Research Report of Intelligent Systems for Medicine Laboratory

Report # ISML/01/2007, Jan 2007

MESH FREE METHODS FOR SOFT TISSUE DEFORMATION IN COMPUTER INTEGRATED SURGERY II

A. Horton

Intelligent Systems for Medicine Laboratory
School of Mechanical Engineering
The University of Western Australia
35 Stirling Highway
Crawley WA 6009, AUSTRALIA
Phone: + (61) 8 6488 7323
Fax: + (61) 8 6488 1024
Email: ahorton@mech.uwa.edu.au
http://www.mech.uwa.edu.au/ISML/

Contents

1 Introduction

In June 2005, I submitted a research proposal which was followed by an annual report in January 2006 for 'Mesh Free Methods for Soft Tissue Deformation in Computer Integrated Surgery'. This report covers the work I have completed since January 2006 and gives the direction of future work. Fundamental ideas of mesh free methods and motivation for mesh free simulation of soft tissue deformation were given in the research proposal and are not repeated here.

The major work of the past year has been the development of a working 3D fully nonlinear meshless code for solving partial differential equations that arise in solid mechanics. Many smaller codes have been written for 1 and 2 dimensions as well as linear statics but these are not included in this report since they have been superseded. Similarly many algorithms (to handle geometries and output data etc) have been written to support the main code and again these are not detailed in this report.

2 A Total Lagrangian Explicit Dynamics Meshless Algorithm

2.1 Motivation for the Algorithm

In [1] we proposed the use of the Element Free Galerkin method (EFG) introduced by [2] for surgical simulations. In that study, we simulated craniotomy induced brain shift with both EFG and Finite Element Analysis (FEA) in LS DYNA [3, 4]. Although slightly slower, EFG gave very similar results to FEA and can certainly be considered for use in future simulation. The major limitation of LS DYNA's implementation of EFG is the fact that it requires a complete mesh of hexahedral elements that conform to the geometry exactly. By employing this mesh of hexahedral elements, LS DYNA ensures that nodes are evenly distributed through the problem domain which reduces the possibility of near singular shape functions. Background integration is also performed over these elements so the volume of integration matches the volume of the problem's geometry perfectly. However, requiring this structured hexahedral mesh, removes the method's ability to deal with irregular geometries easily.

An algorithm that does not require a geometry conforming hexahedral element mesh is a necessary step towards using meshless methods for fast, efficient simulation of surgical procedures. For this reason, we proposes an algorithm that integrates over a regular background grid. This grid does not conform to the simulation geometry. Independant of this integration grid are the nodes where we calculate displacement. To allow for complicated boundaries etc, our method can accept an almost[1] arbitrary placement of nodes throughout the simulation geometry. Both the integration grid and node placement for our algorithm can be created automatically for any geometry. This distinguishes it from the hexahedral dependant EFG offered in LS DYNA. Our shape functions are created by Moving Least Squares (MLS) [5].

Any algorithm to be used in surgical simulations must be capable of producing dynamic results in real-time. Most commercial dynamic FEA solvers (such as [6, 4]) use Updated Lagrangian (UL) where our algorithm uses Total Lagrangian (TL). The difference between these methods is that in UL, the calculated variables are referred to the previous calculated configuration, as opposed to the initial configuration for TL [7]. As in [8], 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.

In our algorithm, we use explicit time integration based on the central difference method. Unlike implicit time integration, this does not require solving any systems of equations at every time step.

2.2 Algorithm overview

The following is a brief description of the developed algorithm. Notation is based on that used in [7]. More detailed explanations of each part are given in the following sections.

Preprocessing

1. Load simulation geometry Ω 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.
   - Create and store the $3 \times n$ matrix $D\Phi(\mathbf{x})$ of moving least squares shape function derivatives
   $$D\Phi_{k,i}(\mathbf{x}) = \frac{\partial \phi_i(\mathbf{x})}{\partial x_k} \quad k = 1,2,3 \quad i = 1,2 \cdots n$$
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[2]

[1] Totally arbitrary placement will never be possible. An extreme example of this is to imagine all nodes being placed at the same location.

