sigma} \boldsymbol{\nabla} N_i \, dv - \int_{\Omega} \rho N_i \sum_{j}^{n} N_j \ddot{\boldsymbol{u}} \, dv - \int_{\partial\Omega} N_i \boldsymbol{t} ds - \int_{\Omega} N_i \boldsymbol{b} \, dv = 0 \tag{47}$$

$$\boldsymbol{f}_{int} - \boldsymbol{f}_{kin} - \boldsymbol{f}_{ext} = 0 \tag{48}$$

Equations (47), (48) are nonlinear equations in $\boldsymbol{u}$ and gives n-simultaneous equations, where $\boldsymbol{f}_{int}$, $\boldsymbol{f}_{kin}$ and $\boldsymbol{f}_{ext}$ are internal, kinetic and external forces respectively.

It should be noted that in FEM, nonlinearity (i.e. where any variable in the equilibrium equation becomes a nonlinear function of $\boldsymbol{u}$) can be introduced in three ways: material nonlinearity and contact and kinematic nonlinearity. The latter could lead to nonlinearity in the geometry of the body or applied constraint, due to motion. Depending on the type of nonlinearity (or, absence of a given type), eq. (48) could be simplified accordingly [80].

When the load is slowly applied and the inertia forces are negligible, $\boldsymbol{f}_{kin}$ can be dropped to give the static equilibrium equation:

$$\boldsymbol{f}_{int} = \boldsymbol{f}_{ext} \tag{49}$$

Note that for discretization of FEM equation using continuous shape function, the derivative of the shape function $\boldsymbol{N}$ in space with respect to the original configuration may be written for the 3D case as

$$\boldsymbol{\partial N_X} = \begin{bmatrix} \frac{\partial N_1}{\partial X} & \frac{\partial N_1}{\partial Y} & \frac{\partial N_1}{\partial Z} \\ \frac{\partial N_2}{\partial X} & \frac{\partial N_2}{\partial Y} & \frac{\partial N_2}{\partial Z} \\ \vdots & \vdots & \vdots \\ \frac{\partial N_n}{\partial X} & \frac{\partial N_n}{\partial Y} & \frac{\partial N_n}{\partial Z} \end{bmatrix} \tag{50}$$

4.2.2 Solution of Nonlinear Equilibrium Equations

For an equilibrium problem which is nonlinear and time dependent, the explicit or implicit time integration methods could be adopted. For the iterative step in implicit time integration, Newton Raphson based methods are used [80]. These methods which combine iterative and time integration algorithm are collectively called direct integration methods.

Both methods can be described as a procedure that solves for static solution of equilibrium at discrete time steps. In other words, if a constrained body experiences some motion during a given time T, displacement solutions are obtained at n discrete periods or time steps based on a chosen time $t < T \; and \; n = T/t$. Depending on the method, the time t could vary over the entire T [80].

Several types of each methods exist. The explicit time integration method such as the central differencing method, differs from its implicit counterparts which includes Newmark method from the perspective of stability and solution procedure. While explicit integration method is only conditionally stable and the nodal displacement solution at a current time step is obtained using solution of variables from previous time, implicit time integration method is unconditionally stable and solutions of variables at current time step is used. Thus, for this thesis, the explicit method was adopted, and further details of the chosen type is given in the next section.

4.3 Total Lagrangian Explicit Dynamics (TLED)

The TLED method is adopted, where the finite element discretization of the total Lagrangian (TL) weak form is combined with the central differencing explicit method.

4.3.1 Dynamic Equilibrium Equation: Total Lagrangian Formulation

TL formulation computes field variables (i.e. displacement, stress or strain), derivatives, and integrals with respect to the reference configuration; as such, second Piola-Kirchhoff stress $\boldsymbol{S}$ and Green strain $\boldsymbol{E}$ are the adopted stress and strain measures. For nonlinear analysis, referring variables and integrals to the reference configuration has the advantage that element shape functions derivatives remain unchanged throughout an analysis and can be precomputed; this is particularly useful for simulating soft tissue incisions which is restricted by computational time [81].

Internal forces

The internal nodal forces $\boldsymbol{f}^{int}$ may be evaluated element-wise, according to:

$$\boldsymbol{f}^{int} = \sum_{e} \boldsymbol{f}_{(e)}^{int} \tag{51}$$

with

$$\boldsymbol{f}_{(e)}^{int} = \int_{\Omega_{(e)}} \boldsymbol{\partial N_X S F}^T d\Omega, \tag{52}$$

Here, $\boldsymbol{f}_{(e)}^{int} \in \mathbb{R}^{n \times D}$, where n is the number of nodes per element and D is the spatial dimensionality of the problem, and the operation described in (56) is in fact an assembly, rather than a direct summation and the deformation gradient $\boldsymbol{F}$ is computed from

$$\boldsymbol{F} = \boldsymbol{I} + \boldsymbol{U}_{(e)}^{T} \boldsymbol{\partial N_X}. \tag{53}$$

$\boldsymbol{U}_{(e)}$ is an n-by-D matrix form of the element nodal displacements. $\Omega_{(e)}$ is the undeformed element domain (equivalent to an area and volume in 2D and 3D, respectively). The second Piola-Kirchhoff stress $\boldsymbol{S}$ is evaluated from an appropriate constitutive relation, for example for the hyperelastic case:

$$\boldsymbol{S} = \frac{\partial W(\boldsymbol{E})}{\partial \boldsymbol{E}}, \tag{54}$$

where W is a strain energy potential and

$$\boldsymbol{E} = (\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{I})/2 \tag{55}$$

is the Green-Lagrange strain.

Mass and damping matrices

As for the internal forces, the mass matrix $\boldsymbol{M}$ for the system is obtained as an assembly of the element mass matrices:

$$\boldsymbol{M} = \sum_{e} \boldsymbol{M}_{(e)}, \tag{56}$$

with the consistent mass matrix $\boldsymbol{M}_{(e),}$, computed as

$$\boldsymbol{M}_{(e)} = \int_{\Omega_{(e)}} \rho \boldsymbol{N}^T \boldsymbol{N} \; d\Omega, \tag{57}$$

where $\boldsymbol{M}_{(e)} \in \mathbb{R}^{nD \times nD}$, nD is the product of the number of nodes n and degree of freedom D, and ρ is the mass density. In practice, a lumped form of $\boldsymbol{M}$, in which $\boldsymbol{M}$ is rendered diagonal, is usually adopted in explicit analyses, since this greatly simplifies the solution of (48) and leads to a far more efficient update formula for nodal displacements (i.e. the multiplication with the inverse of $\boldsymbol{M}$ is easily obtained since all off-diagonal values are zero).

Mass lumping may be achieved in different ways; in the present work row summation is adopted, in which the entries of each row are summed at the corresponding diagonal and all other values become zero. In practice, this structure can be achieved efficiently by computing the mass of each element directly and distributing this evenly to each element's nodes. So, for a given element e, with mass m, diagonal entries are computed as:

$$M_{(e)ii} = \frac{m}{n} \, , \tag{58}$$

Also, for computational ease, the mass-proportional Rayleigh damping is employed. With damping coefficient α, we have:

$$\boldsymbol{C} = \alpha \boldsymbol{M}. \tag{59}$$

4.3.2 Explicit Time Integration Method

To obtain the displacement solutions $\boldsymbol{u}_t$ at each time point t, the FE equations are discretized in time using the central difference method, which is an explicit time integration method. With this method, the incremental update formula only involves information from previous and current time steps:

$$\boldsymbol{u}_{t+\Delta t} = \gamma_1 \boldsymbol{M}^{-1}\left(\boldsymbol{f}_t^{ext} - \boldsymbol{f}_t^{int}\right) - \gamma_2 \boldsymbol{u}_t + \gamma_3 \boldsymbol{u}_{t-\Delta t}, \tag{60}$$

where Δt is the time step size, and subscripts have been added to indicate time points: $t + \Delta t$, t and $t - \Delta t$ are next, current and previous time steps, respectively. Coefficients γ_{1-3} are functions of the step size and damping parameter: $\gamma_1 = 2\Delta t^2/(\alpha\Delta t + 2)$, $\gamma_2 = 4/(\alpha\Delta t + 2)$, and $\gamma_3 = 1 - 4/(\alpha\Delta t + 2)$. As suggested earlier, the use of mass lumping, which turns $\boldsymbol{M}$ into a diagonal matrix, makes the above formula very cheap to evaluate, since the $\boldsymbol{M}^{-1}$ term is then trivially computed.

Explicit methods are only conditionally stable [80]: the chosen time step must be less than some critical value ($\Delta t \leq \Delta t_{cr}$), which is a function of the maximum natural frequency ω_{max} of the system :

$$\Delta t_{cr} = \frac{2}{\omega_{max}}. \tag{61}$$

For elements with homogeneous strain, Belytschko redefined Δt_{cr} as the time taken by a sound wave with velocity c to propagate through the element of a mesh with the shortest characteristic element length L_e:

$$\Delta t_{cr} = \frac{L_e}{c} = \frac{L_e\sqrt{\rho}}{\sqrt{\lambda + 2\mu}}, \tag{61′}$$

where λ and μ are Lamé parameters [82]. Due to the danger of round off errors, Cook proposed that the chosen time step should in practice not be greater than $0.75\Delta t_{cr}$ [83].

Although, the explicit method is conditionally stable, parallelization of the algorithm gives it an advantage over implicit methods for acceleration of the method via GPU implementation [67], [84].

