Biomechanical Simulation for Remote Surgery

Using Meshless Method to Predict Brain Phantom Deformation Upon Needle Insertion

Agnes Kang
Masters of Professional Engineering
Mechanical Engineering
The University of Western Australia

30th May 2016

Supervisor: Professor Adam Wittek

Content word count : 6600
Abstract

Historically, the calculations of soft tissue deformation have been based on Finite Element Method (FEM), and this method has been verified in numerous applications in computer-aided engineering and biomechanics. However, the FEM presented two major limitations of requiring time-consuming patient-specific mesh generation; and the deterioration of the solution accuracy when large distortions occur. Using a numerical method that does not require strict spatial discretisation is one solution to overcome these limitations of the FEM.

In this study, a verified Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm is adapted and utilised to simulate point indentation on a hyperelastic Neo-Hookean cylindrical solid. This indentation is to simulate the needle insertion into a soft tissue prior to puncture and to predict the soft tissue behaviour, which can be modelled as a hyperelastic, almost incompressible solid. Verification of the MTLED simulation has been attempted by using a finite elements solver, Abaqus. Furthermore, CUDA accelerated computation has been explored to facilitate parallel computing through GPU, and resulted in an impressive decrease in MATLAB computation time. To understand the needle-soft tissue insertion response, experiments were conducted on previously constructed brain phantom samples.
Acknowledgement

I would like to acknowledge my supervisor, Professor Adam Wittek, for the continuous support he provided throughout this research project with his wisdom and depth of knowledge.

I would like to acknowledge the work of Dr Ashley Horton (former PhD student at the UWA Intelligent Systems for Medicine Laboratory) in lighting the path to complete this project.

I would like to acknowledge the grant ARC DP160170014 ‘Biomechanics Meets Robotics: Methods for Accurate and Fast Needle Targeting’, in which the project topic originated from.

I would like to acknowledge the funding support from ARC DP1092893 and ARC DP120100402 grants, which enabled the creation of the Meshless Total Lagrangian Explicit Dynamics code I had used in the project.
Contents

1. Introduction .................................................................................................................. 4
   1.1 Biomechanical simulation .................................................................................. 4
   1.2 Limitations of Finite Element Methods .......................................................... 4
   1.3 Meshless Method: A Possible Solution to Limitations of Finite Element Method .................................................................................................................. 6
   1.4 Meshless Total Lagrangian Explicit Dynamics Method ....................................... 6
   1.5 CUDA Accelerated Simulation ........................................................................... 9
   1.6 Brain Phantom ...................................................................................................... 9

2. Experiment: Needle-Phantom Insertion ...................................................................... 11
   2.1 Motivation ............................................................................................................ 11
   2.2 Experimental Design ......................................................................................... 11
   2.3 Tissue phantom Layers ...................................................................................... 12
   2.4 Strain Rate Dependency ...................................................................................... 13
   2.5 Effect of Needle Surface Area ............................................................................ 14
   2.6 Surface Deformation On Puncture ..................................................................... 15

3. Model Implementation Using Meshless Method ......................................................... 17
   3.1 Motivation ............................................................................................................ 17
   3.2 Geometry of the Analysed Continuum ................................................................ 17
   3.3 Boundary Conditions and Loading ...................................................................... 19
   3.4 Material Properties ............................................................................................ 19
   3.5 Explicit Integration ............................................................................................. 21

4. Result .......................................................................................................................... 22
   4.1 Meshless Method Simulation .............................................................................. 22
   4.2 Integration Point Variation .................................................................................. 24
   4.3 Comparison With Other Node Arrangement ....................................................... 26

5. Discussion .................................................................................................................... 29
   5.1 Verification Using Finite Element Method ........................................................ 29
   5.2 CUDA Accelerated Computation ...................................................................... 31

6. Conclusions and Future Work .................................................................................... 33
   6.1 Brain Phantom ..................................................................................................... 33
   6.2 Using Meshless Method to Predict Deformation Upon Needle Indentation ....... 33
   6.3 CUDA Accelerated Framework .......................................................................... 34

Appendix A ....................................................................................................................... 35
1. Introduction

1.1 Biomechanical simulation

In contemporary clinical and technical domain, the advantages of surgical robots and manipulators are well recognised and their uses sought after due to their precision, accuracy and the potential for remote surgery (Miller et al. 2012; Seifabadi et al. 2012). The surgical robots are operated by motion trajectory planning algorithms, which update the trajectory to reach surgical target. The initial surgical trajectory can be established by using pre-operative images attained through magnetic resonance imaging (MRI), X-ray computed tomography (CT scan) or ultrasound – although, ultrasound is not ideal for neurosurgery imaging as the brain tissue and ultrasound device contact is required for medical imaging, however possible for internal soft tissue organs. However, tracking the deformation using medical imaging is cumbersome due to the surgical robot having to operate within these medical imaging devices. The simplest approach is to “warp” the pre-operative images using linear transformation; however this is a major simplification and does not ensure plausibility of the predicted deformation especially for the soft tissue exhibiting non-linear behaviour. The preferred approach is to calculate the deformation field within the organ using a biomechanical model and then update the images appropriately (Clatz et al., 2003; Wittek et al., 2005; Sun et al., 2005; Miller and Wittek, 2006; Horton, 2015). The simulation can be used to predict the reaction forces acting on the surgical tools, deformation of the tissue and hence the change in the target location. This information is to be fed back into the trajectory planning algorithm to adapt its path accordingly. Biomechanical simulation of soft tissue deformation - specifically brain in this study – can also be used for surgical planning and training of neurosurgeons in modern clinics, while the ultimate goal entails control systems of neurosurgical robots with tissue deformation prediction module, maximising the use of its potential when real-time application is achieved.