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

- From precomputed list, find n local nodes and associated shape function derivatives $D\Phi(\mathbf{x})$ for the given integration point $\mathbf{x}$.
- Find $n \times 3$ local nodal deformation matrix ${}^{t}U$.
- Calculate deformation gradient ${}^{t}_{0}X$.
- Calculate strain-displacement matrix ${}^{t}_{0}B_L$.
- Calculate second Piola-Kirchoff stress vector ${}^{t}_{0}\hat{S}$ (using material properties).
- Calculate and store local nodal reaction forces

$$ {}^{t}_{0}F = \int_{V^0} {}^{t}_{0}B_L^T \; {}^{t}_{0}\hat{S} \; dV^0 $$

2. Combine all local nodal reaction forces to create global nodal reaction forces vector ${}^{t}_{0}\mathbf{F}$.

3. Calculate global nodal displacements at time $t + \Delta t$ using central difference method

$$ {}^{t+\Delta t}\mathbf{U} = -\Delta t^2 \mathbf{M}^{-1}({}^{t}_{0}\mathbf{F} - {}^{t}\mathbf{R}) + 2\,{}^{t}\mathbf{U} - {}^{t-\Delta t}\mathbf{U} $$

   where $\mathbf{M}$ is the diagonal mass matrix and ${}^{t}\mathbf{R}$ is the boundary condition forces applied at time t.

2.3 Geometry Discretisation

Although a major advantage of meshless methods is the freedom of node placement, this freedom is not total. There is significantly more flexibility than hexahedral meshes can offer, but still not the option to place all nodes on one side of the geometry. What is needed is a roughly even density of nodes throughout the domain.

We also require that all boundary nodes must share a support domain with at least one interior node. To satisfy this requirement, we need to consider our placement of integration points. The general form for numerical integration of a function f over a region V, is

$$ \int f(x)\; dV \approx \sum f(x_i) w_i $$

which leaves the questions of where to place $x_i \in V$ and what weights w_i to use. Available options include:

Background FEA mesh A mesh is created conforming to the nodes and standard FEA integration methods are used.

Nodal integration Some or all nodes are used as single sampling points and the weights are set as the volume associated with each node.

Background grid A regular grid of cells is imposed over the geometry and integration is performed in each cell using standard techniques for a regular 3D region.

Creating a background mesh removes some of the flexibility of the meshfree method. A hexahedral mesh would likely require manual construction for complicated geometries so a tetrahedral mesh is a better option. The tetrahedral integration mesh can be constructed to conform to the existing nodes or not. Conforming to the nodes restricts the possibility of varying integration densities independently to the nodes. A non-conforming tetrahedral mesh has more promise since the simulation volume can be discretised automatically and would be accurately modeled.

Nodal integration is said to be fast and efficient [9] but this is claimed in comparison to background meshes and background grids which use several gauss points per region[3]. The speed of

[3] Nodal integration by nature involves only single point integration.

nodal integration is balanced by its instability [10, 11] so we have not used it(regardless of the claims made in the previous annual report).

In this study we have used a regular background grid with single point integration for each cell. This is fast in theory because of its low number of integration points and in practice because the simplicity lends itself to efficient coding.

2.4 Support Domains and Moving Least Squares Shape Functions

The support domain and MLS theory as initially developed by [5] and used in meshless methods in the Diffuse Element Method of [12] was detailed in the research proposal for this study and is not repeated here. We have experimented with many other shape functions and there are advantages to each, but the simple robustness of the MLS makes it the best choice at this stage.

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

2.5 Mass allocation

All mass in the simulation is located at the nodes, but unlike FEA, we do not believe that all interior nodes are created equal. Each integration cell is allocated a mass based on its volume and density. This mass is split evenly to the n nodes in the support domain of that cell. Many nodes will thus have different masses proportional to the number of support domains that they are included in. This is a good method of distributing weight because nodes in more support domains will receive more forces. For good results, every node should be included in at least 2 support domains, preferably $3-4$ with integration points roughly surrounding the node in question.

The result of a node being included in few support domains, is low mass and unbalanced forces. Since our explicit time integration is based on Newton's second law of motion, this leads to high accelerations and unstable simulations. Even worse is the case of a node which escapes all support domains. No force will be applied to that node, but its massless nature means it must be removed entirely or the diagonal mass matrix will be singular.

The simplest method to avoid nodes with low or zero mass, is to increase the density of the background integration grid. This will reduce the volume and hence mass of each cell, but creates a more even distribution of mass across the entire simulation volume. The obvious cost is in computation.

2.6 Force calculation

From [7] we have the TL formulation

$${}_0^tF = \int_{{}^0V} {}_0^tB_L^T \; {}_0^t\hat{S} \; d\,{}^0V$$

which we integrate numerically.