4.3.3 TLED algorithm

Based on eq. (61) and associated equations, TLED algorithm is given in Table 4-1 [84]

Table 4-1: Algorithm 1 of TLED formulation

Initialization $\Delta t \leftarrow$ choose a time step using (58′) $\boldsymbol{u}_{t-\Delta t}, \boldsymbol{u}_t \leftarrow \mathbf{0}$ Precompute lumped element mass $\boldsymbol{M}_{(e)}$ **for all** e ∈ ElementSet **do** $\boldsymbol{\partial N} \leftarrow$ compute shape function derivatives using (50) $M_{(e)ii} \leftarrow$ compute element diagonal nodal mass entries using (58) **end** $\boldsymbol{M} \leftarrow \sum_e \boldsymbol{M}_{(e)}$ [Assemble system mass matrix] (56)

Time integration
for all TimeSteps **do**
for all e ∈ ElementSet **do**
$\boldsymbol{F} \leftarrow$ compute deformation gradient using (53)
$\boldsymbol{S} \leftarrow$ compute 2nd Piola-Kirchhoff stress using (54)
$\boldsymbol{f}_{(e)}^{int} \leftarrow$ compute internal forces using (52)
end
$\boldsymbol{f}^{int} \leftarrow \sum_e \boldsymbol{f}_{(e)}^{int}$ [Assemble global internal force vector] (51)
$\boldsymbol{f}_{ext} \leftarrow$ apply external loads
$\boldsymbol{u}_{t+1} \leftarrow$ compute next incremental displacement using (60)
Apply constraint for the next time step
$\boldsymbol{u}_{t+1} \leftarrow$ apply displacement constraint for the next time step
$\boldsymbol{u}_{t-1} \leftarrow \boldsymbol{u}_t$
$\boldsymbol{u}_t \leftarrow \boldsymbol{u}_{t+1}$
end

4.4 Conclusion

Presented in this section is the TLED method which is based on the variational formulation. The simplest form of variational approach, principle of virtual work is chosen to obtain the weak form of the differential equilibrium equations. With the

weak form, solution could easily be obtained for simple and complex boundary conditions, since, the weak form is satisfied in the integral sense, as opposed to the strong form. To solve the resulting differential equations, the well-established finite element method was chosen and formulated using the total Lagrangian (TL) formulation, while explicit time integration method was chosen for time integration of the equations. The TL formulation ensures that element shape functions derivatives remain unchanged throughout the analysis and thus can be precomputed; a particularly useful feature for simulating real time soft tissue incision, which is limited by a haptic feedback response of about 1kHz. On the other hand, the explicit method which although is conditionally stable, has the advantage over the implicit integration method of parallelization of the algorithm making the hybrid method, TLED suitable for acceleration via GPU implementation and for achieving real time simulation. Thus, the TLED method presented here will serve as the basis for the extended numerical method presented in the next chapter.

5 EXTENDED TOTAL LAGRANGIAN EXPLICIT DYNAMICS (XTLED)

An extended total Lagrangian explicit dynamics method (XTLED) is presented here for interactive, physically based simulation of surgical incision of soft tissues. The standard formulation described in chapter 3 can in principle be used to simulate incision, but this would require repeated updating of the mesh to ensure it conforms with the evolving discontinuity. This process, combined with the requirement for nonlinear solution procedures, produce algorithms that are prohibitively slow for interactive simulations. Therefore, an enrichment-based approach which addresses both issues is presented. The algorithm for the XTLED is verified by comparing numerical results with FEM analysis and the method is also validated using a silicon sample which is cut and which deforms under tension, causing a Mode I deformation (opening of the incision).

5.1 Extended Finite Element Methods (XFEM)

As described in the standard TLED formulation, the analysed structures are assumed to be continuous, leading to the displacement approximation (50). However, in the presence of a cut, this assumption no longer holds since the displacement field becomes discontinuous. Such discontinuity is modelled here using an extended finite element method (XFEM) approach. The superiority of XFEM emanates from the lack of need to modify the finite element mesh to conform to the cut, or to geometrically model or mesh the cut, making it computationally efficient in comparison to other mesh based approaches.

5.1.1 XFEM Discretization: Shape function enrichment

XFEM is based on the principle of partition of unity; the summation of a set of functions evaluated at a given position equals unity:

$$\sum_{i=1}^{n} f_i(\boldsymbol{x}) = 1 \tag{62}$$

Based on this, the idea of enriching the standard FEM shape functions with additional functions that reflect specific physical phenomena, such as a cut-associated discontinuity, arises. The enriched displacement approximation can be written as

$$\begin{aligned} \boldsymbol{u}^h(\boldsymbol{x}) = \boldsymbol{u}^{FE}(\boldsymbol{x}) + \boldsymbol{u}^{enr} &= \sum_{i=1}^{n} N_i(\boldsymbol{x})\boldsymbol{u}_i + \sum_{j=1}^{n_{enr}} N_j(\boldsymbol{x})\psi_j(\boldsymbol{x})\boldsymbol{a}_j \\ &= \widetilde{\boldsymbol{N}}(\boldsymbol{x})\boldsymbol{u}_{(e)} \end{aligned} \tag{63}$$

where $\psi_j(\boldsymbol{x})$ is the discontinuous enrichment function, $\boldsymbol{a}_j$ is a vector of additional DOFs for node j, analogous to $\boldsymbol{u}_i$, $\widetilde{\boldsymbol{N}} = [N_1\boldsymbol{I} \quad N_2\boldsymbol{I} \quad \cdots \quad N_n\boldsymbol{I} \quad \psi_1 N_1\boldsymbol{I} \quad \psi_2 N_2\boldsymbol{I} \quad \cdots \quad \psi_n N_{n_{enr}}\boldsymbol{I}]$ is a matrix of standard and enriched shape function terms, and $\widetilde{\boldsymbol{u}}_{(e)} = [\boldsymbol{u}_1^T \quad \boldsymbol{u}_2^T \quad \cdots \quad \boldsymbol{u}_n^T \quad \boldsymbol{a}_1^T \quad \boldsymbol{a}_2^T \quad \cdots \quad \boldsymbol{a}_n^T]^T$ collects displacement vectors for standard and enriched DOFs for element e .

5.1.2 Enrichment function

The choice of the enrichment function depends on the type of discontinuity. For example, for a discontinuity such as a crack, the Heaviside and asymptotic crack tip enrichments are applied to the crack path and crack tip respectively. The enrichment type adopted in many surgical simulation literature [85], [86] and used here for modelling a cut is the Heaviside function:

$$H(\boldsymbol{x}) = sign(\boldsymbol{x}) = \begin{cases} 1 \\ -1 \end{cases} \tag{64}$$

To fulfil the Kronecker delta properties, that is to ensure that the discontinuity vanishes beyond nodes that are not within the vicinity of the discontinuity, a shifted Heaviside function was proposed as

$$\psi_j(\boldsymbol{x}) = \frac{H(\boldsymbol{x}) - H_j}{2}, \tag{65}$$

where

$$H(\boldsymbol{x}) = \sum_{j}^{n} N_j(\boldsymbol{x}) H_j. \tag{66}$$

5.2 XTLED Formulation

5.2.1 Shape Function Derivative

For elements containing the discontinuity (enriched elements) enriched shape function derivatives matrices $\boldsymbol{\partial \widetilde{N}_X}$ may be defined as

$$\boldsymbol{\partial \widetilde{N}_X} = \begin{bmatrix} \boldsymbol{\partial N} \\ \boldsymbol{\partial N}^{enr} \end{bmatrix}, \tag{67}$$

where the corresponding enriched shape function is

$$\boldsymbol{\partial N}^{enr} = \begin{bmatrix} \psi_1 \dfrac{\partial N_1}{\partial x} & \psi_1 \dfrac{\partial N_1}{\partial y} & \psi_1 \dfrac{\partial N_1}{\partial z} \\ \psi_2 \dfrac{\partial N_2}{\partial x} & \psi_2 \dfrac{\partial N_2}{\partial y} & \psi_2 \dfrac{\partial N_2}{\partial z} \\ \vdots & \vdots & \vdots \\ \psi_n \dfrac{\partial N_n}{\partial x} & \psi_n \dfrac{\partial N_n}{\partial y} & \psi_n \dfrac{\partial N_n}{\partial z} \end{bmatrix} \tag{68}$$

Dependences on $\boldsymbol{x}$ have been omitted.

It should be noted that the shape function derivatives of elements adjacent to fully enriched elements remain unenriched (for instance, element A), since enrichment is only for the purpose of modelling the discontinuity within cut elements (Figure 5-2). As such the enrichment values at and beyond nodes of cut elements should be zero.

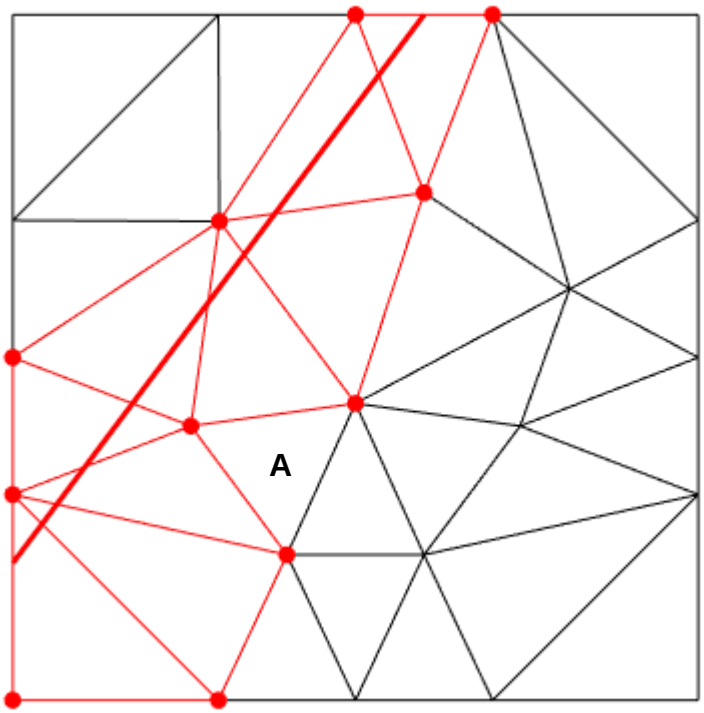

Figure 5-1: Discretised geometry showing enriched elements and nodes (red)

Since the internal forces $\boldsymbol{f}^{int}$ and mass matrix $\boldsymbol{M}$ are formulated in terms of the element shape functions, new formulae are needed to reflect the enrichment of these variables, as detailed in the following sections.

