Coupling Finite Element and Mesh-free Methods for Modelling Brain Deformation in Response to Tumour Growth

Jamie Berger[1], Ashley Horton[1], Grand Joldes[1], Adam Wittek[1], Karol Miller[1]

May 29, 2008

Intelligent Systems for Medicine Lab
[1]The University of Western Australia
35 Stirling Highway
Crawley WA 6009, Australia
Email: kmiller@mech.uwa.edu.au
www.mech.uwa.edu.au/ISML

Abstract

Very little is known about the deformation effects of tumour growth within the brain. Computer simulations have the potential to calculate such deformations. A method for computing localised high deformations within the brain's soft tissue is presented. Such knowledge would be significant towards neuroscience and neurosurgery, particularly for quantifying tumour aggressiveness, therapy planning, as well as surgical planning and simulation. A Finite Element mesh used in the vicinity of a growing tumour is very quickly destroyed and cannot be used reliably unless complicated automatic re-meshing exists. Mesh-free methods are capable of handling much larger deformations, however are known to be less reliable that Finite Element analysis for moderate deformations. A mixed-mesh approach utilises mesh-free regions within localised high-deformation zones, with the remaining model comprised of a Finite Element mesh. In this study, a new algorithm is proposed coupling the Finite Element and Element Free Galerkin methods for use in applications of high localised deformation, such as brain tumour growth. The algorithm is verified against a number of separate Finite Element and mesh-free problems solved via validated/commercial software. Maximum errors of less than 0.85 *mm* were maintained, corresponding to the working resolution of an MRI scan. A mixed-mesh brain model is analysed with respect to different tumour growth volumes located behind the left ventricle. Significant displacements of up to 9.66 *mm* surrounding a 4118 mm^3 sized tumour are noted, with 14.5% of the brain mesh suffering deformation greater than 5 *mm*.

Contents

In biomechanics of soft tissues, it is common to encounter extreme deformations that cannot be handled by traditional modelling methods, such as the Finite Element method. An example of this is brain tumour growth. Very little is known within this field and such a model should be useful for medical use, particularly in quantifying tumour aggressiveness, therapy planning, as well as surgical planning and simulation. The three dimensional mechanical response of the brain is highly non-linear, involving extremely complex constitutive models and geometry, which is very time consuming to model using public Finite Element (FE) software (Miller, Taylor, *et al.*, 2005). Furthermore the Finite Element method on its own will be inaccurate and problematic for modelling the brain deformation response to tumour growth, since the mesh surrounding the tumour is easily distorted, consequently destroying elements. Complicated re-meshing can combat this, however it is extremely time consuming. Alternatively a solely mesh-free Element Free Galerkin (EFG) model will be capable of handling larger deformations and topology changes (Li and Liu, 2004). Despite this the method is known to be less reliable than Finite Element analysis for moderate deformations and suffers from Dirichlet boundary difficulties (Fries and Matthies, 2003). A coupled Finite Element / Element Free Galerkin approach is proposed to overcome the shortcomings of each individual method, by placing a mesh-free domain around the tumour affected location, with the remaining brain tissue modelled as a hexahedral mesh.

1. Background Theory

The Finite Element method is a numerical approach for solving systems of partial differential equations, by discretising the domain into small volumes (elements) and estimating the solution in each of the elements via shape functions. The estimated solutions are then substituted into integral differential equations of the weak form with the residuals minimised (Bathe, 1996). The Element Free Galerkin method (Horton, 2006) conducts the same process, without requiring the connectivity of elements. Shape functions are not within elements but small neighbourhoods of nodes, called support domains, each of which is associated to an integration point (Belytschko, Krongauz, *et al.*, 1996). The Moving Least Squares formulation is used to minimise residuals within the EFG method.

2. Coupled Finite Element / Element Free Galerkin Method

Mixed-mesh coupling is achieved by constructing interface support domains in between FE and EFG boundaries, as shown in *Figure 2.1.* Interface support domains are created by allowing the EFG nodal support domains to extend into the FE region, consuming nodes. They follow the same numerical approach as the EFG method. FE nodes that exist within a nodal support domain are considered by both the EFG and FE methods with their nodal forces summed together.

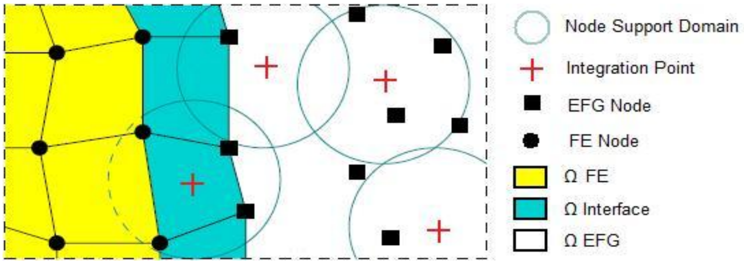

Figure 2.1: Coupled FE/EFG domain with interface region and support domains highlighted.

3. Total Lagrangian Formulation

The Total Lagrangian (TL) formulation is a general convergent method for dealing with materially non-linear effects and large deformations. It is derived from the Principle of Virtual Displacements. The TL formulation references all static and kinematic variables to the initial configuration of the system, which means all derivatives with respect to spacial coordinates can be precomputed. Hence, the number of mathematical operations performed in each time step of the TL algorithm is reduced, and is therefore favourable for surgical simulation (Bathe, 1996, and Miller, Joldes, *et al.*, 2007).

3.1 Explicit Integration

The method of integration for the TL formulation is very important for the performance of the algorithm. Computing the deformation field and internal forces within a soft tissue organ, such as the brain, requires the utilisation of an efficient integration method for integrating static or dynamic equations within the time domain. Both implicit and explicit integration methods are suitable for this problem. The Central Difference Method of explicit integration is chosen over implicit integration as it does not require the solution of simultaneous non-linear equations and performs much faster on materials of very low stiffness, such as the brain (Miller, Joldes, *et al.*, 2007).