1.2 Limitations of Finite Element Methods

Historically, the calculations of soft tissue deformation have been based on finite element method, (Cotin et al., 1999; Bathe, 1996; Szekely et al., 2000; Picinbono et al., 2003; Luboz et al., 2005; Wittek et al., 2007; Miller et al., 2007; Wittek et al., 2009) and this method has been verified in numerous applications in computer-aided engineering and biomechanics (Miller et al.2012). The finite element method utilises computational grids that form a mesh of interconnected hexahedral and tetrahedral elements. The biomechanical models utilise the
finite element method (Bathe, 1996) to solve sets of partial differential equations accompanied by constitutive parameters, boundary and initial conditions governing the deformation behaviour of the analysed continuum (Miller et al. 2012; Wittek. 2004). Since human body organs during surgical procedures undergo large deformations, fully non-linear geometric formulations and material models have to be used (Wittek. 2004). In Wittek et al. (2010) it is demonstrated that a high level of precision can be achieved in patient-specific simulations of surgical procedures using these non-linear –both geometric and material-biomechanical models.

While the use of finite element methods in predicting soft tissue deformation has been promising; showing that near real-time simulations of surgical procedures, using fully nonlinear biomechanical models can be achieved with a high level of precision (Joldes et al, 2008; Joldes et al, 2009; Wittek et al., 2005; Miller et al., 2007; Wittek et al.; 2007), two major limitations of the method have been identified over the years:

1. Time consuming generation of patient specific finite element meshes of the brain and other body organs.
2. Deterioration of the solution accuracy and instability as the elements undergo distortion when surgical tools induce large local deformations.

The accuracy of the finite element calculations depends heavily on the mesh discretising the geometry, hence the use of quality hexahedral elements is desired (Yang and King, 2011) to obtain accurate result. Creation of patient-specific meshes typically requires involvement of an experienced analyst due to the highly irregular geometry of the brain and other internal organs. Creation of patient-specific finite element models is time consuming, expensive and incompatible with clinical work-flow (Wittek et al., 2015).

Even when a good quality mesh is created, the accuracy and reliability of the solution generated through finite element methods declines when the mesh undergoes distortion induced by large deformations of the analysed continuum. In Wittek et al. (2015) it is expressed that simulations of surgical tool and brain cutting interaction tends to result in small and poorly shaped elements.
1.3 Meshless Method: A Possible Solution to Limitations of Finite Element Method

One solution to overcome limitations of the finite element method is to use a numerical method that does not require strict spatial discretisation. This study utilises a meshless method, which is part of the Element Free Galerkin family that uses a cloud of nodes to discretise the geometry instead of the elements (Belytschko et al., 1994). The arrangement of the nodes is arbitrary (although there are some conditions that meshless ‘grids’ need to satisfy- Joldes et al., 2015) and hence the placement of the nodes on the problem domain can be automated (Horton, 2015).

Miller et al. (2012) presents the Meshless Total Lagrangian Explicit Dynamics Method (MTLED) that can be used to compute brain deformation during surgery, which reduces the complexity of model generation and is as accurate as the finite element method. In this method the nodes are distributed automatically through the domain which simplifies the problem of creating a patient-specific computational grid to a trivial exercise (Miller et al. 2012). Integration is performed over a simple, regular background grid which does not need to conform to the geometry boundaries. In the fully geometrically nonlinear Total Lagrangian formulation, the stresses and strains are measured with respect to the original configuration which allows for preoperative computation of most spatial derivatives, and thus reduces the intraoperative computational load.

A further advantage of using a meshless method is the ability to deal with extremely large deformations and boundary changes (Melenk and Babuska, 1996; Babuska and Melenk, 1997; Liu, 2003; Li and Kam, 2004) that occur during neurosurgical procedures such as tissue retractions, cuts and removal.

Miller et al. (2012) endorses the meshless algorithms that can deal with irregular 3D geometries, large deformations, non-linear, almost incompressible materials and contacts. It allows automatic discretisation of the problem domain and is sufficiently fast to produce clinically useful results in a short time using only easily-available consumer hardware.

1.4 Meshless Total Lagrangian Explicit Dynamics Method

The Meshless Total Lagrangian Explicit Dynamics Method (MTLED) can accept an almost arbitrary placement of nodes throughout the simulation geometry, and can automatically create the node placement and the integration grid (Horton, 2015). The method
precomputes the constant strain-displacement matrices for each integration cell and uses the deformation gradient to calculate the full matrix at each timestep (Miller et al., 2007), and uses explicit time integration based on the central difference method and hence unlike implicit time integration, the method does not require solving systems of equations at every timestep.

There are three main frameworks available to discretise the problem geometry when using the meshless method (Horton et al., 2010; Zhang et al. 2013; Jin et al. 2014; Horton, 2015):

1. Background finite element mesh, where a mesh is created conforming to the nodes and standard finite element integration methods are used.
2. Nodal integration, where the nodes are used as single sampling points and the weights are set as the volume associated with each node. Given the integration points, the problem domain is discretised via Voronoi decomposition.
3. Background grid, where a regular grid of cells is imposed over the geometry and integration is performed in each cell.

The background integration grid is mostly used with the MTLED method to take advantage of the flexibility of the meshless method and to overcome the instability imposed by the nodal integration framework.

In Horton (2015) the algorithm overview for this method is presented as shown below, in notation based on Bathe (1996).