5.2.2 Enriched Internal Force

The enriched element force matrix can be computed as:

$$\tilde{\boldsymbol{f}}_{(e)}^{int} = \int_{\Omega_{(e)}} \boldsymbol{\partial} \tilde{\boldsymbol{N}}_{\boldsymbol{X}} \tilde{\boldsymbol{S}} \tilde{\boldsymbol{F}}^{T} d\Omega, \tag{69}$$

where

$$\widetilde{\boldsymbol{F}} = \boldsymbol{I} + \widetilde{\boldsymbol{U}}_{(e)}^{T} \boldsymbol{\partial} \widetilde{\boldsymbol{N}}_{\boldsymbol{X}}, \tag{70}$$

is the enriched deformation gradient, with $\widetilde{\boldsymbol{U}}_{(e)}$ expressed as

$$\widetilde{\boldsymbol{U}}_{(e)} = \begin{bmatrix} \boldsymbol{U}_{(e)} \\ \boldsymbol{A}_{(e)} \end{bmatrix}, \tag{71}$$

where $\boldsymbol{A}_{(e)} \in \mathbb{R}^{n^{enr} \times D}$ is a matrix form of the additional degrees of freedom associated to an enriched element, analogous to $\boldsymbol{U}_{(e)}$. Enriched Green-Lagrange strain and 2nd Piola-Kirchhoff stresses follow as

$$\widetilde{\boldsymbol{E}} = \left(\widetilde{\boldsymbol{F}}^{T} \widetilde{\boldsymbol{F}} - \boldsymbol{I}\right)/2 \tag{72}$$

and

$$\tilde{\boldsymbol{S}} = \frac{\partial W(\widetilde{\boldsymbol{E}})}{\partial \widetilde{\boldsymbol{E}}}, \tag{73}$$

respectively. Global force, $\boldsymbol{f}^{int}$ is then assembled as:

$$\boldsymbol{f}^{int} = \sum_{e} \boldsymbol{f}_{(e)}^{int} + \sum_{e} \widetilde{\boldsymbol{f}}_{(e)}^{int} \tag{74}$$

5.2.3 Enriched Mass

As for the standard formulation, a lumped mass matrix is desirable. For enriched elements, this takes the form:

$$\widetilde{\boldsymbol{M}}_{(e)} = \begin{bmatrix} \boldsymbol{M}_{(e)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{(e)}^{enr} \end{bmatrix}, \tag{75}$$

where $\boldsymbol{M}_{(e)}$ is computed as in (11), and $\boldsymbol{M}_{(e)}^{enr} \in \mathbb{R}^{n^{enr} \times n^{enr}}$ is a diagonal matrix whose non-zero components may be computed efficiently using

$$M_{(e)ii}^{enr} = \frac{\rho}{n} \int_{\Omega_{(e)}} \psi_i^2 \, d\Omega. \tag{76}$$

Global mass is then assembled as:

$$\boldsymbol{M} = \sum_{e} \boldsymbol{M}_{(e)} + \sum_{e} \widetilde{\boldsymbol{M}}_{(e)} \tag{77}$$

As described in Wu et al., this form is preferable to either a consistent formulation or a lumped matrix produced by row summation, since its effect on the critical time step is comparatively small (and, certainly, Δt_{cr} never approaches zero, as it does in the consistent matrix case) [21]. The stability and associated computational cost of the algorithm are correspondingly similar to those of the standard TLED formulation.

The complete XTLED algorithm is summarised in Table 5-1

Table 5-1: Algorithm 2 of XTLED formulation.

Initialization $\Delta t \leftarrow$ choose a time step using (58′) $\boldsymbol{u}_{t-\Delta t}, \boldsymbol{u}_t \leftarrow \boldsymbol{0}$ cutElement ← Compute list of enriched elements Assign Heaviside values to each nodes of the cut element by computing signed distances Precompute lumped element mass $\boldsymbol{M}$ and shape function derivative $\boldsymbol{\partial N}$ **for all** e ∈ ElementSet **do** $\boldsymbol{\partial \tilde{N}_X} \leftarrow$ compute shape function derivatives using (67) $M_{(e)ii} \leftarrow$ compute element diagonal nodal mass entries using (58) **if** e ∈ EnrElementSet $M^{enr}_{(e)ii} \leftarrow$ compute enriched element diagonal nodal mass entries using (76) **End** $\boldsymbol{M}^{enr}_{(e)} \leftarrow$ compute enriched element diagonal nodal mass entries $\widetilde{\boldsymbol{M}}_{(e)} \leftarrow$assemble element mass (75) **end** $\boldsymbol{M} \leftarrow \sum_e \widetilde{\boldsymbol{M}}_{(e)} + \sum_e \boldsymbol{M}_{(e)}$ [Assemble enriched system mass matrix] (77)

Time integration
for all TimeSteps **do**
for all e ∈ ElementSet **do**
if e ∉ EnrElementSet
$\boldsymbol{F} \leftarrow$ compute deformation gradient using (53)
$\boldsymbol{S} \leftarrow$ compute 2nd Piola-Kirchhoff stress using (54)
$\boldsymbol{f}^{int}_{(e)} \leftarrow$ compute internal forces using (52)
else
$\tilde{\boldsymbol{F}} \leftarrow$ compute enriched deformation gradient using (70)
$\tilde{\boldsymbol{S}} \leftarrow$ compute enriched 2nd Piola-Kirchhoff stress using (8)
$\tilde{\boldsymbol{f}}^{int}_{(e)} \leftarrow$ compute enriched internal forces using (69)
end
end
$\tilde{\boldsymbol{f}}^{int} \leftarrow \sum_e \tilde{\boldsymbol{f}}^{int}_{(e)} + \sum_e \boldsymbol{f}^{int}_{(e)}$ [Assemble enriched global internal force vector] (74)
$\boldsymbol{R} \leftarrow$ apply external loads
$\boldsymbol{u}_{t+1} \leftarrow$ compute next incremental displacement using (60)
Apply constraint for the next time step
$\boldsymbol{u}_{t+1} \leftarrow$ apply displacement constraint for the next time step
$\boldsymbol{u}_{t-1} \leftarrow \boldsymbol{u}_t$
$\boldsymbol{u}_t \leftarrow \boldsymbol{u}_{t+1}$
End

5.2.4 Evaluation of Force and Mass Integrals

As in the standard finite element case, the force eq. (69) and mass integrals (eq. 76) must be evaluated efficiently and accurately. Gauss integration is efficient but is only exact for polynomial integrands. Nonetheless, it may give an accurate approximation if integrands are smooth and continuous. Therefore, in the present case, each cut

element, with domain $\Omega_{(e)}$, is partitioned into contiguous, non-overlapping, and geometrically simple subdomains $\Omega_{(e)}^{i}$, i.e. $\bigcup_i \Omega_{(e)}^{i} = \Omega_{(e)}$. Importantly, the subdomains conform to the cut, ensuring the integrands are continuous on each $\Omega_{(e)}^{i}$. Figure 5-2 shows an example of an element with integration points prior to and after cutting; the subdomains are thus created to align to the edges of the cut.

The integrals are then evaluated piecewise over the subdomains of the cut elements, where the integrands indeed satisfy these requirements. The integrals become:

$$\int_{\Omega_{(e)}} (\cdot)\, d\Omega = \sum_{i} \int_{\Omega_{(e)}^{i}} (\cdot)\, d\Omega. \tag{78}$$

Element subdomains are formed by finding the element-incision intersection points and constructing a Delaunay triangulation from the combined element nodal coordinates $\boldsymbol{x}_i$ $(i = 1 \ldots n)$ and intersection point coordinates.

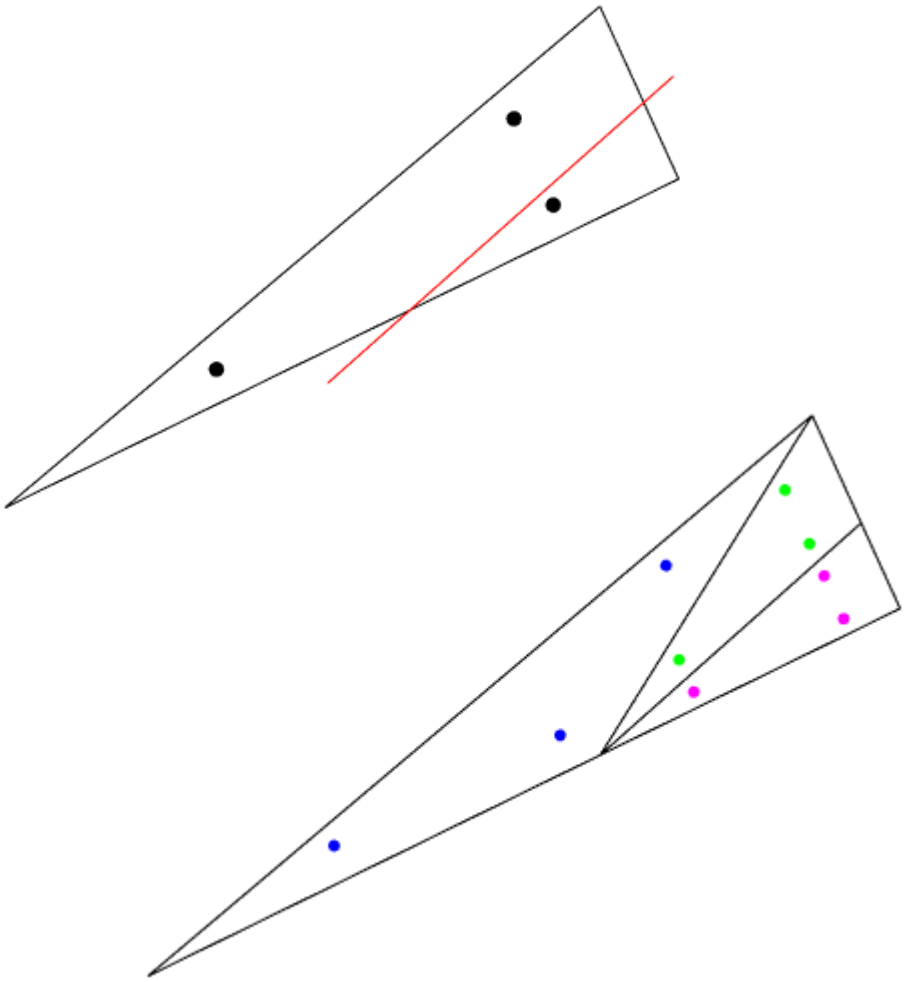

Figure 5-2: a) Triangular element with showing 3-point integration points before (top) and after (bottom) incision (red line)

The location of the Gauss points in the integration cells are obtained by isoparametric mapping using the standard displacement interpolation function. The algorithm is given in Table 5-2: Algorithm for computing the integration points of elements intersected by a cut.