The Central Difference Method is derived from Newton's second law as shown,

$$\boldsymbol{M}\frac{\partial^2\,{}^t_0u}{\partial t^2} = {}^{\boldsymbol{t}}\boldsymbol{R} - {}^{\boldsymbol{t}}_{\boldsymbol{0}}\boldsymbol{F} \tag{3.1}$$

$$\frac{\partial^2\,{}^t_0u}{\partial t^2} \approx \frac{1}{\Delta t}\left({}^{t-\Delta t}u_i - 2\,{}^tu_i + {}^{t+\Delta t}u_i\right) \tag{3.2}$$

$${}^{t+\Delta t}u_i = \frac{\Delta t^2}{M}\left({}^tR_i^{(n)} - {}^tF_i^{(n)}\right) + 2\,{}^tu_i - {}^{t-\Delta t}u_i \tag{3.3}$$

where, $\boldsymbol{M}$ represents the nodal mass, Δt the simulation time-step, ${}^tF_i^{(n)}$ and ${}^tR_i^{(n)}$ the net force and reaction force on the node. The previous, current, and future nodal displacements are described by ${}^{t-\Delta t}u_i$, ${}^tu_i$, and ${}^{t+\Delta t}u_i$ respectively.

Unfortunately two issues arise with the use of explicit integration. Firstly explicit methods are conditionally stable, requiring time-steps below a critical value (Joldes, 2006). Furthermore explicit methods cannot obtain static solutions unless damping is used. Quasi-static simulations can be conducted such that the load is applied very slowly over a long period of time. The longer the simulation time, the closer the solution is to a static one.

3.2 Total Lagrangian Explicit Dynamics Algorithm

The Total Lagrangian Explicit Dynamics (TLED) algorithm combines the TL formulation with explicit integration. The TLED algorithm has been successfully utilised for non-linear soft tissue deformations in FE and EFG numerical approaches individually (Horton, 2006, and Miller, Joldes, *et al.*, 2007). Both numerical methods have provided reliable and accurate results using the TLED algorithm, thus it is desirable for use in the coupled FE/EFG approach. The following psuedocode describes the TLED algorithm in relation to solving mixed-mesh problems:

```
Pre-process:
    1. Load simulation geometry (FE & EFG nodes), elements, integration points, and boundary
       conditions.
    2. Pre-compute element / integration point spacial derivatives and support domains.
    3. Calculate appropriate mass matrix, $\boldsymbol{M}$, for allocating mass to every node.
    4. Initialise displacements.
Begin Time Loop:
    - Loop Over Elements / Support Domains
        5. Update FE and EFG displacements from previous time-step.
        6. Compute displacement derivatives, $\boldsymbol{u_{i,j}}$, and deformation gradient, $\boldsymbol{X}$:
            $\boldsymbol{u_{i,j}} = \boldsymbol{N}'\,{}^{\boldsymbol{t}}\boldsymbol{u}$          ($\boldsymbol{N'}$ – FE/EFG shape function derivatives matrix)
            ${}^{\boldsymbol{t}}_{\boldsymbol{0}}\boldsymbol{X} = \boldsymbol{u_{i,j}} + \boldsymbol{I}$          ($\boldsymbol{I}$ – Identity matrix)
        7. Compute inverse of Right Cauchy-Green Deformation tensor, $C_{ij}$, and Second Piola-Kirchoff
```

stress vector, ***S***, at each integration point:

$$C_{ij}=({}^t_0\boldsymbol{X}^T\,{}^t_0\boldsymbol{X})^{-1}$$

$$S_{ij}=\mu(I-C_{ij})+\lambda J(J-1)C_{ij}$$ (μ, λ – Lame` parameters; J – Jacobian determinant)

$${}^t_0\boldsymbol{S}=\begin{bmatrix} {}^t_0S_{11} & {}^t_0S_{12} & {}^t_0S_{31} \\ {}^t_0S_{12} & {}^t_0S_{22} & {}^t_0S_{23} \\ {}^t_0S_{31} & {}^t_0S_{23} & {}^t_0S_{33} \end{bmatrix}$$

8. Compute elemental/support domain forces, ***F***, using Gaussian Quadrature:

$${}^t_0\boldsymbol{F}=\int_{{}^0V} {}^t_0\boldsymbol{X}\,{}^t_0\boldsymbol{S}\,\boldsymbol{N}'\,d\,{}^0V$$

9. Combine local reaction forces to obtain net nodal reaction forces at time, *t*.
10. Explicitly determine displacements, $u^{t+\Delta t}$, using Central Difference formula:

$${}^{t+\Delta t}u_i=\frac{\Delta t^2}{M}({}^tR_i-{}^tF_i)+2\,{}^tu_i-{}^{t-\Delta t}u_i$$ (${}^tR_i$ – Total nodal reaction force at time, t)

4. Program Implementation

The implementation of the tailor made localised soft tissue deformation simulator is divided into three main sections:

- Preprocessor - Reads in mixed mesh model and constraints, pre-computing all initial configuration stationary properties.
- Analysis Solver - Executes the main time loop performing calculations in accordance with the Total Lagrangian Explicit Dynamics Algorithm.
- Postprocessor - Uses series of visualisation tools to view and identify implications of analysis.