**Preprocessing**

1. Load simulation geometry $\Omega$ in the form of two lists:
   - Node locations.
   - Integration point locations.
2. Load boundary conditions.
3. Loop through list of integration point locations. For each integration point:
   - Identify $n$ local nodes associated with the integration point.
   - Create and store the $3\times n$ matrix $D\Phi(x)$ of moving least squares shape function derivatives

\[
D\Phi_{k,i}(x) = \frac{\partial \phi_i(x)}{\partial x_k} \quad k = 1,2,3 \quad i = 1,2, ..., n
\]
4. Loop through nodes and associate to each a suitable mass.
5. Initialise global nodal displacements $^{-\Delta t}U$ and $^{0}U$.

**Solving**

In every timestep $t$:

1. Loop through integration points
   - From precomputed list, find $n$ local nodes and associated shape function derivatives $D\Phi(x)$ for the given integration point $x$.
   - Find $n \times 3$ local nodal deformation matrix $^{t}u$.
   - Calculate deformation gradient $^{t}X$.
   - Calculate strain-displacement matrix $^{t}B_L$.
   - Calculate second Piola-Kirchoff stress vector $^{t}S$ (using material properties).
   - Calculate and store local nodal reaction forces
     \[
     ^{t}f = \int_{V^{0}}^{t}B_L^{T}^{t}S dV^{0}
     \]
2. Combine all local nodal reaction forces to create global nodal reaction forces vector $^{t}F$.
3. Calculate global nodal displacements at time $t + \Delta t$ using central difference method
   \[
   ^{t+\Delta t}U = -\Delta t^{2}M^{-1} (^{t}F - ^{t}R) + 2^{t}U - ^{t-\Delta t}U
   \]
   where $M$ is the diagonal mass matrix and $^{t}R$ is the load applied at time $t$.

Support domain and Moving Least Squares theory initially developed by Lancaster and Salkauskas (1981) and used in meshless methods in EFG (Nayroles et al., 1992; Belytschko et al. 1994) involves the relationship between integration points and nodes.

Consider an arrangement of nodes in the domain $\Omega$, where at each of these nodes a field variable, nodal displacement, is attached in the case of mechanical deformation. To find the displacement of an integration point, the field variables at nearby nodes are considered to create approximate interpolation. The support domain defines the local nodes for an integration point, where the ‘support domain’ of a point $x \in \Omega$ is some bounded region $S \subset \Omega$ that contains both $x$ and at least one node. The generation of the support domains is critical as this binds the disjointed and distant nodes by overlapping support domains. The
interpolation is performed by moving least squares shape functions. The support domain and moving least squares shape functions are used in this method for its simplicity and robustness (Horton, 2015).

1.5 CUDA Accelerated Simulation

Computing tissue deformations is the most computationally expensive part of surgery simulation (Shahingohar, 2010). To be compatible with a clinical work-flow, such computations must be performed within the real-time constraints of surgery on hardware that can be easily deployed in an operating theatre. Graphics Processing Units (GPUs), an off-the-shelf and cost-effective (around $5,000 for state of the art GPU) consumer hardware, have been proposed and used to facilitate such computations (Taylor, Cheng & Ourselin, 2008; Joldes et al., 2009). The first Graphics Processing Unit (GPU) implementation of a non-linear finite element solver was presented by Taylor, Cheng & Ourselin (2008), and their study demonstrated that the Total Lagrangian Explicit Dynamics Method (TLED) algorithms can be executed in fast computational speed on GPU due to the parallel computations. Utilisation of GPU in numerical modelling in both finite element method and meshless method have been conducted and resulted favourably toward achieving real-time applications (Joldes et al., 2010; Shahingohar 2010 Miller et al. 2012; Wittek et al., 2015;) with computation time ranging in the seconds. NVIDIA’s CUDA is an affordable technology enabling users to take advantage of GPU’s parallel computing capability for general purpose computing. Thus, the results can be obtained and provided to surgeons in ‘real-time’ without the need for supercomputers.

1.6 Brain Phantom

A previous Final Year Project student Scott List has constructed a mechanical brain phantom for his research along with determining the constitutive properties of the phantom and creating a validated brain phantom finite element model. Construction of this brain phantom overcame the limitations of using animal and human tissue for evaluation, where degradation of biological tissue inhibits the repeatability of experiments and results. The use of the phantom resolved potential logistical, ethical and safety hurdles presented when using bio-hazardous material (List, 2015). The brain phantom was made of Sylgard 527 silicone gel (List, 2015), which is known to closely reflect the behaviour of brain tissue (Brands et al., 1999; Brands et al., 2000).
The brain phantom is constructed layer by layer; from A to F represent different parts of the brain having varying degree of stiffness, figure 1.6.1 illustrates the brain phantom surface model created from MRI images slices, and the finite element model constructed using hexahedral elements.

![Figure 1.6.1 a) Brain phantom surface model created on 3D Slicer from MRI slices. b) Finite element model of the brain phantom in HyperMesh with colour-coded element sets defined for each phantom layer](image)

The phantom image was acquired at Australian National Imaging Facility at the Centre for Microscopy, Characterisation and Analysis, The University of Western Australia and is retrieved from the Final Year Project by Scott List (List, 2015).

The scope of this study is to utilise a previously verified meshless method, MTLED, to predict brain phantom indentation upon surgical needle insertion before puncture, using CUDA to accelerate its calculation. The study also includes conducting needle insertion experiments on the brain phantom samples layers A to F representing the different areas of the brain with different stiffness. The purpose of the experiment is to understand the behaviour of the phantom upon needle insertion by observing the force exerted onto the needle before, during, and post puncture.
2. Experiment: Needle-Phantom Insertion

2.1 Motivation

Deformation and force needed to puncture a particular tissue depend on the tissue properties, needle insertion speed, needle size and tip shape (Abolhassani et al., 2007; Cowan et al., 2011; Mahvash and Dupont, 2009; Wan et al., 2005; Wedlick and Okamura, 2012). By observing the force exerted onto the needle, the needle-phantom insertion experiments aimed to analyse four characteristics of the brain phantom: the stiffness of the phantom layers A to F; the strain rate dependency; the effect of the needle surface area; and the surface deformation prior to puncture.