Table 5-2: Algorithm for computing the integration points of elements intersected by a cut

1. Intersection points using Fast Triangle Intersection [87]
2. Loop through the elements and compute signed distances of nodes from the cut. An additional condition is added to eliminate elements whose product of signed distances is positive. (this implies that the element is not cut)

3. Loop through the edges of all cut elements; determine if any edge is cut; compute intersection points

for all $i \in$ nEdg **do**
  $\phi_{node\ 1} \leftarrow$ Get the distance of node 1 of the i^{th} edge
  $\phi_{node\ 2} \leftarrow$ Get the distance of node 2 of the i^{th} edge
  If $\phi_{node\ 1} \times \phi_{node\ 2} < 0$ [i.e. the cut intersects the edge]
    $\mathrm{r} \leftarrow \phi_{node,i}/(\phi_{node,i} - \phi_{node,i+1})$ [compute ratio]
    $\mathrm{L_{IntPnt}} \leftarrow (1-\mathrm{r})\mathrm{coord}_{node,i} + (\mathrm{r})\mathrm{coord}_{node,i}$ [get intersection points using interpolation function]
    N ← compute shape function of the nodes at the intersection point, L_IntPnt
    intCoord ← $\sum_n \mathrm{N}^T \mathrm{coord}$ [get global intersection point]
    coord ← [coord, intCoord] [store point]
    localCoord ← [coord, L_lntPnt]
  end
end

4. Using local intersection points, localCoord, Delaunay triangulation in MATLAB, subdivide elements into integration cells. Then delete cells with one subdomain

   $Cells \leftarrow$ Get integration cells

5. Get the standard Gauss points (q, w) for a 3-point integration for a tetrahedral
[q, w] ← Gauss

6. Get Gauss points within integration cells by mapping
for all $nC \in nCells$ **do**
  L_coordCell ← Get the local nodal coordinates of the integration cell
  G_coordCell ← Get the global nodal coordinates of the integration cell
  for all $i \in N_{\boldsymbol{intPts}}$ **do**
    N ← compute shape function of the nodes at the integration point, q_i
    gpCell ← $\sum_n \mathrm{N}^T \times$ L_coordCell [compute Gauss points for the integration cell]
    $\mathrm{gpCell}_{nCells \times dof} \leftarrow$ gpCell
    detJ ← get determinant of Jacobian
    $\mathrm{detJ}_{nCells \times 1} \leftarrow$ detJ
  end
end

Although a tolerance value was used when selecting the elements that were cut, the use of Delaunay triangulation as an additional constraint to delete all elements which returned only one cell after triangulation ensured the eliminated of unenriched

elements. This is very important, since zero entries in the mass matrix implies an infinite displacement solution (that is division by zero).

5.3 Numerical Analysis

The suitability of the XTLED for creating incisions of nonlinear soft tissue was verified by conducting numerical tests, using cuboid and brain sample with soft tissue properties. XTLED was also validated via experimental tensile tests using a silicon strip. The tensile test displacement results obtained via 2D digital image correlation technique was compared with results from FEM and XTLED. FEM simulations were conducted using Abaqus/CAE 6.14-2.

5.4 Verification of XTLED using 3D Geometries

5.4.1 Cuboid Sample: Numerical Simulation Details

Cuboid samples with horizontal and kinked edge cuts are presented here. The former sample is displaced by 22cm at the top surface and fixed at the bottom face, while the latter sample with kinked cut is displaced by 3mm at the top surface and fixed at the bottom surface (Figure 5-3). XTLED and FEM results begins with Figure 5-4.

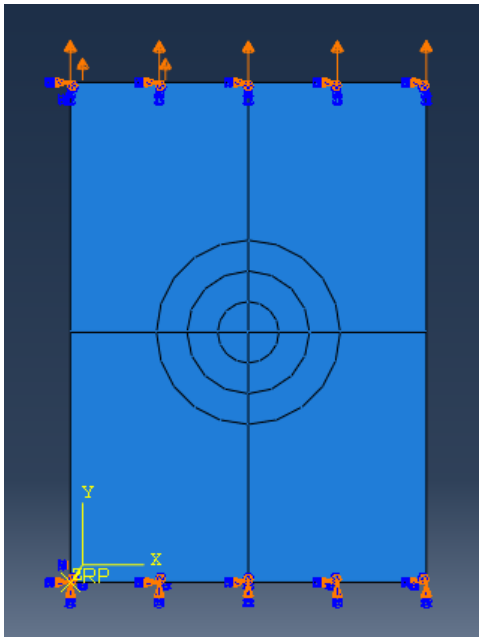

Figure 5-3: Cuboid with horizontal cut, fixed in all DOF at the bottom surface and displaced by 22mm at the top surface

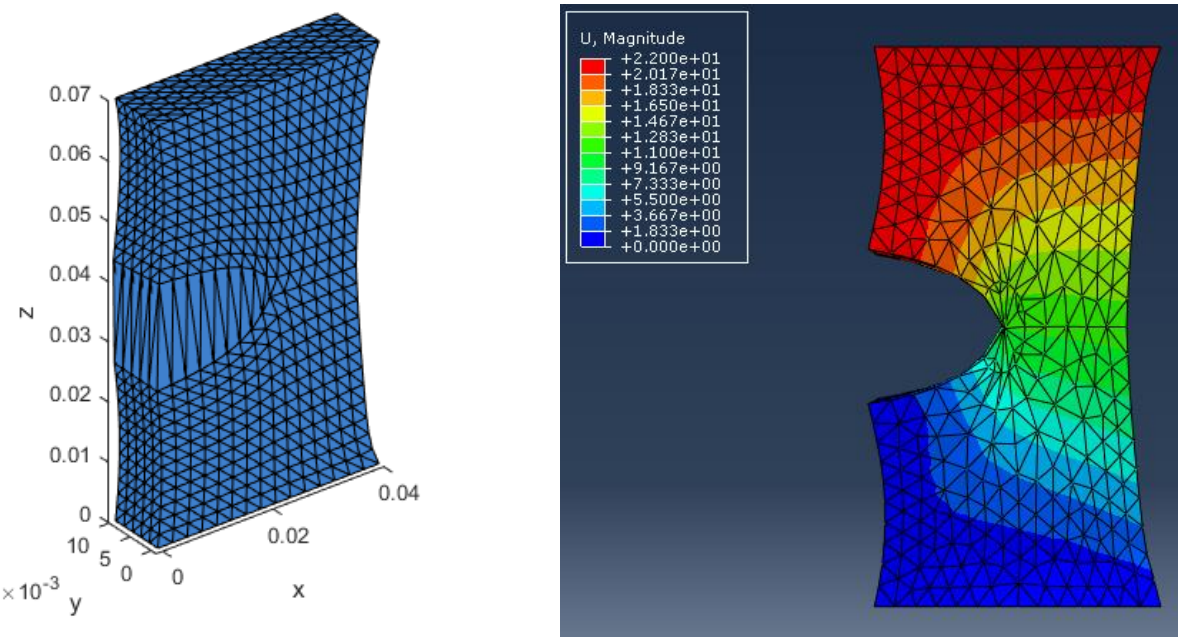

Figure 5-4: XTLED (left) and FEM (right) result of deformed cuboid with horizontal cut

To be sure that deformation is as expected: there are no volumetric locking effects due to the tetrahedral elements, the sample in Abaqus was discretized using wedges (C3D6) at the cut tip (C3D6) and hexahedral elements (C3D8R), elsewhere (Figure 5-5). Results of the displacement contour of cuboid sample with kinked cut is shown in Figure 5-6.

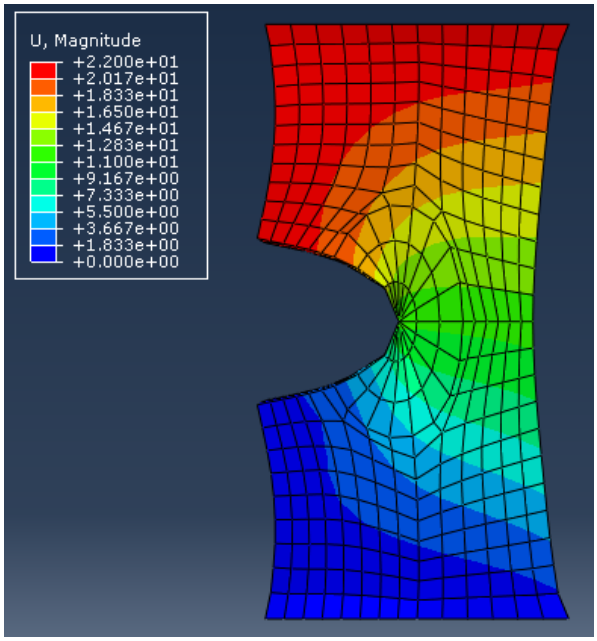

Figure 5-5: Cuboid sample with hexahedral and wedge elements

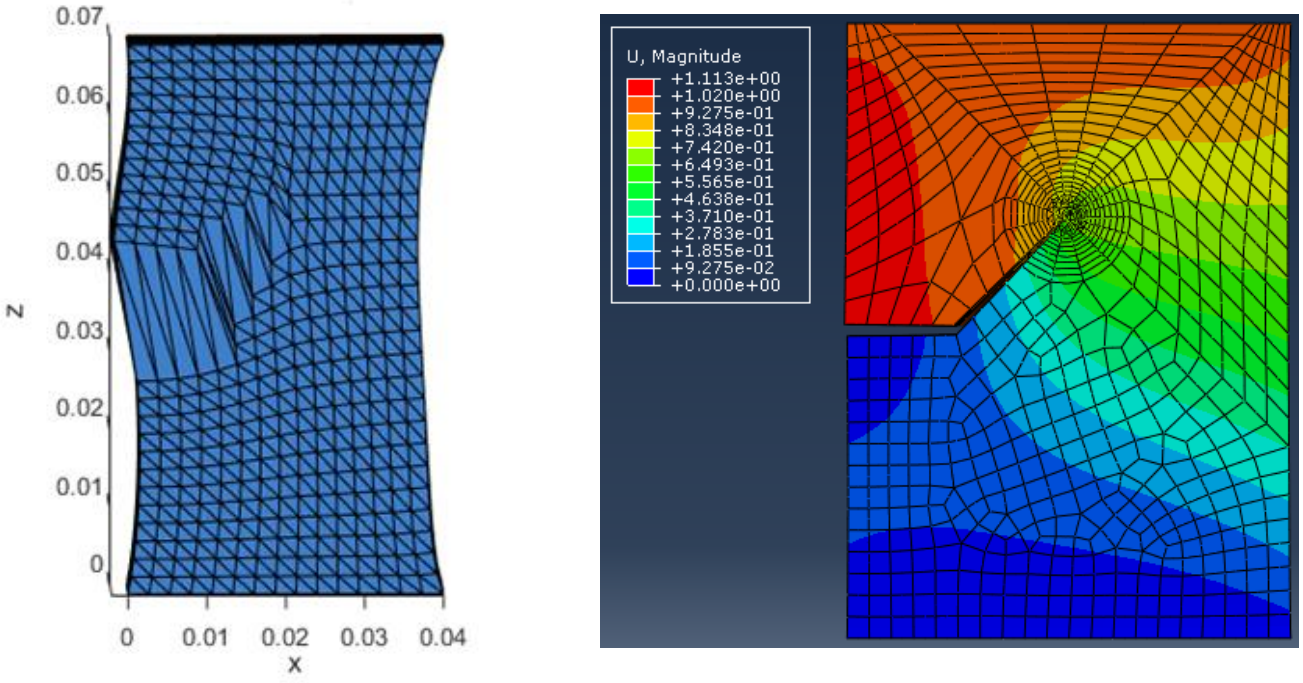

Figure 5-6: XTLED (left) and FEM (right) results of deformed cuboid with kinked cut

Comparison of the cut opening of each sample above are presented in Figure 5-13.

