

Subject-Specific Biomechanical Simulation of Brain Indentation Using a Meshless Method

Ashley Horton, Adam Wittek, and Karol Miller

Intelligent Systems for Medicine Laboratory
School of Mechanical Engineering
The University of Western Australia
`kmiller@mech.uwa.edu.au`

Abstract. We develop a meshless method for simulating soft organ deformation. The method is motivated by simple, automatic model creation for real-time simulation. Our method is meshless in the sense that deformation is calculated at nodes that are not part of an element mesh. Node placement is almost arbitrary. Fully geometrically nonlinear total Lagrangian formulation is used. Geometric integration is performed over a regular background grid that does not conform to the simulation geometry. Explicit time integration is used via the central difference method. To validate the method we simulate indentation of a swine brain and compare the results to experimental data.

1 Introduction

Calculation of soft tissue deformation for surgical simulations have typically been based on Finite Element Analysis (FEA) [1,2,3,4,5]. The results from these FEA calculations have been promising, showing that near real-time simulations of surgical procedures, using nonlinear (both geometric and material) biomechanical models, can be achieved with a high level of precision [2,6,7]. Accuracy in FEA, relies heavily on the element mesh which discretises the geometry in question, and when dealing with incompressible continua such as soft tissue, we wish to use only hexahedral elements. When the geometry is highly irregular, an experienced analyst is required to manually create such a mesh which consumes valuable time. This is a major bottleneck; efficient generation of models to be used in real-time simulation of surgical procedures [8].

One solution to this bottleneck is to use a numerical method that does not have such strict discretisation requirements. For example, a meshless algorithm which uses a cloud of unconnected nodes to discretise the geometry instead of elements. Placement of these nodes can be done automatically since their arrangement is almost arbitrary.

Any algorithm to be used in surgical simulations must be capable of producing dynamic results in near real-time. Our algorithm uses Total Lagrangian (TL) formulation to calculate reaction forces. In TL, quantities are calculated with respect to the original configuration as detailed in [9,7]. Hence our method precomputes the constant strain-displacement matrices for each integration cell

and uses the deformation gradient to calculate the full matrix at each time step. For the sake of fast simulations, we also use explicit time integration based on the central difference method.

The purpose of this study is to evaluate the usefulness of our method for 3D surgical simulation. For this purpose, we perform simple indentation experiments on a swine brain and compare the results to those obtained with our method. While indentation is only a simple procedure, calculation of reaction forces on surgical instruments is an important field of study [10,11,12,13]. This is also a necessary validation to make before simulating more complicated procedures.

2 Meshless Method

The following is a brief description of the meshless algorithm we used to simulate indentation of brain tissue. The notation is based on that used in [9] where a left superscript indicates the time when a quantity occurs.

Preprocessing

1. Load simulation geometry in the form of two lists:
 - Node locations.
 - Integration point locations.
2. Load boundary conditions.
3. Loop through list of integration point locations¹. For each integration point:
 - Identify n local nodes associated with the integration point.
 - Create and store the $3 \times n$ matrix $D\Phi(\mathbf{x})$ of moving least squares shape function derivatives $D\Phi_{k,i}(\mathbf{x}) = \frac{\partial \phi_i(\mathbf{x})}{\partial x_k}$ where $k = 1, 2, 3$ $i = 1, 2 \dots n$.
4. Loop through nodes and associate to each a suitable mass.
5. Initialise global nodal displacements $^{-\Delta t}\mathbf{U}$ and $^0\mathbf{U}$.

Solving

In every time step t :

1. Loop through integration points.
 - From precomputed list, load n local nodes and associated shape function derivatives $D\Phi(\mathbf{x})$ for the given integration point \mathbf{x} .
 - Find $n \times 3$ matrix of local nodal displacements tU .
 - Calculate deformation gradient tX .
 - Calculate full strain-displacement matrix tB_L .
 - Calculate second Piola-Kirchoff stress vector $^t\hat{S}$.
 - Calculate and store local nodal reaction forces tf .
2. Put tf for each integration point into global nodal reaction forces vector $^t\mathbf{F}$.
3. Calculate the global nodal displacements for the next time step $^{t+\Delta t}\mathbf{U}$.