2.2 Experimental Design

The needle-phantom sample insertion experiment was conducted using the Portable Experimental Device for Determining the Mechanical Properties of Brain-Skull Interface constructed by Mr Sudip Agrawal, PhD student at Intelligent Systems for Medicine Laboratory at The University of Western Australia. Agrawal (2014) explains that the Device was designed to accurately measure mechanical properties of brain and adjoining tissues to develop corresponding and realistic mathematical model of brain that can be used in computer simulation and other related fields like virtual surgeon training system and computer-integrated and robot-aided surgery.

A conical tip needle of diameter 0.8 mm is fixed onto the load cell perpendicularly using adhesives, and a no-slip boundary between the gel sample and the plate was achieved by attaching sandpaper to the bottom plate- as the Sylgard gel sticks firmly to most materials including sandpaper- following a verified method used in shear compression experiments by Bilston, Liu and Phan-Thien (2001) and Ma et al., (2010). The Experimental Device is programmed to translate the loading head in the negative z-direction (see Figure 2.1.1 for axis definition), inserting the needle into the phantom sample at a constant speed and desired displacement as set by the Device operator. The samples are of height 24 mm and diameter 38 mm. Figure 2.1.1 illustrates the experimental set up.

Device automatically records the force exerted onto the load cell via needle, then a spreadsheet of results is generated for analysis.
2.3 Tissue phantom Layers

The conical tip needle of diameter 0.8 mm was inserted 15mm into each phantom samples at the speed of 1mm/s. The stiffness of material is expressed by,

\[ k = \frac{F}{\delta} \]

Where  \( k = \text{stiffness of the body of material} \)
\( F = \text{force applied on the body} \)
\( \delta = \text{displacement produced by the force along } z - \text{axis} \)

It can be observed from the graph shown in Figure 2.2.1; the phantom sample layer A has the lowest stiffness while layer C has the highest stiffness.

It should be noted that the force experienced by the needle during insertion is due to the cutting force and the adhesive force of the Sylgard gel sticking on to the needle surface. Frictional force is not present.
2.4 Strain Rate Dependency

To test the strain rate dependency of the brain phantom undergoing deformation due to needle insertion, the conical point needle was inserted into the phantom samples A to F at two varying speeds of 1mm/s and 0.5 mm/s over the total displacement of 15mm. It is observed that both force vs displacement curve generated by the two varying speeds are alike. Minor discrepancies between the two curves shown in Figure 2.3.1 can be observed, however the difference is not significant. This trend of exerted force being independent of the rate of needle insertion was consistent throughout the other samples tested. It can be deduced that the brain phantom constructed from Sylgard 527 gel does not exhibit strain-rate dependency, which is consistent to findings from the literature (Ma, 2006) and previous uniaxial compression test conducted (List. 2015)
2.5 Effect of Needle Surface Area

The needle-phantom sample insertion experiment was repeated with a different needle of diameter 1.1 mm and with a flat top surface. In figure 2.4.1 the two different types of needles used in the experiment are shown. The flat top needle experienced greater force of adhesion as it was inserted 15mm through the phantom sample compared to the conical point needle.

Figure 2.4.2. Force-Displacement response of Needle-Phantom Layer A Insertion using two different types of needles. The needle with a greater surface area experienced greater exerted force.
The result from this experiment was consistent with numerous literatures on force modeling for needle insertion into soft tissue (Okamura, Simone & O’Leary, 2004; Wan et al., 2005; Mahvash and Dupon, 2009; Wedlick and Okamura, 2012).

### 2.6 Surface Deformation On Puncture

Elgezua et al. (2013) explained that the process of needle-soft tissue cutting follows a distinctive pattern that can be divided into two phases. First the needle pushes the tissue increasing the insertion force steadily, and during this phase the tissue deforms until the stress limit is reached. Second phase is when the needle punctures the tissue and advances into it followed by tissue relaxation. A puncture is observed as a sharp drop in insertion force. This process repeats itself as long as the needle is inserted with another series of pushing and puncturing. The pattern of increase and decrease in the insertion force is illustrated in Figure 2.6.1. The brain phantom response of the force build-up, surface deflection prior to puncture, and the sudden decline of the insertion force upon puncture have been consistent with numerous literature findings (Okamura, Simone & O’Leary, 2004; Mahvash and Dupon, 2009; Casanova et al., 2014).