5.5 Verification of XTLED using Brain sample

Further verifications were attempted using soft tissues such as a brain model. The brain model used was obtained from Taylor et al. [75] (Figure 5-7). The brain model was constraint at the brain stem and displaced at the frontal lobe by 0.01m and the analysis type was nonlinear analysis that considered change in geometry. The brain was modelled as a hyperplastic, Neo-Hookean material with Lame parameter, $\lambda = 49329\text{Pa}$, $\mu = 1007\text{Pa}$, for a brain with stiffness value $E = 3000\text{Pa}$ and Poisson ration of 0.49. The brain was discretized into meshes of 6000 to 39323 elements. For FEM, the cut was conducted using the contour integral method used for crack analysis. For XTLED, the cut plane was extracted from the FEM model and its dimensions were

used to create a distance map from which the location of cut was determined; and intersection algorithm with the MATLAB code were used to enrich the elements intersected by the cut.

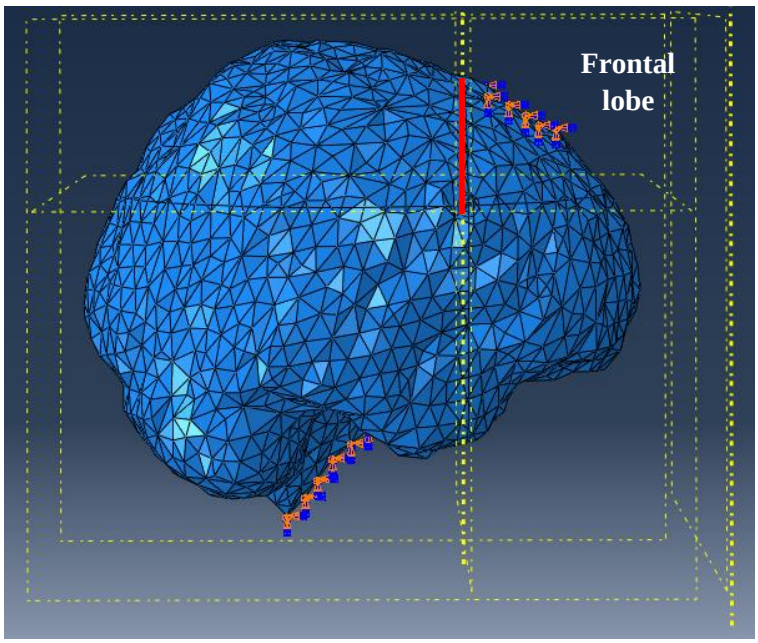

Figure 5-7: Brain sample constrained at the stem and displaced at frontal lobe with cut along red line for Mode deformation

5.5.1 Simulation Results of a Brain Sample

The XTLED simulation of the brain with a cut was implemented but a solution could not be obtained due to complex number errors. Possible reasons for this error include warped or ill conditioned elements that arise during deformation of the brain sample, especially in regions of sharp corners where the tetrahedral elements are significantly small in comparison to other elements in the brain. It is also likely that the errors were caused by volumetric locking due to the use of fully integrated tetrahedral elements [36]. Another reason could be that the applied displacement constraint was too high.

It is stated in Taylor et al. [75] that a displacement of 20% was applied for soft tissue deformation not induced by cutting.

5.6 Numerical Validation of XTLED: Silicon Strip Sample

XTLED method was validated by comparing experimental and FEM results of a tensile test on silicon strip to results from XTLED.

5.6.1 Simulation Details of Silicon Strip Sample

DIC technique and sample set up

DIC technique was used to determine the displacements of the silicon sample. The DIC technique is an optical technique which measures displacements of a sample by tracking points or features on the sample over time, then assigning and comparing the positions of the points to a predetermined coordinate system. The method used is called cross correlation method.

To measure displacement of points on the sample, a recognizable pattern on the sample is required to create a contrast across the sample to enable distinguished recognition of points or regions, for displacement measurement. This pattern is usually referred to as a speckle pattern. The pattern was created by spraying the sample with aerosol spray (Figure 5-8); other methods of creating a speckle pattern include brush flicking, airbrushing and rubber stamp.

Figure 5-8: The creation of speckle pattern on sample

The equipment used for the DIC technique typically include light source, two cameras mounted on an adjustable stand (Figure 5-9).

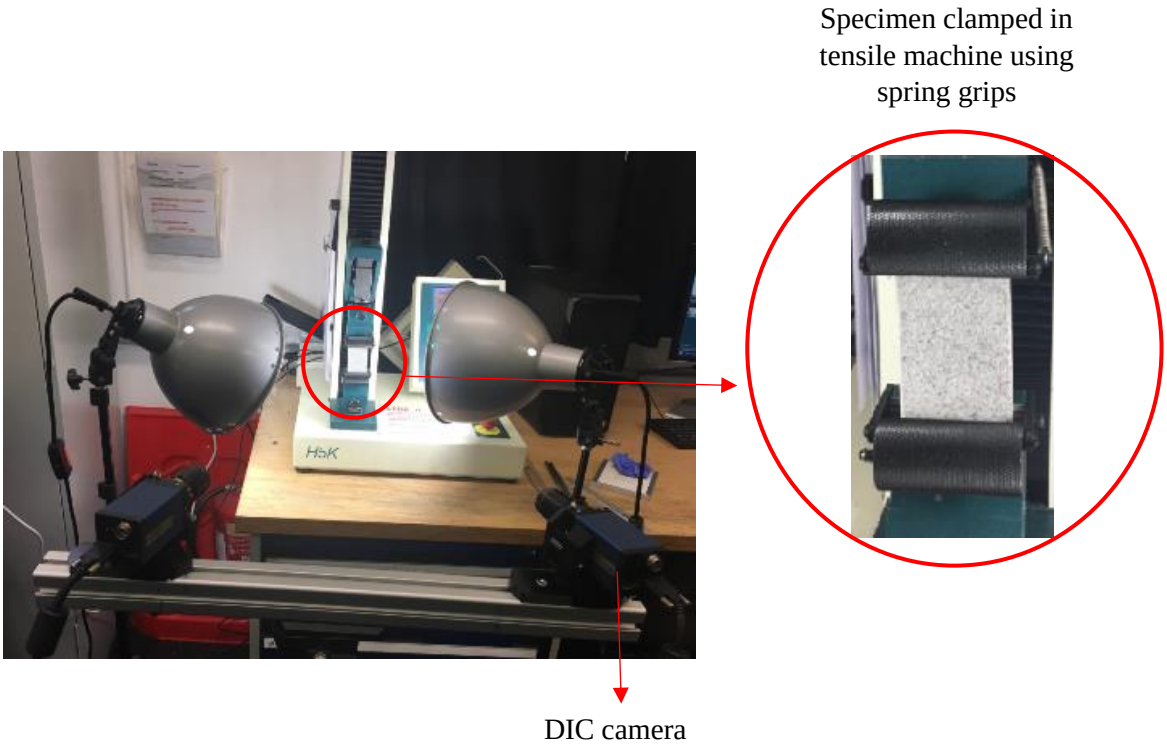

Figure 5-9: Sample set up for tensile experiment and DIC technique

Both cameras and the light were set up such that the images obtained from the cameras were clear (no reflection due to light), sharp (good camera focus), and

included the area of interest on the sample (cut and region around it). These images were viewed using the accompanying DIC software. The system was then calibrated using a standard calibration target (Figure 5-10).

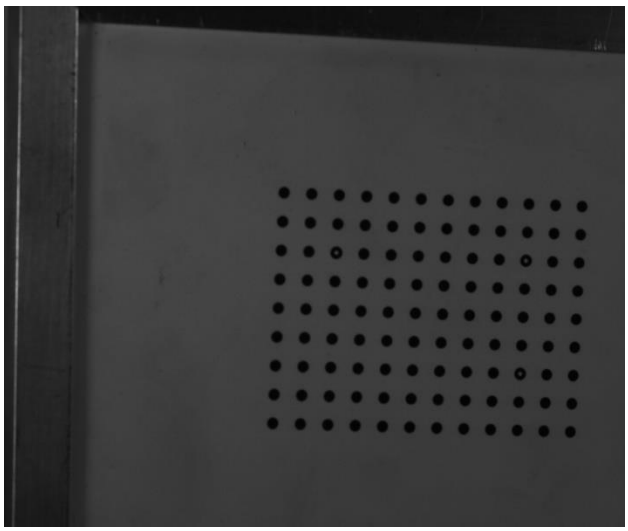

Figure 5-10: Calibration target

The target was placed in front of the camera, such that it filled the field of view of both cameras, to enable extraction of calibration images and points on images. The calibration target was placed in front of the specimen and tilted to obtain fifty differently oriented images from each camera; note that the number of images taken can vary. Images were taken by rotating calibration target in the vertical and horizontal axis and in-plane rotation. Calibration helps to "triangulate the cameras positions relative to one another and also calculates and corrects any lens distortions". The calibration pictures taken are used to determine a reference coordinate system and subsequent images are referred to that reference image. Depending on the software used, a calibration score could be generated to ascertain the quality of the calibration done and whether there is need to repeat calibration.

Images of the sample are then taken over the duration of a test. Thereafter, post-processing is done to obtain the displacements of the sample at any given time during the tensile test.

Tensile Test

Tensile tests were conducted on two silicon samples of dimension (100mm by 40mm by 10 mm) each (Figure 5-11). Sample T1 had a cut at mid-height and halfway along 40mm dimensions while sample T2 had a similar cut but with the cut inclined upwards at 45 degrees 10mm along 40mm dimension.