¹ Technically we should be looping through integration regions. We use single point integration so this is equivalent.

2.1 Support Domains and Moving Least Squares Shape Functions

Support domain and Moving Least Squares theory deals with the relationship between integration points and nodes. The theory was initially developed by [14] and used in meshless methods such as the Diffuse Element Method and Element Free Galerkin in [15,16]. We use it in this simulation for its simplicity and robustness.

For each integration point, we only require the $n \times 3$ first partial, spatial derivatives of the shape functions $\frac{\partial \phi_k(\mathbf{x})}{\partial x_k}$ for $k = 1, 2, 3$ and n the number of nodes in the support domain.

2.2 Force Calculation

In the total Lagrangian formulation, we calculate forces on nodes local to a given integration point with

$${}^t f = \int_{\text{ov}} {}^t B_L^T {}^t \hat{S} d \text{ov}$$

which we integrate numerically.

The full strain-displacement matrix ${}^t B_L$ has the following construction

$${}^t B_L = \left[\begin{pmatrix} \frac{\partial \phi_1(\mathbf{x})}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial \phi_1(\mathbf{x})}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial \phi_1(\mathbf{x})}{\partial x_3} \\ \frac{\partial \phi_1(\mathbf{x})}{\partial x_2} & \frac{\partial \phi_1(\mathbf{x})}{\partial x_1} & 0 \\ 0 & \frac{\partial \phi_1(\mathbf{x})}{\partial x_3} & \frac{\partial \phi_1(\mathbf{x})}{\partial x_2} \\ \frac{\partial \phi_1(\mathbf{x})}{\partial x_3} & 0 & \frac{\partial \phi_1(\mathbf{x})}{\partial x_1} \end{pmatrix} {}^t X^T, \dots, \begin{pmatrix} \frac{\partial \phi_n(\mathbf{x})}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial \phi_n(\mathbf{x})}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial \phi_n(\mathbf{x})}{\partial x_3} \\ \frac{\partial \phi_n(\mathbf{x})}{\partial x_2} & \frac{\partial \phi_n(\mathbf{x})}{\partial x_1} & 0 \\ 0 & \frac{\partial \phi_n(\mathbf{x})}{\partial x_3} & \frac{\partial \phi_n(\mathbf{x})}{\partial x_2} \\ \frac{\partial \phi_n(\mathbf{x})}{\partial x_3} & 0 & \frac{\partial \phi_n(\mathbf{x})}{\partial x_1} \end{pmatrix} {}^t X^T \right]$$

where each partial derivative is taken from the precomputed $D\Phi(\mathbf{x})$. The deformation gradient ${}^t X^T$ is also calculated with the shape function derivatives and compares the current node locations to the original configuration. Calculation of the second Piola Kirchoff stress ${}^t \hat{S}$ is specific to the material used in the simulation and is discussed in section 3.3. The $3n$ nodal forces calculated at each integration point are combined to form the global force vector ${}^t \mathbf{F}$. These forces are the only data that is stored at each step of the integration point loop.

2.3 Explicit Time Integration

We use Newton’s second law

$$\mathbf{M} \ddot{\mathbf{U}} = {}^t \mathbf{R} - {}^t \mathbf{F}$$

where the forces on the right hand side are the difference between applied (boundary) forces and the reaction forces calculated in section 2.2. Mass is constant, so we apply the finite difference method to acceleration to find the nodal displacements

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

All mass in the simulation is located at the nodes. The mass of a given node is determined by the number of support domains it is included in. The number of support domains a node is included in reveals how many integration points influence and are influenced by the node. Each integration point is allocated a mass according to its volume and this mass is distributed to the nodes in its support domain.

This concludes the description of one time step. If the simulation involves any enforced displacements or contacts, they are enforced here by adjusting ${}^{t+\Delta t}\mathbf{U}$ appropriately.