However the phantom behaviour immediately following the puncture applied only to the results obtained from our experiment, where homogenous Sylgard 527 gel was used in place of soft tissue under needle insertion. As can be seen from the figure 2.6.2, the force curve depresses below zero, demonstrating the phantom is inducing tensile force greater than the insertion force onto the needle just immediately after the puncture. This trend was not observed in previous ex-vivo experiments involving various soft tissues ranging from bovine liver (Elgezua et al. 2013) to rat brain (Casanova et al. 2014). It is speculated that this behaviour of the brain phantom upon needle puncture may be due to the gel adhesion forming around the needle and the lack of membrane in phantom sample resulting in its limitation on simulating complex internal responses of soft tissue.
Figure 2.6.1 Typical axial force measured during needle insertion into the caudate putamen (CPu) (Fm: surface puncture force, Fstop: force at the stop of insertion). This experiment was conducted in vivo at a 0.2 mm/s insertion speed. Copied from Elgezua et al. (2013)

Figure 2.6.2 Axial force measured during needle insertion into the phantom sample layer E
3. Model Implementation Using Meshless Method

3.1 Motivation

In this study Meshless Total Langrangian Explicit Dynamics Method (MTLED) was adopted and used to simulate needle-phantom indentation prior to puncture and to predict the resulting deformation of the phantom sample. I used a verified MTLED algorithm (Horton et al., 2010) coded in MATLAB by Guiyong Zhang (2013). The relevant sections within the code were modified to suit the objective of this study. Zhang’s codes were used along with Joldes et al. (2012)’s implementation which incorporates subroutines in NVIDIA’s CUDA to facilitate parallel computations on Graphics Processing Unit (GPU) when conducting explicit integration to reduce the computation time.

3.2 Geometry of the Analysed Continuum

The geometry used for the meshless method simulation of a cylindrical brain tissue phantom indentation from the needle was a cylinder of height 0.1m and radius 0.05m. This cylinder model was chosen to simulate the section within the phantom sample that undergoes deformation due to needle indentation. Since the needle diameter is significantly small compared to the diameter of the phantom sample, the simulation has been simplified to consider the nodes undergoing prescribed deformation and the affected surrounding nodes. This truncation of the phantom sample geometry into a cylinder of affected nodes is to concentrate on the large local stresses that are generated around the deformed nodes, and to disregard the unaffected sections of the phantom sample. Moreover, a cylinder is simple enough to obtain a good finite element mesh for comparison and when compared to using models of other shapes such as a cube, unrealistic perfect corners are not a major issue.

The cylinder volume was filled with a total of 4538 nodes, where the nodes were imported from an Abaqus (ABAQUS, 2011) input file. Figure 3.2.1 shows the arrangement of the node placement and the integration points from an orthogonal view.

![Figure 3.2.1 Orthogonal view of 4538-noded cylinder](image-url)
Tetrahedral faces were drawn on between the nodes to visualise the shape of the cylinder and the contour when deformed as seen in Figure 3.2.2, otherwise the cylinder with only the nodes visible looked like a clutter and gave little indication of the deformed shape. The node placement in the cylinder volume can be seen in Figure 3.2.3.

In this study, a tetrahedral background integration grid is used with 4 integration points per tetrahedral cell which provides exact integration for polynomial up to second order (Zhang et al., 2013). The increased integration point to node ratio increases the statistical chance that a small group of nodes will be split into multiple support domains by placing an integration point closer to the centre of any large gaps between the nodes. Adding integration points slows down the computation, however enables almost arbitrary placement of nodes in an irregular patient specific geometry (Horton, 2015). Other integration points considered in this study are 1, 4 and 5 per tetrahedral cell.
3.3 Boundary Conditions and Loading

The boundary conditions are chosen to simulate the experiment conducted in section 2.1. The simulation is programmed to perform constrained compression in the z-direction, where all degrees of freedom (x y z) on the bottom surface of the model are constrained and displacement on the z-direction is predefined. This satisfies the no-slip boundary condition imposed in the needle-phantom sample insertion experiment with the use of sandpaper.

The loading is applied by prescribing displacement on the selected nodes on the top surface. The nodes are selected if within the 0.007m radius from the origin to simulate the indentation from a needle with a small surface area. Figure 3.3.1 illustrates the nodes selection range from the top surface of the cylinder model.

The maximum nodal displacement of 0.03m is chosen to reflect the maximum strain of 30% that can be imposed on a finite element model (Miller 2005; Morriss 2008), which is utilised to verify the meshless method simulation in section 5.1.

3.4 Material Properties

Neo-Hookean material model is used to model the brain phantom sample, which is a nearly incompressible, hyperelastic solid with the Poisson’s ratio of 0.49 (Incompressible solids have Poisson ratio of 0.5). The Neo-Hookean model can be used for predicting the nonlinear stress-strain behaviour of materials undergoing large deformations, and with its two parameters it is the simplest to implement in this study, where the focus is on using the algorithm rather than evaluating the biomechanics.
The Neo-Hookean material has the strain-energy density functional

\[ W = \frac{\mu}{2} (\mathcal{I}_1 - 3) + \lambda (J - 1)^2 \]

Where,

\( \mu \) is the Shear Modulus expressed by \( \mu = G = \frac{E}{2(1+\nu)} \)

\( \lambda \) is the first Lamé parameter expressed by \( \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \)

\( J = \det( \mathbf{C} ) \)

\( \mathcal{I}_1 \) is the first strain invariant of the right Cauchy Green deformation tensor.

The second Piola-Kirchoff stress tensor is found from the strain-energy density functional,

\[ \mathbf{\mathit{t}}_0 \mathbf{S} = \lambda J (J - 1) \mathbf{C}^{-1} + \mu \frac{2}{3} \mathbf{I} \]

And hence form the required vector \( \mathbf{\mathit{t}}_0 \mathbf{S} \).