The full strain-displacement matrix ${}_0^tB_L$ has the following construction

$$
\begin{aligned}
{}_0^tB_L &= \left[{}_0^tB_L^{(1)},\ {}_0^tB_L^{(2)},\ \cdots\ ,{}_0^t B_L^{(n)}\right] \\
{}_0^tB_L^{(i)} &={}_0^t B_{L0}^{(i)}\ {}_0^tX^T \\
{}_0^tB_{L0}^{(i)} &= \begin{pmatrix} \frac{\partial\phi_i(\mathbf{x})}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial\phi_i(\mathbf{x})}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial\phi_i(\mathbf{x})}{\partial x_3} \\ \frac{\partial\phi_i(\mathbf{x})}{\partial x_2} & \frac{\partial\phi_i(\mathbf{x})}{\partial x_1} & 0 \\ 0 & \frac{\partial\phi_i(\mathbf{x})}{\partial x_3} & \frac{\partial\phi_i(\mathbf{x})}{\partial x_2} \\ \frac{\partial\phi_i(\mathbf{x})}{\partial x_3} & 0 & \frac{\partial\phi_i(\mathbf{x})}{\partial x_1} \end{pmatrix}
\end{aligned}
$$

where every element of ${}_0^tB_{L0}^{(i)}$ is taken from the precomputed $D\Phi(\mathbf{x})$. This update is fast and efficient since all shape function calculations are done in the preprocessing stage. An UL formulation would involve recalculating shape functions and their derivatives at every time step and for every integration point.

2.7 Explicit time integration

The $3n$ nodal forces calculated at each integration point are combined to form the global force vector ${}_0^t\mathbf{F}$. These forces are the only data that is stored at each step of the integration point loop.

We use Newton's second law

$$\mathbf{M}\ {}^t\ddot{\mathbf{U}} ={}^t \mathbf{R} -{}_0^t \mathbf{F} \tag{1}$$

where the forces on the right hand side are the difference between applied (boundary) forces the reaction forces calculated in section 2.6. Mass is constant, so we apply the finite difference method to acceleration to find

$$\ddot{\mathbf{U}} \approx \frac{1}{\Delta t^2}\left({}^{t-\Delta t}\mathbf{U} - 2\ {}^t\mathbf{U} +\ {}^{t+\Delta t}\mathbf{U}\right) \tag{2}$$

Putting (2) into (1) gives us

$${}^{t+\Delta t}\mathbf{U} = \Delta t^2\mathbf{M}^{-1}({}^t\mathbf{R} -\ {}_0^t\mathbf{F}) + 2\ {}^t\mathbf{U} -\ {}^{t-\Delta t}\mathbf{U} \tag{3}$$

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

3 A simplified 3D experiment

The following example is given to show how our algorithm can be used in the place of LS DYNA's existing code. The example is a simple one, but contains all the major components (3D, finite deformation, nonlinear material models, irregular geometry, multiple parts and contact definitions) of a more complicated simulation. Parallel simulations are run in both LS DYNA and with our algorithm. The major simplification here is that only a thin slice of brain is simulated, rather than a complete solid.

3.1 Base Images and Geometry

The geometry for our patient-specific, brain shift simulation is based on pre-operative MRIs (Figure 1) segmented with 3D Slicer [13] by the Surgical Planning Laboratory at Brigham and Women's Hospital and Harvard Medical School We choose our experimental geometry to be a thin, 3D slice