3 Simulation of Swine Brain Indentation

3.1 Geometry

From MRI images of the brain, we were able to construct the volume and discretise it with 4250 nodes. These nodes were evenly distributed throughout the volume and their placement was done by a computer with no manual adjustment. Similarly, 6371 integration points were placed in the volume, independent of node location. We performed a convergence analysis and confirmed that using more nodes would not increase accuracy. It is worth noting here that a tetrahedral mesh using similar node placement would involve around 18000 elements, consuming much more computation time than our method. Fig. 1 shows the cloud of nodes and integration points whilst Fig. 2 gives a surface visualisation.

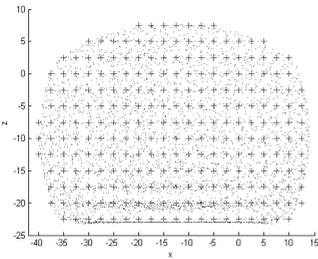


Fig. 1. Side view showing arbitrarily placed nodes (.) and regular integration grid (+)

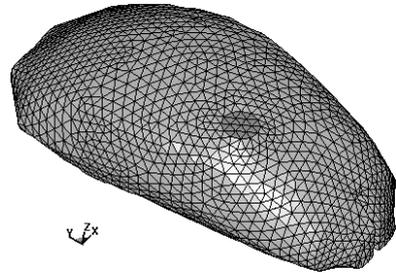


Fig. 2. Surface visualisation. A circle can be seen, representing the nodes that are displaced during indentation. Note that the triangles shown are a visualisation aid rather than elements.

3.2 Boundary Conditions

The boundary conditions described here are chosen to simulate the indentation experiment detailed in section 4. To simulate a fixed base (Fig. 3), we constrain all nodes on the bottom surface of the brain. This is reasonable, given that the real swine brain was glued in place and the mould was significantly stiffer than the brain tissue.

Instead of simulating an indenter (which would require a contact algorithm) we enforce displacement on nodes contained within a 10mm diameter circle on the brain surface. See Fig. 2 for the displaced nodes. The displacement is enforced vertically at a rate of 1mm s^{-1} for a period of 8.0s.

3.3 Material Properties

The purpose of this paper is to verify the algorithm rather than to conduct a complete simulation of actual surgery. It is therefore sufficient to model brain tissue constitutive behaviour with the Neo-Hookean hyperelastic model which has the strain-energy functional

$$W = \frac{\mu}{2}(\bar{I}_1 - 3) + \lambda(J - 1)^2$$

where μ, λ are Lamé parameters, $J = \det(\mathbf{X})$ and \bar{I}_1 is the first strain invariant of the right Cauchy Green deformation tensor tC . From this we calculate the second Piola-Kirchhoff stress tensor

$${}^tS = \lambda J(J - 1) {}^tC^{-1} + \mu J^{-\frac{2}{3}} I$$

and hence form the required vector ${}^t\hat{S}$. Our subject-specific shear modulus is $\mu = 210\text{Pa}$ as was found for this particular brain in [3]. With Poisson's ratio of $\nu = 0.499$ we find $\lambda \approx 105\text{kPa}$. The mass density in our simulation is 10^3kg m^{-3} .

4 Brain Indentation Experiment

For our indentation experiment, the subject brain was that of a 6 month old swine (obtained as a by-product during a commercial slaughtering at Tokyo Shibaura Zooki, Tokyo, Japan) and can be seen in Fig. 3. This brain had a mass of 89.9g and was approximately 92.5mm, 62.5mm, 28.5mm in its major axis, minor axis and height respectively.

Throughout the experiment, the brain was constrained on its base by glue and a custom-made mould (see Fig. 3). To simulate in-vivo conditions, we submerged the brain and mould in a 37°C saline solution.

The indenter was an aluminium cylinder of diameter 10mm and was attached to one end of a force sensor. The other end of the force sensor was attached to a linear motion table which moved vertically. See Fig. 4.