List (2015) has conducted compression experiments to attain constitutive properties of the phantom samples constructed, however there were too many unknown and unclear variables and hence previous verified and validated parameters were used to minimise error. These constitutive material parameters used are summarised in the table 3.4.1, where the shear modulus of 1007 kPa is recognized as appropriate to model the brain’s mechanical behaviour (Miller 2002; Millet et al., 2012).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shear modulus ( \mu ) (Pa)</td>
<td>1.007 x 10³</td>
</tr>
<tr>
<td>Lame’s first parameter ( \lambda ) (Pa)</td>
<td>49.33 x 10³</td>
</tr>
<tr>
<td>Poisson’s ratio</td>
<td>0.49</td>
</tr>
<tr>
<td>Density (kg/m³)</td>
<td>1000</td>
</tr>
<tr>
<td>Young’s Modulus ( \mathbf{E} ) (Pa)</td>
<td>3000</td>
</tr>
</tbody>
</table>

Table 3.4.1 Constitutive Properties on modelling brain phantom sample
3.5 Explicit Integration

For each integration point, the $n \times 3$ first partial, spatial derivatives of the shape functions $D\Phi_{k,i}(x) = \frac{\partial \phi_i(x)}{\partial x_k}$ are computed for $k = 1, 2, 3$ and $n =$ number of nodes in the support domain.

In this study where the problem domain has 4538 nodes and 23069 elements (tetrahedron cells), 92276 integration points were used. The influence radius is computed for each node to return the minimum and maximum support nodes being 11 and 39 respectively.

In the Total Lagrangian formulation the forces on nodes local to a given integration point are calculated by

$$\dot{f} = \int_{V_0} B^T L_s \dot{\Phi} \, dV_0$$

The $3n$ nodal forces calculated at each integration point are combined to form the global force vector $\dot{F}$, and these forces are the only data that is stored at each step of the integration point loop (Horton, 2015). Since the mass is constant, the central difference method is used to find the nodal displacements at time $t + \Delta t$,

$$^{t+\Delta t}U = -\Delta t^2 M^{-1}(\dot{t}F - \dot{t}R) + 2\dot{t}U - ^{t-\Delta t}U$$

Dynamic Relaxation feature with the inclusion of mass-proportional damping is added to this explicit integration to control the oscillations and to reach the steady state solution.
4. Result

4.1 Meshless Method Simulation

Using the MTLED Method as outlined in Section 3, simulations of the almost incompressible Neo-Hookean solid cylinder model with prescribed nodal displacement of 0.03m (30% strain) in the –z direction is performed.

MATLAB code has been written to generate a 3-D displacement plot (Figure 4.1.1) illustrating the deformation with coloured contour, as well as the 3-D force plot (Figure 4.1.1), showing the location of the range of the force exerted by each node. I have written this code to be an addition to the set of MATLAB codes written by Zhang (2013) and Joldes (2012) on MTLED Method computation; the code can be found in Appendix A.

![Displacement (z-axis)](image.png)

Figure 4.1.1. 3-D Displacement plot with coloured contour of deformation of the phantom model cylinder with 4658 nodes. Tetrahedral faces are drawn onto the surface to visualise the contour
For the cylinder of 4538 nodes arranged in a tetrahedral grid, total of 9 nodes on the top surface within the radius of 0.007m were selected and prescribed displacement.

Figure 4.1.2. 3-D Force plot with coloured contour of the range of exerted force on the nodes.

Figure 4.1.1 Force-Displacement Response for 4538 noded cylinder with 4 integration points per tetrahedral cell. The Force denotes the total reaction force exerted by the 9 displaced nodes in the z-direction.
From the simulation, the force – displacement response of the nodes are calculated. Figure 4.1.3. shows the total force exerted by the displaced nodes, and Figure 4.1.4. shows the maximum force exerted by a single node.

4.2 Integration Point Variation

The integration points per tetrahedral cell have been varied in the simulation to observe the resulting force – displacement response of the cylindrical model with 4538 nodes. In Figure 4.2.1 it can be seen that the resulting maximum reaction forces attained from using 4 and 5 integration points per cell were almost identical, and also resulted in very similar single nodal forces as shown in the figure 4.2.2. A large inconsistency was observed in the force – displacement curve obtained by using 1 integration point per cell, in both total reaction force in z- direction (Figure 4.2.1) and the maximum force on a single node (Figure 4.2.2.). In this study, it may be speculated that using only one integration point per cell on meshless method calculations results in inaccurate results.
Figure 4.2.1. Total Force-Displacement Response (z-direction) of Cylinder on 30% indentation with varying number of integration points used.

Figure 4.2.2. Single nodal Force-Displacement Response (z-direction) of Cylinder on 30% indentation with varying number of integration points used.
4.3 Comparison With Other Node Arrangement

Another cylinder model comprising of different number of nodes and the different node placement was also simulated using the MTLED Method.

A 4663-noded cylinder created on Abaqus, a finite element solver, originally in a hexahedral butterfly mesh was imported onto MATLAB to subject it to meshless method simulation of 30% strain indentation. The node placement resulted from the butterfly meshing caused more concentrated volume of nodes in the inner volume of the cylinder and less concentrated on the outer volume. The orthogonal view of the butterfly mesh node placement is shown in Figure 4.3.1, where tetrahedral grid is drawn on the surface to visualise. The circled area indicates the nodes that are selected for indentation, which are within radius of 0.0045m from the cylinder. The symmetrical node placement enabled symmetrical indentation area compared to the 4538-noded cylinder where the surface undergoing the indentation was irregular due to the semi-arbitrary node placement. While the total number of displaced nodes were 9 in both 4538-noded and 4663-noded cylinders, the force exerted by the indentation for the two cylinders were different due to the different node placement and hence the radius of imposed deformation.

Figure 4.3.1. Orthogonal view of the node placement in 4663-noded cylinder in the butterfly mesh formation. Tetrahedral faces were drawn on to visualise the difference in the inner and outer node density. The circled area indicates the radius 0.0045m at which the nodes were selected for indentation.
The 3-D displacement (Figure 4.3.2) and 3-D force plots (Figure 4.3.3) are generated for the 4663-noded cylinder undergoing 0.03m indentation.