Figure 5-11: Silicon samples with horizontal, T1 (LHS) and kinked cut, T2 (RHS)

Both samples were fixed at the bottom surface but displaced at the top surface by 22mm and 3mm, respectively. Displacement was applied at a rate of 0.8mm/min and 2mm/min for T1 and T2, respectively. The displacement contour of the sample was obtained using the DIC technique as described above. Table 5-3 gives a summary of the samples.

Table 5-3: Sample specifications for DIC technique

Sample	Gauge dimension (mm)	Frame number	Applied displacement (mm)	Linear/Nonlinear
T1	56 x 40 x 10	2000	22	Nonlinear
T2	54 x 40 x 10	100	3	Linear
Images were captured every minute				

5.6.2 Results of Validation analysis

Contour plots of the displacement field for both samples as obtained from the DIC technique is shown in Figure 5-12. These results were compared to FEM and XTLED simulation results which were obtained using Abaqus/CAE 6.14-2 and MATLAB codes respectively.

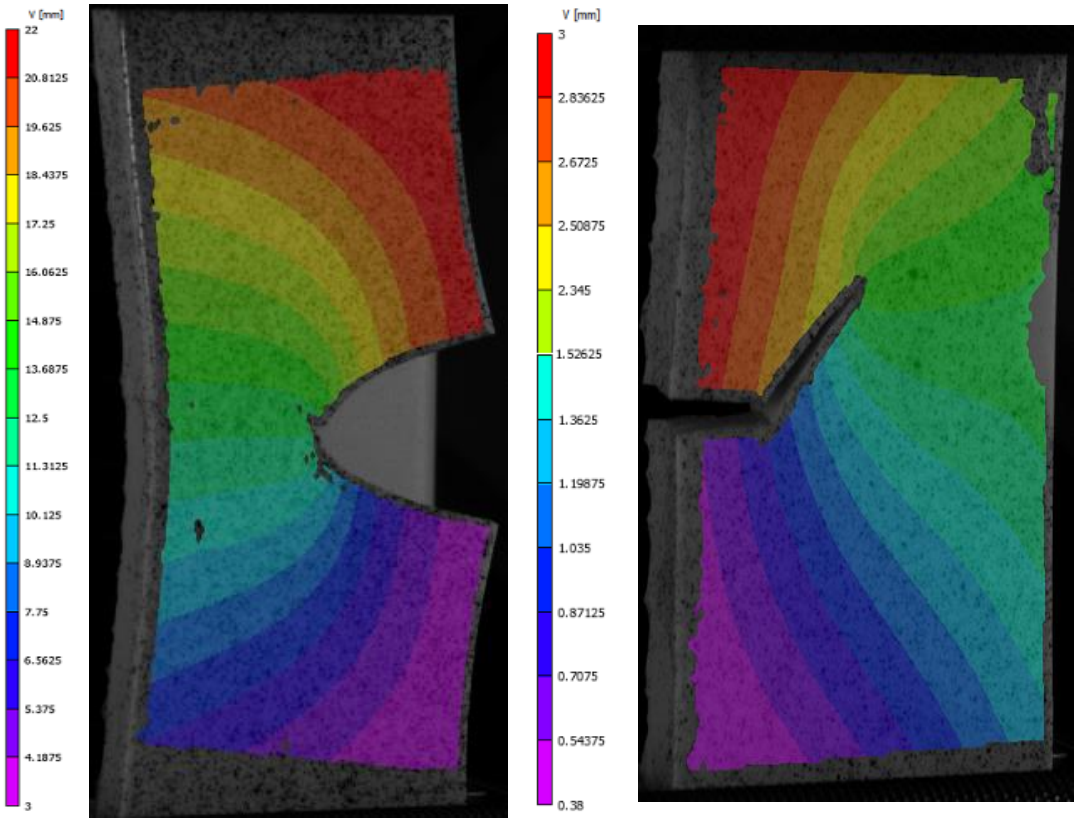

Figure 5-12: Vertical displacement contour plots of sample T1 (LHS), and T2 obtained from DIC analysis

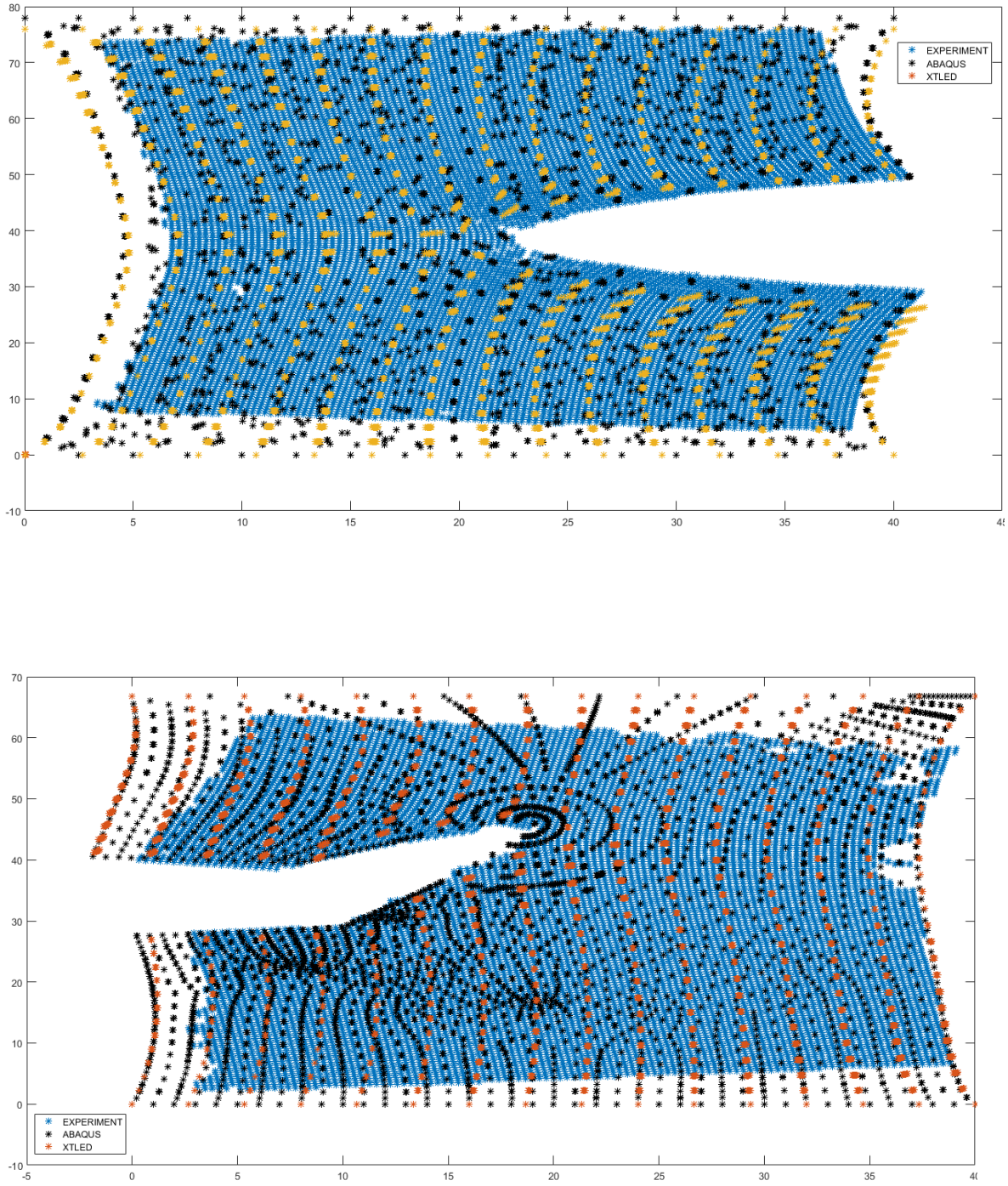

Figure 5-13:Comparison of T1 (top) and T2 (bottom) sample deformation obtained from XTLED, FEM and tensile experiment

5.7 Discussion and Conclusion

A cuboid and brain samples were simulated to verify the solution obtained from XTLED method while the silicon samples were used to validate XTLED method. The final displacement results of sample T1 using FEM method were obtained from a mesh of 1770 nodes, and 7867 tetrahedral elements, while results from XTLED were obtained using 2604 nodes, and 11,250 tetrahedral elements. Unlike in Abaqus FEM, where the mesh around the cut tip region could be more refined than other parts of the geometry, this feature is not present in the current version of XTLED. This could explain the substantial difference in the number of elements used in both methods. Nonetheless, the deformed shape of TI sample obtained from both methods agree, by visual inspection (Figure 5-4: XTLED (left) and FEM (right) result of deformed cuboid with horizontal cut, Figure 5-13). Similarly, displacement results of T2 were obtained from a mesh of 4679 nodes and 864 elements (hexahedral and wedge elements); while results from XTLED were obtained from 15750 nodes and 3544 elements. The difference in mesh can be attributed to the higher order of hexahedral elements and mesh refinement at cut tip.

A comparison of these results with those obtained from the tensile experiments was obtained by registering all three results about the cut tip Figure 5-13. By visual inspection of the T1 sample results, both XTLED and FEM methods conform to the deformation pattern obtained from the tensile experiment. It should be noted that while the scattered points in XTLED and FEM represent the nodal points within the mesh, that of the DIC experiment are sample points selected by DIC system. In addition, the points that form the cut opening in XTLED method are actually nodes of the elements that were intersected by the cut, thus, the cut opening shown here is not the actual cut opening but the edges of the elements adjacent to the elements that were cut; the cut

was created within elements in XTLED, unlike in FEM. This explains why most of the nodes around the cut tip for XTLED solution do not exactly match the cut opening from the experiment of FEM, although the deformation is the same across all three methods.

For a better quantitative analysis, a point-wise comparison of the sample results could not be computed. More recent 2D DIC software could be used to obtain specific points on the samples and this could then be compared to the results from XTLED and FEM. Theoretically, displacements of the cut opening of the samples from XTLED could be computed by finding the elements within which this coordinates/points belong and then using the enriched displacement interpolation function (eq. (63)) to obtain the displacement at this specific points. Since different element types were adopted in FEM and XTLED and only tetrahedral are present in the XTLED algorithm, points defining the cut opening would be used for the computation.