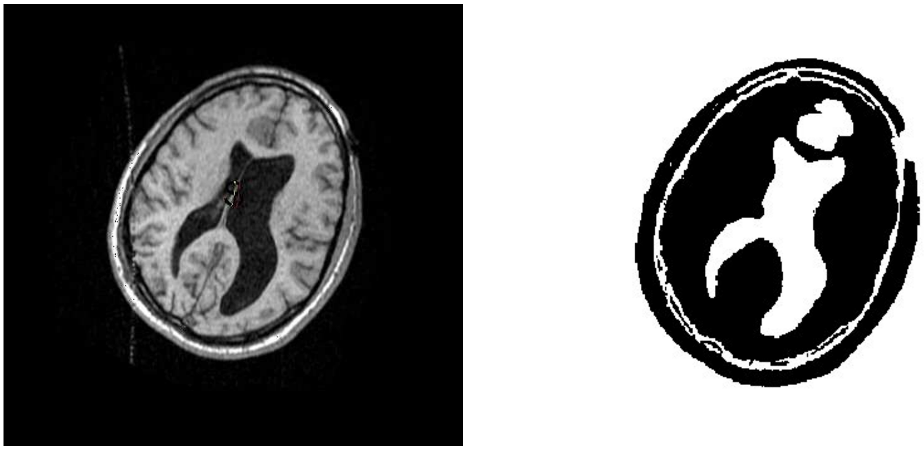


Figure 1: Left: Patient specific data. MRI showing ventricles (dark centre), brain and skull. A tumor is present, but experience and skill are required to accurately differentiate it from healthy brain tissue. Right: Segmented MRI clearly showing ventricles, brain, skull and tumor.

of brain obtained by taking the 2D segmented image shown in Figure 1, and applying a uniform thickness of 5mm using the Ansys pre-processor [14]. Complete justification for this is given in [1], but the main point to note is that full 3D formulations are employed throughout this study.

3.2 Discretising the Geometry

For the LS DYNA simulation, we use the same node arrangement and hexahedral element used in [1]. Integration is done over the hexahedral elements shown in Fig 2

Unlike LS DYNA, our algorithm can take randomly placed nodes. However, a truly random placement is often not useful, since shape functions cannot be effectively created when several nodes in a given region exist in a straight line. To avoid poor node placement, we create a tetrahedral mesh automatically with Ansys and record node locations before discarding element information. This method ensures that we get a roughly even distribution of nodes.

Integration in our algorithm is done over a regular background mesh of varying density as shown in Figure 2. This background mesh is initially much larger than the volume of interest but we reject any cell whose centre is outside the brain volume. For most interior regions, we use a cell with volume of 500mm3. On the boundary and around the tumor, a higher resolution is needed to deal with irregular geometry and/or higher density of nodes. In these areas the volume of the cells drops to 125mm3. All cells are under-integrated (single Gauss point).

It is important to note that while our background cells appear to be standard hexahedral elements, they do not conform to boundaries. This would appear to give less accurate results, but we see in section 3.5 that this is not the case. Of course, if more accuracy was required then additional, smaller cells can be added and some large cells can be broken up to account for any geometry at any resolution. No consideration is needed when multiple small cells share a border with one large cell. All of this can be done automatically by a simple code.

3.3 Material Properties

In our brain slice, there are three parts that require modeling of material properties, the brain, ventricles and tumor. For the brain and tumor, we use the simplest fully nonlinear material formulation, Neo-Hookean [7]. We are justified in using the simplest possible nonlinear form because

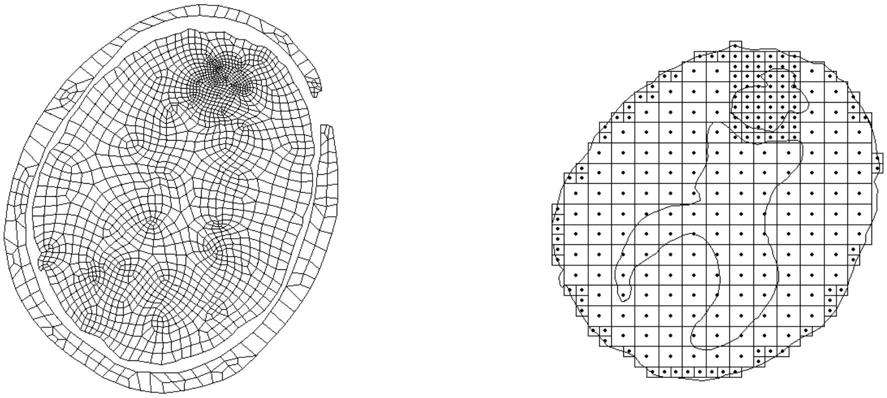


Figure 2: Left: Hexahedral element mesh needed for LS DYNA simulation. Right: Background mesh used in our algorithm. The dot in the centre of each cell is the single integration point Note that in our algorithm, the skull does not have a background mesh.

at this point we are primarily interested in comparing our numerical algorithm to LS DYNA's.

The Neo-Hookean formulation is obtained by simplifying the Mooney-Rivlin [7, 15] rubber model used in LS DYNA. According to [4, 3], Mooney-Rivlin rubber in LS DYNA has the strain energy density function

$$\begin{aligned} W &= A(I-3) + B(II-3) + C(III^{-2}-1) + D(III-1)^2 \\ C &= \frac{A}{B} \\ D &= \frac{A(5\nu-2)+B(11\nu-5)}{2(1-2\nu)} \end{aligned}$$

Where I, II and III are the first, second and third invariants of the right Cauchy-Green deformation tensor.