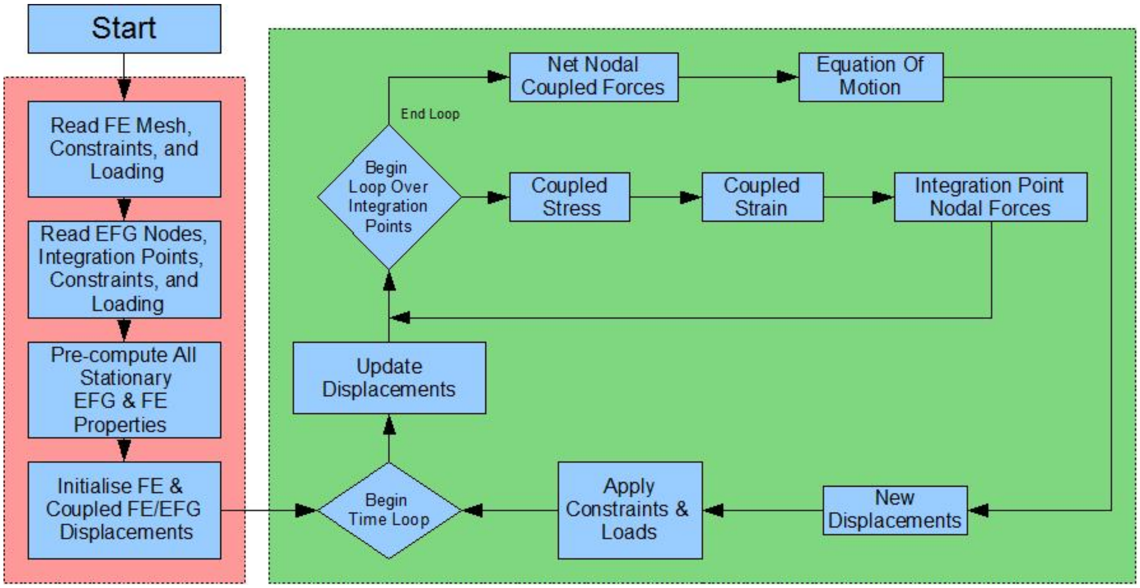
Start
Read FE Mesh, Constraints, and Loading
Read EFG Nodes, Integration Points, Constraints, and Loading
Pre-compute All Stationary EFG & FE Properties
Initialise FE & Coupled FE/EFG Displacements
Begin Time Loop
Update Displacements
Begin Loop Over Integration Points
End Loop
Net Nodal Coupled Forces
Equation Of Motion
Coupled Stress
Coupled Strain
Integration Point Nodal Forces
New Displacements
Apply Constraints & Loads

Figure 3.1: Block diagram of the preprocessing phase (red) and analysis solver (green).

The script language, MATLAB, was chosen for the preprocessing and postprocessing stages of the simulator. MATLAB is a very powerful, high level, language, containing many built in functions, which are of particular use for dealing with matrices. This was beneficial for the preprocessing stage, which requires many large matrix operations. Furthermore MATLAB holds significant advantages for the postprocessing phase as it has excellent visualisation tools, allowing for advanced analysis of the results.

Unfortunately MATLAB performs much slower than compiled languages, hence it was not feasible for use in the implementation of the analysis solver. The functional programming language, C, was chosen as it is very fast, with inbuilt optimisation compilation abilities.

4.1 Preprocessor

The implementation of the preprocessor phase can be broken down into a number of smaller subsections. It should be noted that the computational performance of this phase is less important than the analysis solver, since for coarse mixed-meshes, the run time is negligible in comparison.

4.1.1 Simulation Properties

All of the major simulation properties are user defined and must be set prior to running the preprocessor. This includes the maximum displacement, time-step, and the total simulation time. The simulation properties provide enough information to setup the deformation loading curve, for applying incremental displacements at each time-step. The default deformation loading curve is defined in (4.1),

$$d(t)=[10-15(\frac{t}{T})+6(\frac{t}{T})^2](\frac{t}{T})^3 \quad , \tag{4.1}$$

where T is the total simulation time.

4.1.2 Mixed-Mesh Reader

The nodes, elements, and boundary conditions of the mixed-mesh are read in from ABAQUS output files. Two ABAQUS output files are required, both containing nodal positions and boundary conditions for each method, with the FE output file containing additional information about element composition. The data format of the information read from the FE and EFG ABAQUS files is displayed in *Table 4.1.* A coupled list of all nodes, $\mathbf{X}_{\mathbf{coupled}}$, is then formed by combining $\mathbf{X}_{\mathbf{FE}}$ and $\mathbf{X}_{\mathbf{EFG}}$ in that order.

FE ABAQUS Output	FE Nodes	$\mathbf{X}_{\mathbf{FE}}$ ($N_{\text{nodes-FE}}$ x 3) matrix of all FE node locations. Each row of the matrix corresponds to the FE node number.
	Elements	$\mathbf{E}_{\mathbf{FE}}$ ($N_{\text{elements-FE}}$ x 8) matrix of all FE elements. Each row of the matrix corresponds to the element number, containing eight node numbers forming a hexahedral.
	Constrained FE Nodes	**FE_node_fix** ($N_{\text{nodes-FE}}$ x 3) binary matrix of all FE nodes, 1 represents if the node is fixed for that dimension.
	Displaced FE Nodes	**FE_node_disp** ($N_{\text{nodes-FE}}$ x 3) binary matrix of all FE nodes, 1 represents if the node is displaced for that dimension.
EFG ABAQUS / LS-Dyna Output	EFG Nodes	$\mathbf{X}_{\mathbf{EFG}}$ ($N_{\text{nodes-EFG}}$ x 3) matrix of all EFG node locations. Each row of the matrix corresponds to the EFG node number.
	Constrained EFG Nodes	**EFG_node_fix** ($N_{\text{nodes-EFG}}$ x 3) binary matrix of all EFG nodes, 1 represents if the node is fixed for that dimension.
	Displaced EFG Nodes	**EFG_node_disp** ($N_{\text{nodes-EFG}}$ x 3) binary matrix of all EFG nodes, 1 represents if the node is displaced for that dimension.

Table 4.1: Data format of information read in from the ABAQUS output files.

An integration point grid for the EFG and interface region should also be read in. This is just a matrix holding the three dimensional coordinates of the integration point locations. It is ideal to have a regular grid, such that each integration point can be assigned the same volume.

4.1.3 Material Model

The material model information is to be set, requiring Young's modulus, Poisson's ratio, and the density for each material identified within the mixed-mesh. From this data the lame` material constants can be calculated as follows.

$$\lambda = E\frac{v}{(1+v)(1-2v)} \tag{4.2}$$

$$\mu = E\frac{v}{2(1+v)} \tag{4.3}$$

The material model is based upon the Neo-Hooken model (Bathe, 1996).