Figure 4.3.2 3-D Displacement plot of 4663-noded cylinder on 0.03m indentation radius of 0.0045m. Four integration points were used per tetrahedral cell.

Figure 4.3.3 3-D Force plot of 4663-noded cylinder on 0.03m indentation radius of 0.0045m. A range of single nodal forces are displayed. Four integration points were used per tetrahedral cell.
The effect of varying the number of integration points per tetrahedral cell was also investigated for the 4663-noded cylinder. Figure 4.3.4 summarises the maximum single nodal force exerted in the z-direction during the 30% strain indentation. Similar to the result obtained from the 4538-noded cylinder in section 4.1, this cylinder with butterfly node placement also experiences inaccuracy and particularly instability when only one integration point is used per tetrahedron, as apparent from the graph.

![Graph showing the effect of varying number of integration points on maximum single nodal force exerted in the z-direction.](image)

Figure 4.3.4 Single nodal Force-Displacement Response (z-direction) of Cylinder on 30% indentation with varying number of integration points (1,4,5) used. Notice the unstable response produced when only one integration point per tetrahedral cell is used.
5. Discussion

5.1 Verification Using Finite Element Method

Verification of the MTLED method simulating 30% strain indentation was performed using the Finite Element Methods (FEM) to simulate the same problem on Abaqus.

Static analysis was conducted on Abaqus for the two Neo-Hookean hyperelastic cylinder solids with the same geometry and constitutive properties, both cylinder models comprising of linear tetrahedral elements. The exact displaced nodes from the meshless method simulation were identified to so that the same nodes on the FEM model are also displaced. Data from FEM simulation was compared to the result from that of MTLED using 4 point integration per cell.

For the 4538-noded tetrahedral meshed cylinder the maximum difference between the total reaction forces in the z-direction from the two methods was approximately 23%. Figure 5.1.1 compares the results obtained from using the MTLED Method and FEM to simulate point indentation of 30% strain.

![Force-Displacement Response for 30% Indentation of Cylinder with 4538 nodes](image)

Figure 5.1.1 Comparison graph showing the Force-Displacement Response for the simulation results using the MTLED Method (Meshless) and the Finite Element Method

29
For the 4663-noded butterfly meshed cylinder the maximum difference between the total reaction force in the z-direction from the two methods was approximately 25%. Figure 5.1.2 compares the results obtained from using the MTLED Method and the FEM to simulate point indentation of 30% strain.

It is known that the butterfly mesh is widely and successfully used in FEM modelling of cylindrical volumes (Permeswaran, 2014). When meshed with the butterfly technique, the corner elements are well formed with less warping and collapse of the element shape, reducing the number of highly acute and obtuse angles in the corner elements. Hexahedral elements are most commonly used with the butterfly mesh technique (Permeswaran, 2014), while in our study tetrahedral elements were used.

![Force-Displacement Response for 30% Indentation of Cylinder with 4663 nodes](image)

Figure 5.1.2 Comparison graph showing the Force-Displacement Response for the simulation results using the MTLED Method (Meshless) and the Finite Element Method.

In both verification attempts with the two cylinders, the margin of difference between MTLED Method and FEM was approximately 25%, contrary to the literatures stating the attainment of very close results from the use of the MTLED Method and the FEM (Millet et al., 2010; Miller et al., 2012; Zhang et al., 2013; Horton 2015).
Sources of the discrepancy between results from the MTLED Method and the FEM may have resulted from the following:

- Large (30%) strain imposed by the point indentation resulting in unstable and inaccurate deformation prediction in FEM (Morriss, 2008).
- Use of tetrahedral elements when the hexahedron finite elements are known to be the most effective ones in non-linear finite element procedures using explicit time integration (Wittek et al., 2004; Miller et al., 2010)
- Modifications made in the MTLED MATLAB codes not representing the desired simulation conditions

5.2 CUDA Accelerated Computation

The MTLED Method simulations on MATLAB were performed on a PC with Intel® Core i7 Quad 3.2 GHz Central Processing Unit (CPU) and Graphics Processing Unit (GPU) of GeForce GTX 560 Ti with 384 CUDA cores running on Windows 7.

The range of total elapsed computation time for a complete simulation on MATLAB on CPU and GPU are summarised in table 4.4.1. The computation time depended on the number of integration points used; hence it increased for a model with a large number of nodes using numerous integration points.

<table>
<thead>
<tr>
<th>Total elapsed computation time (s)</th>
<th>CPU</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4538-noded Cylinder</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Integration pt/cell (total 23069)</td>
<td>4674</td>
<td>-*</td>
</tr>
<tr>
<td>4 Integration pt/cell (total 92276)</td>
<td>20911</td>
<td>48</td>
</tr>
<tr>
<td>5 Integration pt/cell (total 115345)</td>
<td>28642</td>
<td>-*</td>
</tr>
<tr>
<td><strong>4663-noded Cylinder</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Integration pt/cell (total 23449)</td>
<td>5520</td>
<td>-*</td>
</tr>
<tr>
<td>4 Integration pt/cell (total 93796)</td>
<td>21755</td>
<td>56</td>
</tr>
<tr>
<td>5 Integration pt/cell (total 117245)</td>
<td>30174</td>
<td>-*</td>
</tr>
</tbody>
</table>

Figure 4.4.1 Summary table of total elapsed computation time (in seconds) comparing CPU and GPU uses -* values could not be obtained for GPU computation times due to MATLAB Parallel Computing Toolbox Licence Checkout failure.
It is evident that facilitating computation through the GPU is much faster than that of CPU by two orders of magnitude and shows potential to deliver surgery simulations in ‘real-time’. However this incredible computation time difference is not a reflection of the CPU’s performance, as MATLAB is most suitable for (and well used as) a prototyping tool rather than a production tool, (Genz et al., 1991).