The parameters we use for healthy brain tissue are A=1052 Pa, B=0 Pa and v=0.49. Note that by setting B=0 we are simplifying the model to almost incompressible Neo-Hookean since v is Poisson's ratio. The value for A is derived from [16, 17]. For the tumor, we use the same justification as [18] to assume the properties of healthy brain.

We model the ventricles as soft, compressible elastic solids with Young's modulus 10Pa and Poisson's ratio of 0.1. Justification for this is given in [18].

3.4 Loading and Boundary Condition

The brain, ventricles and tumor are modeled as having no gaps between parts. They are connected by sharing nodes along part boundaries. Between the brain and skull, there are many materials with complicated properties. For this study, it is sufficient to consider this subarachnoid space to be a gap of approximately 2-4mm between these two parts [18]. Appropriate contact surfaces are defined between the brain and skull with the assumption of no sliding friction.

We constrain all nodes in the direction normal to the slice to simulate the existence of the missing brain volume. During any neurosurgical procedure, the head would be totally anchored and the skull significantly stiffer than its contents. In both our simulations we treat the skull as a rigid and anchored body. In our algorithm this means that the skull only needs to be modeled along its contact region (interior surface). This explains why Figure 2 does not show any background mesh for the skull.

We simulate external pressure on the brain (causing brain shift) by enforcing a displacement on nodes near the craniotomy as shown in Figure 3 The displacement is enforced for 1s before removing the enforced displacement and allowing relaxation for 1s (total simulation time is 2s).

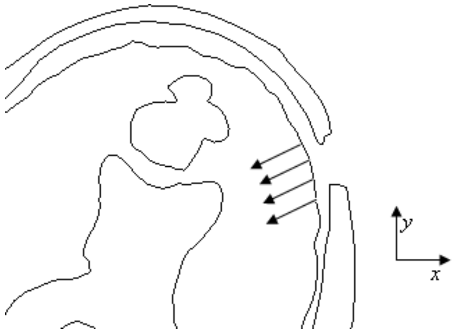


Figure 3: Displacement is enforced on nodes near craniotomy.

3.5 Results

Table 1 shows the resulting displacement of the centre of mass for the brain, ventricles and tumor. We see that the relative displacement differences between the two algorithms are of the order 10^{-2}. Also, the absolute displacement differences are safely less that 0.85mm which is the resolution of the MRIs used in this study [19]. At this level of accuracy, it would now be more important (for this simulation) to refine the initial state than to refine the numerical algorithm.

Figure 4 visually shows the high agreement between simulation results. Outlines from both simulations are shown and line up almost perfectly. As would be expected, the greatest difference is visible (as a slightly thicker outline) at the contact region where the two algorithms differ slightly. We should also be aware that slightly different nodal discretisations of the part boundaries give us a minor discrepancy which is not the fault of either algorithm. Note that it is the nodes, not the integration cells, that discretise the part boundaries.

Part	Direction	Initial Coordinate (mm)	Displaced Coordinates		Relative Differences
			LS DYNA (mm)	Our Algorithm (mm)	
Brain	x	13.3012	0.7133	10.7610	0.0184
	y	7.1770	3.9657	4.2143	0.0774
Ventricles	x	15.3711	12.4832	12.7567	0.0947
	y	-4.8039	-8.1255	-7.8064	0.0961
Tumor	x	34.4503	31.6203	31.8396	0.0775
	y	57.732	54.8762	55.0402	0.0574

Table 1: Coordinates of the centre of mass of the brain, ventricles and tumor. Coordinates in the z direction (normal to the slice) are not shown, since they are all 2.5mm and no differences are seen.