4.1.4 Support Domains

The construction of nodal support domains for the EFG and interface region is quite simple following on from Horton (2007). The method requires a fixed number, *n*, of nodes per support domain, which is user defined. Support domains are then constructed by finding the *n* closest nodes to each integration point. A limit on the number of Finite Element nodes allowed within a single support domain removes the possibility of an entire element being consumed by a support domain, which would have no hourglass control measures. Having a fixed number of nodes per support domain is faster and more robust than typical support domain constructions, which rely on defining a fixed local volume with a varying number of nodes.

4.1.5 Hexahedral Shape Functions

The hexahedral shape functions and derivatives are determined from a series of calculations. The matrix of hexahedral shape function natural derivatives is defined as,

$$\partial \boldsymbol{hr}^{\boldsymbol{T}} = -\frac{1}{8}\begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix} \tag{4.4}$$

using the node numbering convention as described in Bathe (1996). Each elemental Jacobian, $\boldsymbol{J}$, is then calculated based on the element nodal position vector, $\hat{\boldsymbol{x}}$.

$$\boldsymbol{J} = \partial \boldsymbol{hr}^{\boldsymbol{T}}\hat{\boldsymbol{x}} \tag{4.5}$$

Using the elemental Jacobians, hexahedral shape function derivatives, $\partial \boldsymbol{h}$ *,* are then computed by,

$$\partial \boldsymbol{h} = \partial \boldsymbol{hr}(\boldsymbol{J}^{-1})^T \tag{4.6}$$

4.1.6 Moving Least Squares Shape Functions

Calculations of the Moving Least Squares shape functions for each nodal support domain of size *n*, are derived from Fries and Matthies (2003). Consider a three dimensional space vector of monomial basis functions, $\boldsymbol{p}$, of length *m*.

$$\boldsymbol{p}(x)^T = \begin{bmatrix} 1 & x & y & z & xy & xz & yz & x^2 & y^2 & z^2 & xyz & \ldots & x^3 & y^3 & z^3 \end{bmatrix}$$

The nodal displacement approximation, $\boldsymbol{u}^h(x)$, is calculated with respect to the coefficient vector, $\boldsymbol{a}(x)$.

i.e. $$\boldsymbol{u}^{\boldsymbol{h}}(x) = \boldsymbol{p}^{\boldsymbol{T}}(x)\boldsymbol{a}(x) \tag{4.7}$$

The formulation of $\boldsymbol{a}(x)$ is determined by minimising the weighted residual function, *J*, where,

$$J = \sum_{i=1}^{n} W(d_i)(\boldsymbol{p}^{\boldsymbol{T}}(x_i)\boldsymbol{a}(x) - \boldsymbol{u}(x_i))^2 \tag{4.8}$$

In (4.8) $W(d_i)$ represents a weight function with d_i being the distance between the node, x_i, and integration point, *x*. Minimising J is done by considering

$$\frac{\partial J}{\partial a} = 0 \tag{4.9}$$

which leads to the following linear relationship,

$$\boldsymbol{A}(x)\boldsymbol{a}(x)=\boldsymbol{B}(x)\boldsymbol{U} \tag{4.10}$$

In (4.10) ***A*** is an *m* x *m* matrix known as the moment matrix defined by,

$$\boldsymbol{A}(x)=\sum_{i=1}^{n} W(d_i)\,\boldsymbol{p}(x_i)\,\boldsymbol{p}^{\boldsymbol{T}}(x_i) \tag{4.11}$$

B is an *m* x *n* matrix given by,

$$\boldsymbol{B}(x)=[W(d_1)\,\boldsymbol{p}(x_1)\quad W(d_2)\,\boldsymbol{p}(x_2)\quad W(d_3)\,\boldsymbol{p}(x_3)\quad \ldots \quad W(d_n)\,\boldsymbol{p}(x_n)] \tag{4.12}$$

and **U** is the vector of length *n* as shown,

$$\boldsymbol{U}^{\boldsymbol{T}}=[u_1 \quad u_2 \quad u_3 \quad \ldots \quad u_n] \tag{4.13}$$

By finding the inverse of ***A*** equation (4.14) can be solved,

$$a(x)=\boldsymbol{A}^{\boldsymbol{-1}}(x)\,\boldsymbol{B}(x)\,\boldsymbol{U} \tag{4.14}$$

Substituting (4.14) back into (4.7) we get the nodal displacement approximation,

$$u^h(x)=\Phi_i(x)\,\boldsymbol{U} \tag{4.15}$$

where the shape function vector ***Φ*** of length *n* at the ith node in the support domain is given by,

$$\Phi_i(x)=\sum_{j}^{m} \boldsymbol{p}_{\boldsymbol{j}}(x)(\boldsymbol{A}^{\boldsymbol{-1}}(x)\,\boldsymbol{B}(x))_{j,i} \tag{4.16}$$

The length *m* of ***p*** is user defined and should be chosen such that shape functions are all interpolated in a similar fashion in each dimension. There is a trade-off between the total number of integration points and the size of *m* due to the limitations on computational speed. Single point integration is well suited for low order interpolations, hence a lower value of *m* is chosen, while using a larger number of support domains. More support domains relieve the emphasis on stress calculations at any integration point. It has been found in Horton (2007) that setting *m = 4* and using 8 nodes per support domain (*n = 8*) is substantial enough for deformation to be transferred between support domains. In addition it has been noted that by using very small support domain sizes, the weighting of each node can be considered equal, without having a negative impact on the accuracy of the solution. This reduces the risk of generating singular matrix ***A***.

4.1.7 Mass Allocation

Initially a matrix, $\mathbf{M}_{\mathbf{FE}}$, is setup for handling the mass of all nodes within the FE domain. The mass of each node within an element for $\mathbf{M}_{\mathbf{FE}}$ is calculated using the determinant of each elemental Jacobian from (4.5) and the material density, *ρ*.

$$\Delta M_{node}=\rho\,det\,({}^{0}J) \tag{4.17}$$

The nodal contributions for all elements are then summed up to give $\mathbf{M}_{\mathbf{FE}}$.