We lowered the indenter at a speed of 1mm s^{-1} so that the brain was indented by 8mm. Forces were recorded throughout the experiment at a sampling rate of 150 Hz. We performed 2 indentation experiments, one on the left and one on the right hemispheres. See section 5 for the recorded forces.

5 Results

At an indentation of 8mm, the swine brain experiments produced reaction forces of 0.181N and 0.125N for the left and right hemisphere respectively. Our simulation of 8mm indentation yielded a force of 0.131N. Looking at Fig. 5 we observe

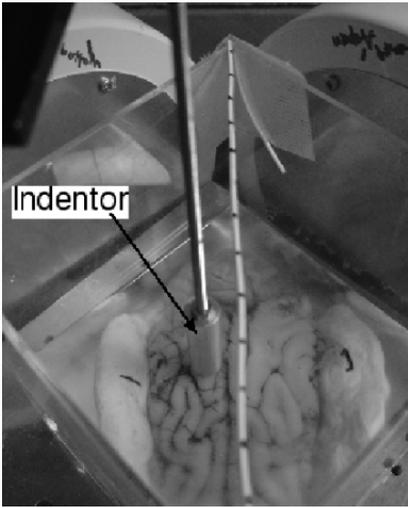


Fig. 3. Swine brain specimen glued onto mould, submerged in warm saline and positioned beneath the indentor

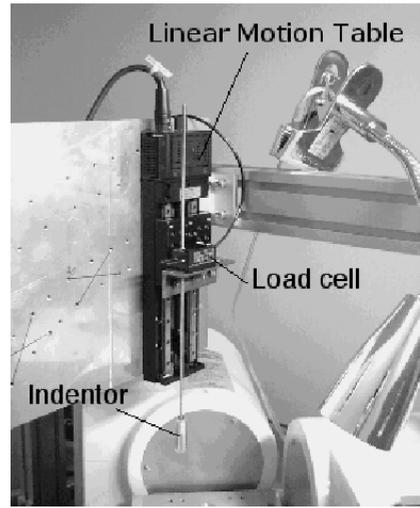


Fig. 4. Apparatus used to perform indentation

that our method is generating results that compare well with reality throughout the simulation.

The simulation was run on a standard desktop personal computer with a 3GHz Pentium 4 processor and 1Gb of internal memory. For this computer, the computation time was 55 seconds (The actual brain indentation took 8 seconds). As efficiency of parallel computing of biomechanical models was reported to be as high as 90% [17], it can be expected that real-time computing of our model can be achieved with as few as 10 Pentium 4 processors.

6 Discussion

Given the results shown in section 5, we consider our simulated forces to be accurate in magnitude. However, we qualitatively notice that the simulation curve shown in Fig. 5 appears more linear than the experimental results which may call for a higher order hyperelastic material (such as Ogden's rubber formulation [18]) to be used in future work. We also note that no pia was included in our simulation which would lead us to expect lower reaction forces [3]. A more realistic future simulation may require simulation of the pia.

Ultimately we note that the simulation involved no tuning and gave results similar to reality. The simulation was performed with automatically positioned nodes and integration points. Future possibilities for our method include simulation of large deformation and topological changes which occur often in surgical procedures but cannot be easily dealt with using FEA.

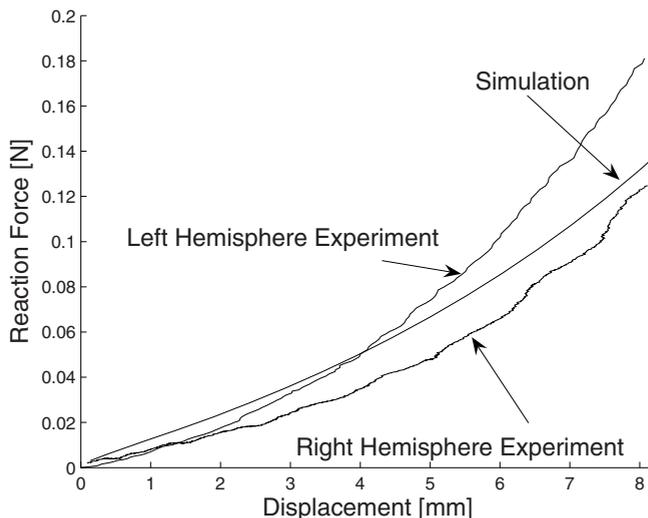


Fig. 5. Reaction force against displacement for our simulation and both swine brain experiments

Acknowledgments

The financial support of the Australian Research Council is gratefully acknowledged, Grant No. DP0664534.