These results conclude the thin slice experiment, having shown that our algorithm gives good results and is worth investigating further. These results we published in the Proceedings of the

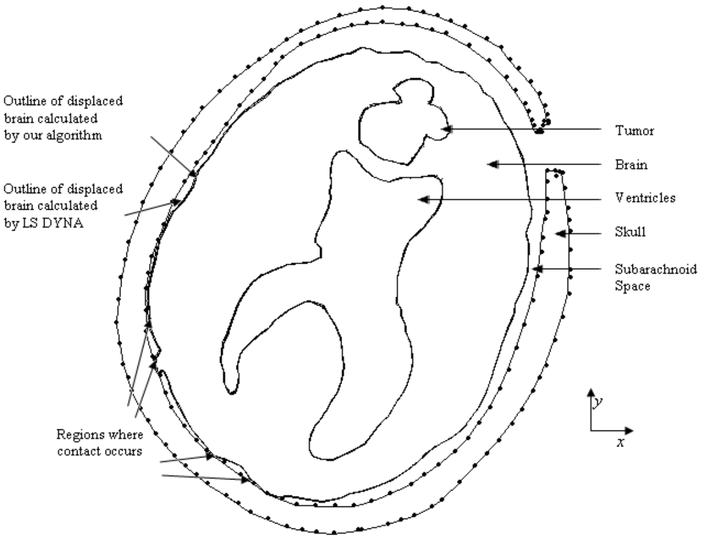
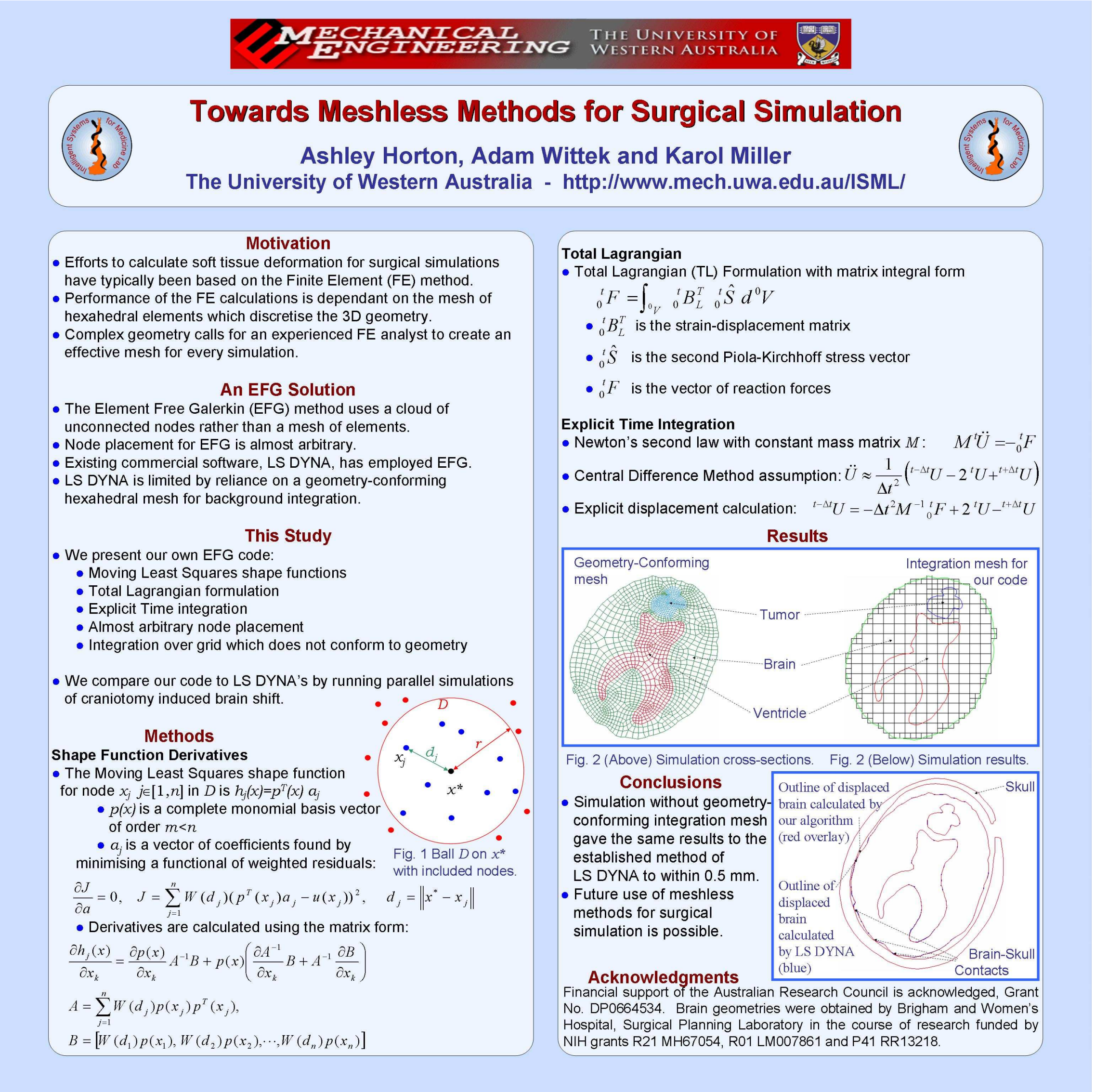