A coupled mass matrix, $\mathbf{M}_{\mathbf{coupled}}$, is then created for allocating masses to all EFG and interface nodes involved in support domains. Each integration point is allocated a volume and consequently a mass based upon the materials density. This mass is equally divided amongst the number of nodes within the support domain,

$$\Delta M_{node}=\frac{\hat{V}^{(g)}\rho}{n} \tag{4.18}$$

where, *n* represents the number of nodes per support domain, and $V^{(g)}$ is the volume of the specific integration point, *g*. The mass of each FE node in $\mathbf{M}_{\mathbf{FE}}$ is then added to $\mathbf{M}_{\mathbf{coupled}}$, giving the entire nodal mass of the system. This is a very effective method of distributing mass throughout the EFG/interface region since nodes that appear in more support domains will receive more forces. One concern, however, is that nodes not included in many support domains will have a low mass, which can result in unbalanced forces and high accelerations. This is undesired, often leading to unstable simulations. It can be avoided by involving each node in at least two or three support domains as suggested in Horton (2007). A further measure is implemented so that any node that manages to escape support domain allocation is removed to prevent massless nodes entering the analysis.

4.2 Analysis Solver

The analysis solver is the most computationally intensive phase of the simulator. It consists of the main time loop described in the Total Lagrangian Explicit Dynamics Algorithm with a few additional considerations. Efficient programming is very important to minimise the number of calculations required in the main time loop, substantially increasing the performance of the algorithm.

4.2.1 Main Time Loop

Both the Finite Element and Element Free Galerkin methods follow the same calculations for the main time loop of the TLED algorithm, making it quite easy to implement the coupling as treating the entire domain as a single method.

Three displacement matrices representing, ${}^{t-\Delta t}_{0}\boldsymbol{u}$, ${}^{t}_{0}\boldsymbol{u}$, ${}^{t+\Delta t}_{0}\boldsymbol{u}$, and a global nodal force vector, ${}^{t}\boldsymbol{F}$, were defined. Psuedocode below presents the implementation of the main time loop.

```
Begin Time Loop ($t = \Delta t$)
–   Update Displacements                                  ${}^{t-\Delta t}_{0}\boldsymbol{u}={}^{t}_{0}\boldsymbol{u}$
                                                          ${}^{t}_{0}\boldsymbol{u}={}^{t+\Delta t}_{0}\boldsymbol{u}$
–   Reset Global  Nodal Force Vector                      ${}^{t}\boldsymbol{F}=0$
–   Loop Over All Elements & Support Domains
        –   Compute elemental/support domain displacement derivatives
                                                          $\boldsymbol{u}^{(k)}_{i,j}=\partial\,\boldsymbol{h}\,{}^{t}_{0}\boldsymbol{u}$
                                                          $\boldsymbol{u}^{(g)}_{i,j}=\boldsymbol{\Phi}'\,{}^{t}_{0}\boldsymbol{u}$
        –   Calculate deformation gradient                ${}^{t}_{0}\boldsymbol{X}=\boldsymbol{u}_{i,j}+\boldsymbol{I}_{\mathbf{3x3}}$
        –   Inverse Right Cauchy-Green deformation tensor
                                                          $\boldsymbol{C}_{ij}=[{}^{t}_{0}\boldsymbol{X}^{T}\,{}^{t}_{0}\boldsymbol{X}]^{-1}$
        –   Jacobian Determinant                          $J=det({}^{t}_{0}\boldsymbol{X})$
        –   Second Piola-Kirchoff Stress
                                                          $S_{ij}=\mu(I-C_{ij})+\lambda J(J-1)C_{ij}$
                                                          ${}^{t}_{0}\boldsymbol{S}=\begin{bmatrix} {}^{t}_{0}S_{11} & {}^{t}_{0}S_{12} & {}^{t}_{0}S_{31} \\ {}^{t}_{0}S_{12} & {}^{t}_{0}S_{22} & {}^{t}_{0}S_{23} \\ {}^{t}_{0}S_{31} & {}^{t}_{0}S_{23} & {}^{t}_{0}S_{33} \end{bmatrix}$
        –   Integrate to get Elemental/Support domain nodal force
                                                          ${}^{t}\hat{\boldsymbol{F}}^{(k)}_{i}=8J\,{}^{t}_{0}\boldsymbol{X}\,{}^{t}_{0}\boldsymbol{S}\,\partial\,\boldsymbol{h}^{(k)}$
                                                          ${}^{t}\hat{\boldsymbol{F}}^{(g)}_{i}=V^{(g)}\,{}^{t}_{0}\boldsymbol{X}\,{}^{t}_{0}\boldsymbol{S}\,\boldsymbol{\Phi}'^{(g)}$
        –   Update Global Force                           ${}^{t}\boldsymbol{F}^{(total)}_{i}={}^{t}\boldsymbol{F}^{(total)}_{i}+{}^{t}\hat{\boldsymbol{F}}^{(k)}_{i}+{}^{t}\hat{\boldsymbol{F}}^{(g)}_{i}$

–   End Loop Over Elements/Support Domains
–   Use Central Difference Method to Calculate Displacements
```

$$ {}^{t+\Delta t}_{0}\boldsymbol{u} = \frac{-\Delta t^{2}}{\boldsymbol{M}} {}^{t}\boldsymbol{F} + 2\,{}^{t}_{0}\boldsymbol{u} - {}^{t-\Delta t}_{0}\boldsymbol{u} $$

– Loop Over Constrained Nodes $\quad {}^{t+\Delta t}_{0}\boldsymbol{u}_{i} = 0$

– Loop Over Displaced Nodes $\quad {}^{t+\Delta t}_{0}\boldsymbol{u}_{i} = d(\Delta t)$

End Time Loop (*t* = *T*)

4.2.2 Hourglass Control

One of the biggest disadvantages to using single-point integration for hexahedral elements is the requirement for controlling zero energy modes, known as hour-glassing (Hallquist, 2006). In order to control hour-glassing within the Finite Element domain, resistance providing artificial stiffness is implemented, which has a negligible effect on stable global modes. This is an efficient method following on from modifications of Flanagan and Belytschko (1984) perturbation method (Joldes, Wittek, *et al.,* 2007). An additional hourglass control force is added to the total force of the system, based on the hourglass resistance and displacement. Hourglass base matrix is setup as:

$$ \boldsymbol{H}^{T} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{4.19} $$