Previously, Wittek et al. (2009) reported computation time of over 1700 s when predicting the brain deformations using a model with around 50,000 degrees of freedom implemented in non-linear finite element solver LS-DYNA. While this computation time is actually vastly faster than what MATLAB has taken, this is still significantly long to be used in the surgical environment. Implementing CUDA acceleration in surgical simulation is a promising direction to take.
6. Conclusions and Future Work

6.1 Brain Phantom

In this study, needle-phantom sample insertion experiments were conducted to observe and understand the phantom’s response in three situations of needle insertion; effect of change in the needle surface area; and the surface deformation prior to puncture. From the experiments it is understood that:

1) During needle insertion the phantom exhibits non-strain dependency.
2) Increased surface area results in increased force exerted onto the needle.
3) Phantom surface deformation prior to puncture is very miniscule and the phantom gel does not behave like a brain tissue immediately following puncture due to tensile force exerted onto the needle that has not been observed in previous literatures.

Future work involving needle-phantom insertion can include construction of the Sylgard-527 phantom with thin and stiffer layers added to simulate the membrane-like behaviour of the brain. Using this phantom will enable needle insertion experiments where data that is more representative of the brain tissue behaviour can be obtained.

6.2 Using Meshless Method to Predict Deformation Upon Needle Indentation

A verified algorithm of Meshless Total Lagrangian Explicit Dynamics (MTLED) Method has been adapted and used to simulate a nonlinear hyperelastic cylinder solid undergoing a prescribed nodal displacement. Using this method, the force-displacement response of single nodes and the indented set of nodes were generated. The effect of varying the number of integration points per tetrahedral cell was also explored. It is concluded that having one integration point per cell is inadequate and inaccurate to be simulating deformation resulted from point indentation due to instability. Contrary to the published studies accomplishing very similar values from MTLED and the Finite Element Methods (FEM) simulations, our study showed a discrepancy of 25% in two different cylinders simulated. The sources of error are speculated to have resulted from imposing non-linear large strain that is not well-handled by FEM; the use of tetrahedral mesh; and error in entering variables into the MATLAB codes.
While this study focused on the deformation of the volume surface upon needle indentation, the future work in MTLED Method can expand further to simulate needle puncturing the volume of the phantom. Nodes can be added and deleted around the needle tip to simulate the cutting and the boundary conditions can be enforced by defining static and dynamic friction forces. A dynamic friction model was developed that takes both the velocity of the needle and the deformation of the tissue should be taken into consideration.

6.3 CUDA Accelerated Framework

By utilising Graphics Processing Unit (GPU)’s ability to perform parallel computation via its multiple cores, this study has quantitatively confirmed the vast decrease in the computation speed when the MATLAB computation is facilitated by NVIDIA’s CUDA. The computation time decreased by two orders of magnitude when facilitated through GPU compared to CPU.

The final product; control systems of neurosurgical robots with tissue deformation prediction module will most likely be coded in C, and potentially be accompanied by the use of GPU to achieve faster computation time suitable for the ‘real-time’ surgical work-flow.
Appendix A

%==========================================================================
%---- Make Final 3D Force & Displacement Plots --------------------------
% Agnes Kang 2016
%==========================================================================

%----FORCE PLOT
fig = figure;
clf;
view(movie_view);
set(fig, 'DoubleBuffer', 'on');

X1 = x2 + disp_time(:,:,end);
%----- Create first image
disp1 = zeros(nonum_tot,dim);
S = abs(disp1(:,disp_comp));
GraphTitle = 'Force (z-axis)';
u32NumNodes = length(S);
S(u32NumNodes+1, 1) = max(abs(forces_time(:,3,end)));
S(u32NumNodes+2, 1) = min(abs(forces_time(:,3,end)));
X1(u32NumNodes+1, :) = [0 0 0];
X1(u32NumNodes+2, :) = [0 0 0];
S(1:end-2) = NodalVariableForPlotting(x2, forces_time(:,:,end), disp_comp);
% color value
DrawFaces(X1, faces, S, 1);
set(gca, 'xlim', [MinCoord(1) MaxCoord(1)], 'ylim', [MinCoord(2) MaxCoord(2)],...
'zlim', [MinCoord(3) MaxCoord(3)], 'NextPlot', 'replace');
if show_colorbar
colorbar;
end
title(GraphTitle);

%----DISPLACEMENT PLOT
fig2 = figure;
clf;
view(movie_view);
set(fig2, 'DoubleBuffer', 'on');

X1 = x2 + disp_time(:,:,end);
%----- Create first image
disp1 = zeros(nonum_tot,dim);
S = abs(disp1(:,disp_comp));
GraphTitle = 'Displacement (z-axis)';
u32NumNodes = length(S);
S(u32NumNodes+1, 1) = max(abs(disp_time(:,3,end)));
S(u32NumNodes+2, 1) = min(abs(disp_time(:,3,end)));
X1(u32NumNodes+1, :) = [0 0 0];
X1(u32NumNodes+2, :) = [0 0 0];
S(1:end-2) = NodalVariableForPlotting(x2, disp_time(:,:,end), disp_comp);
% color value
DrawFaces(X1, faces, S, 1);
set(gca, 'xlim', [MinCoord(1) MaxCoord(1)], 'ylim', [MinCoord(2) MaxCoord(2)],...
'zlim', [MinCoord(3) MaxCoord(3)], 'NextPlot', 'replace');
if show_colorbar
colorbar;
end