Outline of displaced brain calculated by our algorithm
Outline of displaced brain calculated by LS DYNA
Regions where contact occurs
Tumor
Brain
Ventricles
Skull
Subarachnoid Space
y
x

Figure 4: Outlines of the displaced brain, ventricles and tumor from both simulations are displayed here on the same image to show the similarity of results. A greater difference in results is visible near the contact edges. The dots around the skull are reference markers, to ensure that the images are aligned correctly.

Computational Biomechanics for Medicine workshop at MICCAI 2006. A poster presentation was made and the poster can be seen in Figure 5.



MECHANICAL ENGINEERING THE UNIVERSITY OF WESTERN AUSTRALIA

Towards Meshless Methods for Surgical Simulation

Ashley Horton, Adam Wittek and Karol Miller
The University of Western Australia - http://www.mech.uwa.edu.au/ISML/

Motivation

- Efforts to calculate soft tissue deformation for surgical simulations have typically been based on the Finite Element (FE) method.
- Performance of the FE calculations is dependant on the mesh of hexahedral elements which discretise the 3D geometry.
- Complex geometry calls for an experienced FE analyst to create an effective mesh for every simulation.

An EFG Solution

- The Element Free Galerkin (EFG) method uses a cloud of unconnected nodes rather than a mesh of elements.
- Node placement for EFG is almost arbitrary.
- Existing commercial software, LS DYNA, has employed EFG.
- LS DYNA is limited by reliance on a geometry-conforming hexahedral mesh for background integration.

This Study

- We present our own EFG code:
  - Moving Least Squares shape functions
  - Total Lagrangian formulation
  - Explicit Time integration
  - Almost arbitrary node placement
  - Integration over grid which does not conform to geometry
- We compare our code to LS DYNA's by running parallel simulations of craniotomy induced brain shift.

Methods

Shape Function Derivatives

- The Moving Least Squares shape function for node x_j $j \in [1,n]$ in D is $h_j(x)=p^T(x)\, a_j$
  - $p(x)$ is a complete monomial basis vector of order $m<n$
  - a_j is a vector of coefficients found by minimising a functional of weighted residuals:

$$\frac{\partial J}{\partial a}=0, \quad J=\sum_{j=1}^{n} W(d_j)(p^T(x_j)a_j-u(x_j))^2, \quad d_j=\left\|x^*-x_j\right\|$$

Fig. 1 Ball D on x^* with included nodes.

- Derivatives are calculated using the matrix form:

$$\frac{\partial h_j(x)}{\partial x_k}=\frac{\partial p(x)}{\partial x_k}A^{-1}B+p(x)\left(\frac{\partial A^{-1}}{\partial x_k}B+A^{-1}\frac{\partial B}{\partial x_k}\right)$$

$$A=\sum_{j=1}^{n} W(d_j)p(x_j)p^T(x_j),$$

$$B=\left[W(d_1)p(x_1), W(d_2)p(x_2),\cdots,W(d_n)p(x_n)\right]$$

Total Lagrangian

- Total Lagrangian (TL) Formulation with matrix integral form

$${}_0^tF=\int_{{}^0V} {}_0^tB_L^T \; {}_0^t\hat{S} \; d\,{}^0V$$

  - ${}_0^tB_L^T$ is the strain-displacement matrix
  - ${}_0^t\hat{S}$ is the second Piola-Kirchhoff stress vector
  - ${}_0^tF$ is the vector of reaction forces

Explicit Time Integration

- Newton's second law with constant mass matrix M: $M\,{}^t\ddot{U}=-{}_0^tF$
- Central Difference Method assumption: $\ddot{U}\approx\frac{1}{\Delta t^2}\left({}^{t-\Delta t}U-2\,{}^tU+{}^{t+\Delta t}U\right)$
- Explicit displacement calculation: ${}^{t-\Delta t}U=-\Delta t^2M^{-1}\,{}_0^tF+2\,{}^tU-{}^{t+\Delta t}U$

Results

Fig. 2 (Above) Simulation cross-sections. Fig. 2 (Below) Simulation results.