Similarly, the brain sample was simulated to verify XTLED method. However, not-zero-number (NAN) errors were obtained during the solution and the solution had to be aborted. It is not yet clear what the cause of these is since global mass matrix had no zero entries.

To compare the simulation time in XTLED and FEM, XTLED would need to be written and implemented using an executable language such as C++. Currently in MATLAB, majority of the computational time is spent creating subdomains and this is repeated twice in computing mass and force.

6 CONCLUSION

6.1 Thesis Outcome and Contribution

The focus of this thesis is the development of a suitable numerical method for surgical incision, which can potentially alleviate the compromise of accuracy for computational speed present in existing numerical methods. Towards this, the extended Lagrangian explicit dynamic method (XTLED) was developed in MATLAB. The development of XTLED meets the objectives stated in chapter 1: to develop a numerical method that enables arbitrary cutting of deformable soft tissue.

The stated objective presents a similar work as earlier authors, Jeřábková et al. [58] who also combined XFEM and total Lagrangian finite element formulation. However, in this thesis, detailed implementation of the method, specifically the method adopted to determine enriched elements is presented. This was achieved via intersection algorithm and the use of Delaunay triangulation to ensure that the global mass matrix had no zero entries. This aspect is essential, in my opinion, otherwise the displacement solution obtained becomes an infinite solution.

The outcome of the above objective is as follows:

- Despite the conditional stability of the explicit time integration method, it is possible to create arbitrary cut in a cuboid sample subjected to tension: fixed at one end and extended along an opposite face by at least 30% of its dimension along the stretch. This was validated using a silicon strip as presented in Chapter 5. Thus, XTLED method is a potential method for simulating interactive surgical incision.

6.2 Limitations and Further Studies

The element type used in XTLED for simulating incisions is tetrahedron. These elements are known to exhibit volumetric locking for incompressible material. However solutions such as average nodal pressure have been reported to alleviate this issue [36]. Thus, a natural first step would be the incorporation of ANP with XTLED. While this might increase computational time, investigations would have to be carried out to determine how significantly this affects computational time.

A second limitation of the presented method, for which further studies are required, is that it only simulates static incision. Thus, for interactive purposes, XTLED should be adapted for interactive simulation. This could be achieved via the level set method which has been used to track discontinuities such as crack [88], [89].

A recommendation is the acceleration of XTLED via GPU implementation and verification of the method for interactive simulation. Here the implementation could be incorporated in an open source surgical simulating software such as SOFA (sofa.org). Such a simulation will give a clear indication of the use of the presented method for interactive surgical incision.

7 REFERENCES

[1] R. M. Satava, “Historical review of surgical simulation--a personal perspective.,” *World J. Surg.*, vol. 32, no. 2, pp. 141–148, Feb. 2008.

[2] V. Lahanas, E. Georgiou, and C. Loukas, “Surgical simulation training systems: box trainers, virtual reality and augmented reality simulators,” *Int. J. Adv. Robot. Autom.*, vol. 1, no. 2, pp. 1–9, 2016.

[3] A. Darzi and Y. Munz, “The Impact of Minimally Invasive Surgical Techniques,” *Annu. Rev. Med.*, 2004.

[4] R. Riener and M. Harders, *Virtual Reality in Medicine*. Springer-Verlag London, 2012.

[5] C. E. Buckley, E. Nugent, D. Ryan, and P. C. Neary, “Virtual Reality – A New Era in Surgical Training,” 2020.

[6] S. Misra, K. T. Ramesh, and A. M. Okamura, “Modeling of Tool-Tissue Interactions for Computer-Based Surgical Simulation: A Literature Review,” *Presence (Camb).*, vol. 17, no. 5, p. 463, Oct. 2008.

[7] Maimonides-Centre, “Simulation training at maimonides medical center,” 2015. [Online]. Available: http://www.orthobrooklyn.com/blogs/simulation-training-at-maimonides-medical-center/. [Accessed: 05-May-2015].

[8] J. Wu, R. Westermann, and C. Dick, “A Survey of Physically Based Simulation of Cuts in Deformable Bodies,” *Comput. Graph. Forum*, vol. 34, no. 6, pp. 161–187, 2015.

[9] J. Zhang, Y. Zhong, and C. Gu, “Deformable Models for Surgical Simulation: A Survey,” *IEEE Rev. Biomed. Eng.*, vol. 11, pp. 143–164, 2018.

[10] G. S. Ruthenbeck and K. J. Reynolds, “Virtual reality surgical simulator software development tools,” *J Sim*, vol. 7, no. 2, pp. 101–108, May 2013.

[11] Q. Cheng, P. X. Liu, P. Lai, S. Xu, and Y. Zou, “A Novel Haptic Interactive Approach to Simulation of Surgery Cutting Based on Mesh and Meshless Models,” *J. Healthc. Eng.*, vol. 2018, p. 9204949, Apr. 2018.

[12] K. Cleary and T. M. Peters, “Image-Guided Interventions: Technology Review and Clinical Applications,” *Annu. Rev. Biomed. Eng.*, vol. 12, no. 1, pp. 119–142, Jul. 2010.

[13] M. Duprez *et al.*, “Quantifying discretization errors for soft-tissue simulation in computer assisted surgery: a preliminary study,” *eprint arXiv:1806.06944*. p. arXiv:1806.06944, 01-Jun-2018.

[14] H. Delingette, “Toward realistic soft-tissue modeling in medical simulation,” *Proc. IEEE*, 1998.

[15] D. Bielser and M. H. Gross, “Interactive simulation of surgical cuts,” in *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, 2000.

[16] G. A. Holzapfel, *Nonlinear Continuum Mechanics: a continuum approach for*

engineering. John Wiley & Sons Ltd., 2000.

[17] G. Picinbono, H. Delingette, and N. Ayache, "Non-linear anisotropic elasticity for real-time surgery simulation," *Graph. Models*, vol. 65, no. 5, pp. 305–321, 2003.

[18] F. Martínez-Martínez *et al.*, "A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time," *Comput. Biol. Med.*, vol. 90, no. Supplement C, pp. 116–124, 2017.

[19] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. A. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Med. Image Anal.*, vol. 18, no. 2, pp. 394–410, 2014.

[20] T. Belytschko, W. K. Liu, and B. Moran, *Nonlinear Finite Elements for Cnotinua and Structures*. John Wiley & Sons Ltd., 2000.

[21] J. Wu, R. Westermann, and C. Dick, "Physically-based Simulation of Cuts in Deformable Bodies: A Survey." BT - Eurographics 2014 - State of the Art Reports, Strasbourg, France, 2014, pp. 1–19, 2014.

[22] V. P. Nguyen, T. Rabczuk, S. Bordas, and M. Duflot, "Meshless methods: A review and computer implementation aspects," *Math. Comput. Simul.*, vol. 79, no. 3, pp. 763–813, 2008.

[23] M. Bro-Nielsen, "Finite element modeling in surgery simulation," *Proceedings of the IEEE*, vol. 86, no. 3. pp. 490–503, 1998.

[24] T. Belytschko and T. Black, "Elastic crack growth in finite elements with minimal remeshing," *Int. J. Numer. Meth. Engng*, vol. 45, pp. 601–620, 1999.

[25] C. Monserrat, U. Meier, M. Alcañiz, F. Chinesta, and M. C. Juan, "A new approach for the real-time simulation of tissue deformations in surgery simulation," *Comput. Methods Programs Biomed.*, vol. 64, no. 2, pp. 77–85, 2001.

[26] N. Molino, Z. Bao, and R. Fedkiw, "A Virtual Node Algorithm for Changing Mesh Topology During Simulation," *Proceeding of Eurographics*, pp. 73–80, 2007.

[27] C. D. Bruyns, S. Senger, A. Menon, K. Montgomery, S. Wildermuth, and R. Boyle, "A survey of interactive mesh-cutting techniques and a new method for implementing generalized interactive mesh cutting using virtual tools‡," *J. Vis. Comput. Animat.*, vol. 13, no. 1, pp. 21–42, 2002.

[28] E. Sifakis, K. G. Der, and R. Fedkiw, "Arbitrary cutting of deformable tetrahedralized objects," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 73–80.

[29] H. Delingette, S. Cotin, and N. Ayache, "A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation," *Proc. Comput. Animat. 1999*, 1999.

[30] D. Bielser, V. A. Maiwald, and M. H. Gross, "Interactive cuts through 3-

dimensional soft tissue," *Comput. Graph. Forum*, vol. 18, no. 3, pp. C37–C38, 1999.

[31] D. Steinemann, M. A. Otaduy, and M. Gross, "Fast arbitrary splitting of deforming objects," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2006, pp. 63–72.

[32] A. B. Mor and T. Kanade, "Modifying soft tissue models: Progressive cutting with minimal new element creation," *Med. Image Comput. Comput. Interv. - Miccai 2000*, vol. 1935, pp. 598–607, 2000.

[33] N. Pietroni, F. Ganovelli, P. Cignoni, and R. Scopigno, "Splitting cubes: a fast and robust technique for virtual cutting," *Vis. Comput.*, vol. 25, no. 3, pp. 227–239, 2009.

[34] H.-W. Nienhuys and A. F. van der Stappen, "A surgery simulation supporting cuts and finite element deformation," in *International conference on medical image computing and computer-assisted intervention*, 2001, pp. 145–152.

[35] P.-F. Villard *et al.*, "Toward a realistic simulation of organ dissection," *Stud. Health Technol. Inform.*, 2014.

[36] G. R. Joldes, A. Wittek, and K. Miller, "Non-locking Tetrahedral Finite Element for Surgical Simulation," *Commun. Numer. Methods Eng.*, vol. 25, no. 7, pp. 827–836, Jul. 2009.

[37] Z. A. Taylor, S. Crozier, and S. Ourselin, "A Reduced Order Explicit Dynamic Finite Element Algorithm for Surgical Simulation," *IEEE Transactions on Medical Imaging*, vol. 30, no. 9. pp. 1713–1721, 2011.

[38] S. F. Frisken-Gibson, "Using linked volumes to model object collisions, deformation, cutting, carving, and joining," *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 4, pp. 333–348, 1999.

[39] S. Martin, P. Kaufmann, M. Botsch, M. Wicke, and M. Gross, "Polyhedral Finite Elements Using Harmonic Basis Functions," *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1521–1529, Jul. 2008.

[40] G.-J. Song and N. P. Reddy, "Tissue cutting in virtual environments," *Med. Meets Virtual Real. IV*, p. 359, 1995.