The k^th^ elemental hourglass control force, ${}^{t}\boldsymbol{F}^{(k)}_{hg}$, can then be calculated by the following series of equations:

$$ \partial \boldsymbol{h}^{(k)}{}_{hg} = \boldsymbol{H} - \partial \boldsymbol{h}^{(k)} [x^{(k)}]^{T} \boldsymbol{H} \tag{4.20} $$

$$ \boldsymbol{u}_{ij\,hg} = {}^{t}_{0}u_{i,\,jhg} = [\partial \boldsymbol{h}^{(k)}{}_{hg}]^{T}\, {}^{t}_{0}u^{(k)}_{i} \tag{4.21} $$

$$ {}^{t}\boldsymbol{F}^{(k)}_{hg} = \frac{R_{hg}(\lambda + 2\mu)\, Vol^{(k)} [\partial \boldsymbol{h}^{(k)}]' \partial \boldsymbol{h}^{(k)}_{hg} \boldsymbol{u}_{ij\,hg}}{8} \tag{4.22} $$

where $\partial \boldsymbol{h}^{(k)}{}_{hg}$ is the k^th^ elemental hourglass shape function derivative, $\boldsymbol{u}_{ij\,hg}$ represents the elemental hourglass displacement derivatives, and R_{hg} is the hourglass resistance constant. The hourglass force is then added to the elemental force calculated without hourglass control as shown in (4.23).

$$ {}^{t}F^{(k)} = {}^{t}F^{(k)} + {}^{t}F^{(k)}_{hg} \tag{4.23} $$

A good value for the hourglass resistance, R_{hg}, was found to be $R_{hg} = 0.04/9$.

5. Coupling Considerations

5.1 Mixed-Mesh Generation

One of the main considerations of utilising coupled FE/EFG methods is the generation of mixed-meshes. Meshless methods may have much more freedom with node placement in comparison to hexahedral FE methods, although very regular node placements can lead to singular shape functions within one plane (Horton, 2007). Regular node and integration arrangements for meshless methods have been successful, however this becomes equivalent to a hexahedral mesh, suffering from hour-glassing and furthermore is quite complicated to generate over irregular domains, such as the brain (Li, and Liu, 2004). A way around this is to model the meshless region using a well refined tetrahedral mesh, which helps to maintain a roughly even density, while not conforming to very regular node placements. The tetrahedral element information is omitted, since meshless methods do not require the connectivity of nodes. It is possible to convert sections of a hexahedral mesh into tetrahedrals in commercial meshing software such as Hypermesh, which easily gives rise to the generation of mixed-meshes. Unfortunately this procedure relies on the existence of a Finite Element mesh, which, depending on the complexity of the shape, may still be quite complicated and time consuming to create.

Another aspect of mixed-mesh generation is the requirement for an integration point domain. As found in Horton (2006), a background grid of integration points for the EFG domain leads to greater accuracy and stability, while still performing efficiently. The background grid can be extended up to the boundary between the EFG and FE region in order to maximise coupling as mentioned in Chapter 5.2.

5.2 Coupling Integration Point Distribution

An investigation into the depth of coupling and its effects on the accuracy of the method was conducted. The level of coupling is indicated by the number of FE nodes involved in interface support domains. This is strongly dependent on the distribution of integration points around the boundaries of each region. The best results occur when the coupling integration point grid is mapped up to the boundary of the FE region. This maximises the number of FE nodes within interface support domain.

To avoid unnecessarily using too many integration points around the coupling region the findings of Horton (2007) should be employed with respect to interface support domains. That is, by using twice as many interface support domains to interface nodes, the Moving Least Squares approximation will still give very accurate results.

5.3 Interface Support Domain Composition

An area of interest in the coupling method is the number of allowable FE nodes to be considered within a single interface support domain. A support domain should not be allowed to contain only Finite Element nodes, as there is the possibility that it has coincided with an entire element, which would not have any hourglass control measures. For this reason a limit on the maximum number of FE nodes within an interface support domain must be defined, based on the number of nodes within a hexahedral element. This condition would then be checked when allocating support domains within the preprocessor.

It was found that by reducing the amount of allowable FE nodes per interface support domain causes the level of coupling to decrease, which produces results with poorer accuracy. Allowing up to 7 FE nodes per interface support domain, one node less than a hexahedral element, maximises the coupling, consequently leading to increased accuracy of solutions.

6. Validation & Results

6.1 Validation

The coupling method was validated by a series of quasi-static deformation tests, using the material properties of healthy brain tissue. Initially the algorithm was trialled and compared against a Finite Element solution using commercial software (ABAQUS) for a homogeneous cylinder undergoing compression, extension, and shear deformations. The mixed-mesh contained an outer FE region with an inner EFG core. The results were further compared against validated Element Free Galerkin software, showing that the coupling method performs slightly better than a solely EFG method and is still very close to the FE solution. Comparisons against a FE solution is preferable over an analytical solution due to the complicated mathematical nature of analytical methods and general loss of accuracy for non-linear problems. The results shown in *Table 6.1* reflect the maximum nodal displacement error in comparison to the FE ABAQUS solution.

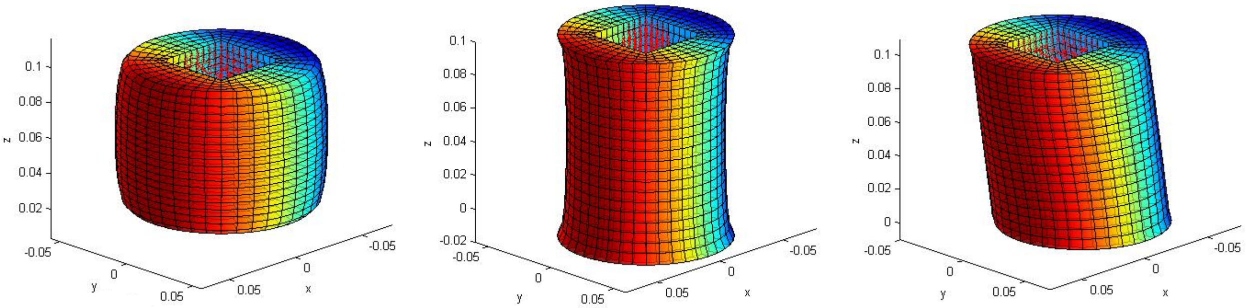

Figure 6.1: Final deformed mixed mesh cylinders. Compression (left), Extension (middle), Shear (right).