Outline of displaced brain calculated by our algorithm (red overlay)
Skull
Outline of displaced brain calculated by LS DYNA (blue)
Brain-Skull Contacts

Conclusions

- Simulation without geometry-conforming integration mesh gave the same results to the established method of LS DYNA to within 0.5 mm.
- Future use of meshless methods for surgical simulation is possible.

Acknowledgments

Financial support of the Australian Research Council is acknowledged, Grant No. DP0664534. Brain geometries were obtained by Brigham and Women's Hospital, Surgical Planning Laboratory in the course of research funded by NIH grants R21 MH67054, R01 LM007861 and P41 RR13218.

Figure 5: Poster presented at the Computational Biomechanics for Medicine workshop at MICCAI 2006.

4 Future work

The algorithm developed in the past year and outlined in this report forms the basis for the final stages of work in my Phd. Over the next 6 months I intend to take this algorithm and:

- Find optimal parameters to increase accuracy and decrease computation expense.
- Extend current FEA theory for explicit time integration to be useful for meshless shape functions.
- Develop code to directly automatically convert segmented MRI images into useful geometry data (without ever creating surfaces and volumes as is standard in FEA).
- Implement 3D contact algorithms for meshless geometries.

- Construct a neurosurgical simulation using complete 3D geometry and compare results to pre and intraoperative MRIs.

Most of the base research for the above goals has already been completed, with only the application left to be done. The following table was presented in my research proposal in June 2005 and shows the intended timeframe for this Phd.

Research in the months since 27/01/05	6	12	18	24	30	36
Literature review	✓					
Begin using LS-DYNA	✓	✓				
Extend LS-DYNA to real data		✓	✓			
Consider improvements to existing software			✓			
Write meshless code			✓	✓		
Extend code for real applications				✓	✓	
Write thesis						✓

I consider myself to be on target to complete research by the middle of this year and begin writing my thesis in the second half.

References

[1] A. Horton, A. Wittek, and K. Miller. Computer Simulation of Brain Shift Using an Element Free Galerkin Method. In *in 7th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering. Antibes, Cote d'Azur, France*, 2006.

[2] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229 – 256, 1994.

[3] LSTC. *LS DYNA Theoretical Manual*. Livermore Software Technology Corporation, Livermore, Claifornia, 1998.

[4] LSTC. *LS DYNA Keyword Manual*. Livermore Software Technology Corporation, Livermore, Claifornia, 2003.

[5] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141 – 58, 1981.

[6] Abaqus. Abaqus online documentation. 2007.

[7] K-J Bathe. *Finite Element Procedures*. Prentice Hall, New Jersey, 1996.

[8] K. Miller, G. Joldes, D. Lance, and A. Wittek. Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Communications in Numerical Methods in Engineering*, 23(2):121–134, 2007.

[9] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3 – 47, 1996.

[10] J-S. Chen, C-T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal For Numerical Methods in Engineering*, 50:435–466, 2001.

[11] K. Y. Sze, J. S. Chen, N. Sheng, and X. H. Liu. Stabilized conforming nodal integration: exactness and variational justification. *Finite Elem. Anal. Des.*, 41(2):147–171, 2004.

[12] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5):307 – 318, 1992.

[13] Brigham Surgical Planning Laboratory, Women's Hospital, and Harvard Medical School. 3dslicer, medical visualization and processing environment for research. 2004.

[14] Ansys. Ansys 9.0. 2004.

[15] M. Mooney. A theory of large elastic deformation. *Journal of Applied Physics*, 11:582–592, 1940.

[16] K. Miller. *Biomechanics of Brain for Computer Integrated Surgery*. Publishing House of the Warsaw University of Technology, Warsaw, 2002.

[17] K. Miller and K. Chinzei. Mechanical properties of brain tissue in tension. *Journal of Biomechanics*, 35(4):483 – 490, 2002. Brain tissue;.

[18] A. Wittek, K. Miller, R. Kikinis, and S K. Warfield. Patient-specific model of brain deformation: Application to medical image registration. *Journal of Biomechanics*, 2007.

[19] G. Bourgeois, M. Magnin, A. Morel, S. Sartoretti, T. Huisman, E. Tuncdogan, D. Meier, and D. Jeanmonod. Accuracy of mri-guided stereotactic thalamic functional neurosurgery. *Neuroradiology*, 41(9):636–645, 1999.