[41] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *Vis. Comput. Graph. IEEE Trans.*, vol. 5, no. 1, pp. 62–73, 1999.

[42] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002, pp. 49–54.

[43] Z. A. Taylor, S. Crozier, and S. Ourselin, "Real-time surgical simulation using reduced order finite element analysis.," *Med. Image Comput. Comput. Assist. Interv.*, vol. 13, no. Pt 2, pp. 388–395, 2010.

[44] B. Ghali and S. Sirouspour, "Nonlinear finite element-based modeling of soft-

tissue cutting," in *TIC-STH'09: 2009 IEEE Toronto International Conference - Science and Technology for Humanity*, 2009, pp. 141–146.

[45] A. Palyanov, S. Khayrulin, and S. D. Larson, "Application of smoothed particle hydrodynamics to modeling mechanisms of biological tissue," *Adv. Eng. Softw.*, vol. 98, pp. 1–11, 2016.

[46] M. K. Rausch, G. E. Karniadakis, and J. D. Humphrey, "Modeling soft tissue damage and failure using a combined particle/continuum approach," *Biomech. Model. Mechanobiol.*, vol. 16, no. 1, pp. 249–261, 2017.

[47] S. De, J. Kim, Y.-J. Lim, and M. A. Srinivasan, "The point collocation-based method of finite spheres (PCMFS) for real time surgery simulation," *Comput. Struct.*, vol. 83, no. 17–18, pp. 1515–1525, Jun. 2005.

[48] K. Miller, A. Horton, G. Joldes, and A. Wittek, "Beyond Finite Elements: A Comprehensive, Patient-Specific Neurosurgical Simulation Utilizing a Meshless Method," *J. Biomech.*, vol. 45, no. 15, p. 10.1016/j.jbiomech.2012.07.031, Oct. 2012.

[49] A. Horton, A. Wittek, G. R. Joldes, and K. Miller, "A meshless Total Lagrangian explicit dynamics algorithm for surgical simulation," *Int. j. numer. method. biomed. eng.*, vol. 26, no. 8, pp. 977–998, 2010.

[50] G. Y. Zhang, A. Wittek, G. R. Joldes, X. Jin, and K. Miller, "A three-dimensional nonlinear meshfree algorithm for simulating mechanical responses of soft tissue," *Eng. Anal. Bound. Elem.*, vol. 42, pp. 60–66, May 2014.

[51] X. Jin, G. R. Joldes, K. Miller, K. H. Yang, and A. Wittek, "Meshless algorithm for soft tissue cutting in surgical simulation," *Comput. Methods Biomech. Biomed. Engin.*, vol. 17, no. 7, pp. 800–811, May 2014.

[52] Y. Zou, P. X. Liu, Q. Cheng, P. Lai, and C. Li, "A New Deformation Model of Biological Tissue for Surgery Simulation," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3494–3503, 2017.

[53] H. A. Chowdhury, A. Wittek, K. Miller, and G. R. Joldes, "An element free Galerkin method based on the modified moving least squares approximation," *J. Sci. Comput.*, vol. 71, no. 3, pp. 1197–1211, 2017.

[54] J. G. Wang and G. R. Liu, "A point interpolation meshless method based on radial basis functions," *Int. J. Numer. Methods Eng.*, vol. 54, no. 11, pp. 1623–1648, Aug. 2002.

[55] R. Aras, Y. Shen, and M. Audette, "An analytic meshless enrichment function for handling discontinuities in interactive surgical simulation," *Adv. Eng. Softw.*, vol. 102, no. Supplement C, pp. 40–48, 2016.

[56] Q. Q. Cheng, P. X. Liu, P. H. Lai, and Y. N. Zou, "An Interactive Meshless Cutting Model for Nonlinear Viscoelastic Soft Tissue in Surgical Simulators," *IEEE Access*, vol. 5, pp. 16359–16371, 2017.

[57] H. Jung and D. Y. Lee, "Real-time cutting simulation of meshless deformable object using dynamic bounding volume hierarchy," *Comput. Animat. Virtual Worlds*, vol. 23, no. 5, pp. 489–501, 2012.

[58] L. Jeřábková and T. Kuhlen, “Stable cutting of deformable objects in virtual environments using XFEM,” *IEEE Comput. Graph. Appl.*, vol. 29, no. 2, pp. 61–71, 2009.

[59] U. Kühnapfel, H. K. Çakmak, and H. MaaB, “Endoscopic surgery training using virtual reality and deformable tissue simulation,” *Comput. Graph.*, vol. 24, pp. 671–682, 2000.

[60] A. Faraci, F. Bello, and A. Darzi, “Soft tissue deformation using a nonlinear hierarchical finite element model with real-time online refinement.,” *Stud. Health Technol. Inform.*, vol. 111, pp. 137–144, 2005.

[61] P. Krysl, S. Lall, and J. E. Marsden, “Dimensional model reduction in non-linear finite element dynamics of solids and structures,” *Int. J. Numer. Methods Eng.*, vol. 51, no. 4, pp. 479–504, 2001.

[62] C. Basdogan, C.-H. Ho, and M. A. Srinivasan, “Virtual environments for medical training: graphical and haptic simulation of laparoscopic common bile duct exploration,” *IEEE/Asme Trans. Mechatronics*, vol. 6, no. 3, pp. 269–285, 2001.

[63] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick, “Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes,” in *Computer Graphics Forum*, 2001, vol. 20, no. 3, pp. 349–358.

[64] H. Courtecuisse, P. Kerfriden, C. Duriez, J. Allard, S. P. A. Bordas, and S. Cotin, “Real-time simulation of surgical cutting in biological tissues using a semi-implicit time integration scheme,” in *International Conference on Computational Mechanics*, 2013.

[65] J. D. Owens *et al.*, “A survey of general-purpose computation on graphics hardware,” *Computer Graphics Forum*. 2007.

[66] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, “GPU Computing,” *Proc. IEEE*, vol. 96, no. 5, pp. 879–899, 2008.

[67] S. F. Johnsen *et al.*, “NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 10, no. 7, pp. 1077–1095, Sep. 2015.

[68] P. Mariappan *et al.*, “GPU-based RFA simulation for minimally invasive cancer treatment of liver tumours,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 1, pp. 59–68, 2017.

[69] Z. A. Taylor, M. Cheng, and S. Ourselin, “High-Speed Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units,” *IEEE Trans. Med. Imaging*, vol. 27, no. 5, pp. 650–663, 2008.

[70] Y. Zhang, J. Zhao, Z. Yuan, Y. Ding, C. Long, and L. Xiong, “CUDA Based GPU Programming to Simulate 3D Tissue Deformation,” in *2010 International Conference on Biomedical Engineering and Computer Science*, 2010, pp. 1–5.

[71] R. Mafi and S. Sirouspour, “GPU-based acceleration of computations in nonlinear finite element deformation analysis,” *Int. j. numer. method. biomed. eng.*, vol. 30, no. 3, pp. 365–381, Mar. 2014.

[72] K. Sase, A. Fukuhara, T. Tsujita, and A. Konno, “GPU-accelerated surgery simulation for opening a brain fissure,” *ROBOMECH J.*, vol. 2, no. 1, p. 17, 2015.

[73] G. R. Joldes, A. Wittek, and K. Miller, “Real-Time Nonlinear Finite Element Computations on GPU - Application to Neurosurgical Simulation,” *Comput. Methods Appl. Mech. Eng.*, vol. 199, no. 49–52, pp. 3305–3314, Dec. 2010.

[74] S. Jia, W. Zhang, X. Yu, and Z. Pan, “CPU-GPU mixed implementation of virtual node method for real-time interactive cuttin deformable objects using OpenCL,” *Int. J. CARS*, 2015.

[75] Z. A. Taylor, M. Cheng, and S. Ourselin, “Real-time nonlinear finite element analysis for surgical simulation using graphics processing units.,” *Med. Image Comput. Comput. Assist. Interv.*, vol. 10, no. Pt 1, pp. 701–708, 2007.

[76] J. Bonet and D. Wood, Richard, *Nonlinear Continuum Mechancis for Finite Element Analysis*. Cambridge University Press, 1997.

[77] T. Belytschko, K. W. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons Ltd., 2000.

[78] C. Daux, N. Moës, J. Dolbow, N. Sukumar, and T. Belytschko, “Arbitrary branched and intersecting cracks with the extended finite element method,” *Int. J. Numer. Methods Eng.*, vol. 48, no. 12, pp. 1741–1760, 2000.

[79] N. Sukumar, N. Moës, B. Moran, and T. Belytschko, “Extended finite element method for three-dimensional crack modelling,” *Int. J. Numer. Methods Eng.*, vol. 48, no. 11, pp. 1549–1570, 2000.

[80] K.-J. Bathe, *Finite Element Procedures*, 2nd ed. Englewood Cliffs, N.J.: Prentice Hall, 1996.

[81] M. Shikta, “Modeling and simulation of cutting in soft biological tissues for surgical simulation,” University of Cincinnati, 2012.

[82] T. Belytschko, “A survey of numerical methods and computer programs for dynamic structural analysis,” *Nucl. Eng. Des.*, vol. 37, no. 1, pp. 23–34, 1976.

[83] R. D. Cook, “Concepts and applications of finite element analysis,” 1981.

[84] K. Miller, G. Joldes, D. Lance, and A. Wittek, “Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation,” *Commun. Numer. Methods Eng.*, vol. 23, no. 2, pp. 121–134, 2007.

[85] L. Vigneron, J. Verly, and S. Warfield, “On Extended Finite Element Method (XFEM) for Modelling of Organ Deformations Associated with Surgical Cuts: Medical Simulation,” *Lecture Notes in Computer Science*, vol. 3078. pp. 134–143, 2004.

[86] L. M. Vigneron, S. K. Warfield, P. A. Robe, and J. G. Verly, “3D XFEM-based modeling of retraction for preoperative image update,” *Comput. Aided Surg.*, vol. 16, no. 3, pp. 121–134, May 2011.

[87] T. Möller, “A Fast Triangle-Triangle Intersection Test,” *J. Graph. Tools*, vol. 2, no. 2, pp. 25–30, Jan. 1997.

[88] M. Duflot, “A study of the representation of cracks with level sets,” *Int. J. Numer. Methods Eng.*, vol. 70, no. 11, pp. 1261–1302, Jun. 2007.

[89] S. Mohammadi, *Extended Finite Element Method: For Fracture Analysis of Structures*. Singapore: Blackwell Publishing Ltd, 2008.