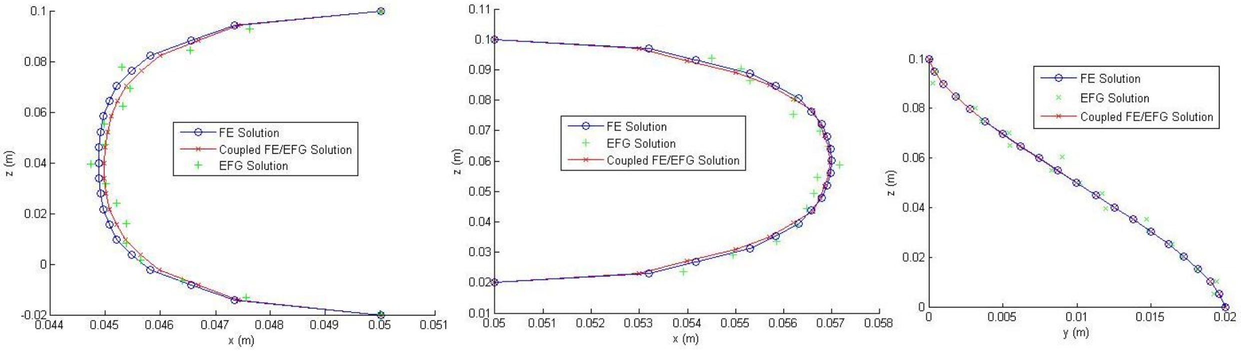

Figure 6.2: Coupled cylinder edge deformation comparisons against FE ABAQUS solution and validated EFG model undergoing extension (left), compression (middle), and shear (right).

The final validation test involved a partially constrained ellipsoid undergoing indentation on the surface. The mixed mesh gave highly accurate results in comparison with a FE mesh simulated in ABAQUS.

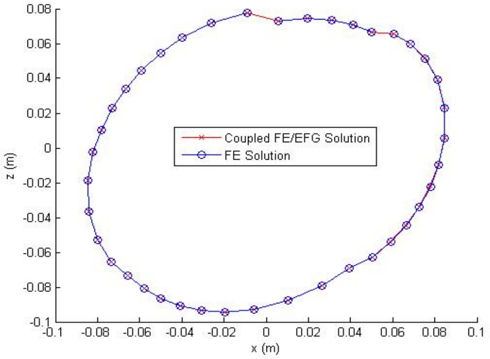

Figure 6.3: Cross-section comparison of deformed boundary for indented ellipsoid.

Deformation Model	**Maximum Error (*mm*)**		
	Δx	***Δy***	***Δz***
Cylinder Extension	0.1744	0.0055	0.0694
Cylinder Compression	0.1058	0.0177	0.0672
Cylinder Shear	0.0193	0.0273	0.0097
Ellipsoid Indentation	0.1628	0.0734	0.1055

Table 6.1: Maximum displacement errors in coupling method compared against FE ABAQUS solutions.

It is evident from *Table 6.1, Figure 6.2,* and *Figure 6.3* that the maximum error in all cases falls within the allowable 0.85 *mm* tolerance for surgical accuracy (Bourgeois, Magnin, *et al.*, 1999).

An additional investigation has been conducted comparing tumour growth on an ellipsoid using a stand alone FE mesh and a mixed-mesh, containing an EFG region of high density surrounding the proposed area of localised high deformation. The accuracy of the FE results became questionable as tumours grew larger than 523.6 mm^3. *Figure 6.4* demonstrates the localised deformation to the FE mesh with a series of increasing tumour growths. The onset of hour-glassing is present during the very early stages of tumour growth, despite control measures in place to prevent this. *Table 6.2* compares the maximum nodal displacements, surrounding the localised high deformation region, using the stand alone FE mesh and the mixed-mesh. It is apparent that initially both methods give quite similar results, however they begin to differ significantly as the distortion to the FE mesh increases. Discrepancies between the two methods are observed for tumour growths greater than 523.6 *mm^3*.

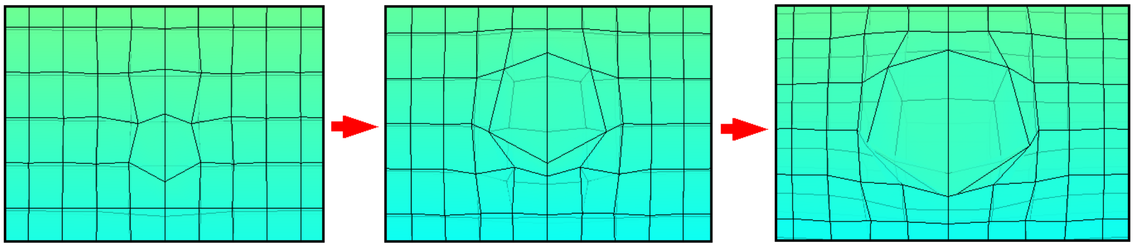

Figure 6.4: Ellipsoid FE mesh tumour progression – 268.1 mm^3, 523.6 mm^3, and 817.3 mm^3 respectively.

	Maximum Difference *(mm)*		
Tumour Volume *(mm^3)*	***Δx***	***Δy***	***Δz***
261.8	0.0076	0.0082	0.0061
523.6	0.0188	0.0007	0.0112
817.3	0.0273	0.0133	0.0617

Table 6.2.: Maximum displacement differences for tumour growths.