References

1. Cotin, S., Delingette, H., Ayache, N.: Real-time elastic deformations of soft tissues for surgery simulation. In: Hagen, H. (ed.) *IEEE Transactions on Visualization and Computer Graphics*, pp. 62–73. IEEE Computer Society Press, Los Alamitos (1999)
2. Wittek, A., Miller, K., Kikinis, R., Warfield, S.: Patient-specific model of brain deformation: Application to medical image registration. *Journal of Biomechanics* (Accepted 2006)
3. Wittek, A., Dutta-Roy, T., Taylor, Z., Horton, A., Washio, T., Chinzei, K., Miller, K.: Analysis of needle insertion into the brain using non-linear biomechanical model. In: *7th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering* (2006)
4. Nienhuys, H.W.: *Cutting in Deformable Objects*. PhD thesis, University of Utrecht, Utrecht, The Netherlands (2003)
5. Szekely, G., Brechbuhler, C., Hutter, R., Rhomberg, A., Ironmonger, N., Schmid, P.: Modelling of soft tissue deformation for laparoscopic surgery simulation. *Medical Image Analysis* 4, 57–66 (2000)
6. Wittek, A., Kikinis, R., Warfield, S.K., Miller, K.: Brain shift computation using a fully nonlinear biomechanical model. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3750, Springer, Heidelberg (2005)

7. Miller, K., Joldes, G., Lance, D., Wittek, A.: Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in Numerical Methods in Engineering* 23(2), 121–134 (2007)
8. Viceconti, M., Taddei, F.: Automatic generation of finite element meshes from computed tomography data. *Critical Reviews in Biomedical Engineering* 31, 27–72 (2003)
9. Bathe, K.J.: *Finite Element Procedures*. Prentice Hall, New Jersey (1996)
10. DiMaio, S., Salcudean, S.: Needle steering and motion planning in soft tissues. *IEEE Transaction on Biomedical Engineering* 52, 965–974 (2005)
11. Okamura, A.M., Simone, C., O’Leary, M.D.: Force modeling for needle insertion into soft tissue. *IEEE Transactions on Biomedical Engineering*. 51, 1707–1716 (2004)
12. Wittek, A., Laporte, J., Miller, K.: Computing Reaction Forces on Surgical Tools for Robotic Neurosurgery and Surgical Simulation. In: *Proceedings of the Australasian Conference on Robotics and Automation 2004* (2004)
13. Brett, P.N., Parker, T.J., Harrison, A.J., Thomas, T.A., Carr, A.: Simulation of resistance forces acting on surgical needles. *Proceedings of the Institution of Mechanical Engineers Part H: Journal of Engineering in Medicine* 211, 335–347 (1997)
14. Lancaster, P., Salkauskas, K.: Surfaces generated by moving least squares methods. *Mathematics of Computation* 37(155), 141–158 (1981)
15. Nayroles, B., Touzot, G., Villon, P.: Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics* 10(5), 307–318 (1992)
16. Belytschko, T., Lu, Y., Gu, L.: Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 37(2), 229–256 (1994)
17. Koh, B.I., Reinbolt, J.A., Fregly, B.J., George, A.D.: Evaluation of parallel decomposition methods for biomechanical optimizations. *Computer Methods in Biomechanics and Biomedical Engineering* 7(4), 215–225 (2004)
18. Ogden, R.W.: *Non-Linear Elastic Deformations*. Ellis Horwood Ltd. Chichester, Great Britain (1984)