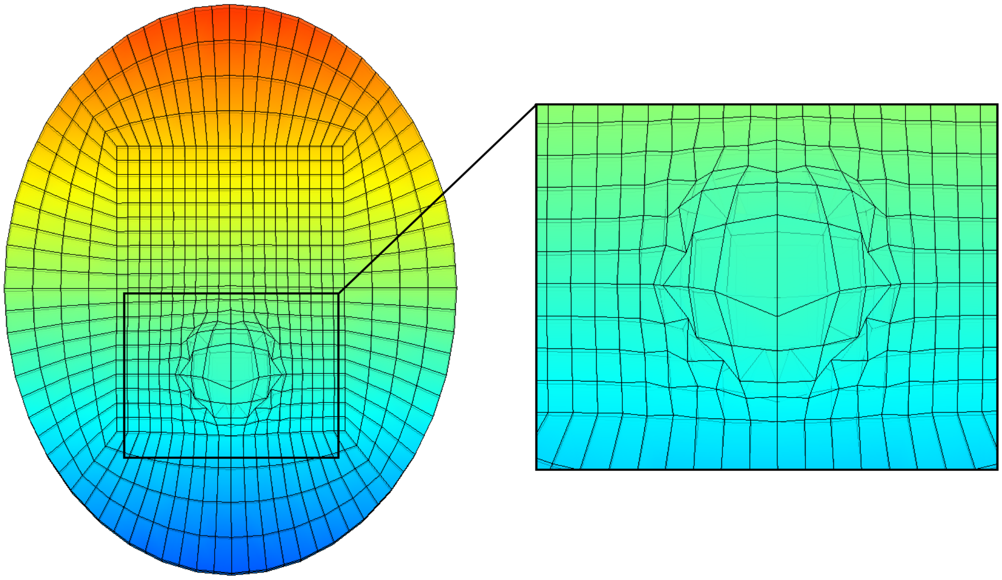

Figure 6.5: Failed 3053.6 mm^3 tumour growth on FE ellipsoid mesh.

The FE mesh fails as the tumour reaches a volume of 3053.6 *mm^3*. Significant distortion to the FE mesh is present in *Figure 6.5*, with the hexahedral elements compressing up to 70%, well beyond the reliable limits as discussed in Wittek, Dutta-Roy, *et al.* (2008). The mixed-mesh deformation for the same tumour growth volume is shown in *Figure 6.6.*

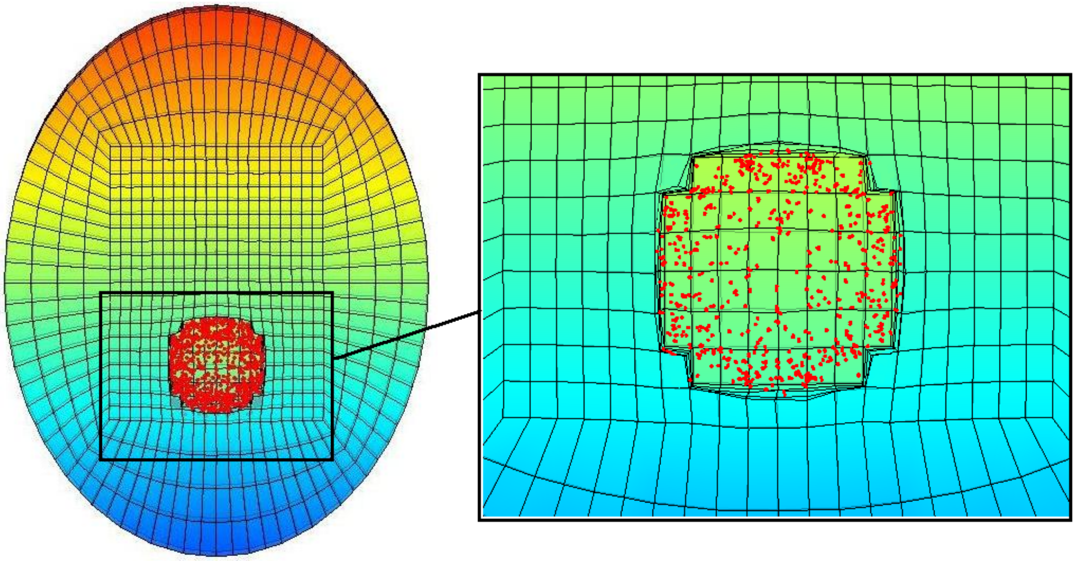

Figure 6.6: Final deformation due to 3053.6 mm^3 tumour using coupled method.

6.2 Tumour Growth Analysis

A mixed-mesh of a brain was created allowing for tumour growth to occur behind the left ventricle, mimicking an MRI scan of a tumour affected brain in Urbach, Binder, *et al.* (2007). An EFG nodal domain of high density surrounds the proposed tumour region allowing for large deformation. The tumour was grown as an ideal sphere, of which the analytical equations are well defined.

For healthy brain tissue and tumour we assume that Young's modulus, *E*, is 3000 Pa, and Poisson's ratio, *v*, is 0.49 (Miller, Chinzei, *et al.*, 2000, Miller, 2002). The ventricles contains cerebro-spinal fluid (CSF), which has very similar material properties to water, hence they are modelled as a soft elastic compressible solid, with *E* = 10 Pa and a low Poisson's ratio, *v* = 0.1. A low Poisson's ratio allows to simulate leakage of the cerebro-spinal fluid which may occur under static deformation conditions. (Wittek, Miller, *et al.*, 2006).

A number of different tumour growth sizes were investigated, with deformation volume change in the ventricles, from an initial volume of 57.1 *ml,* noted in *Table 6.3.*

V_{Tumour} (*mm^3*)	113.1	523.6	1436.8	2144.6	4118.8
$\Delta V_{Left\text{-}Ventricle}$ (*ml*)	-0.099	-0.524	-1.224	-1.312	-1.968
$\Delta V_{Right\text{-}Ventricle}$ (*ml*)	-0.075	-0.331	-1.099	-1.143	-1.310
$\Delta V_{Ventricles}$ (*ml*)	-0.174	-0.855	-2.323	-2.455	-3.278

Table 6.3: Change in left ($\Delta V_{Left\text{-}Ventricle}$), right ($\Delta V_{Right\text{-}Ventricle}$), and total ventricular volume ($\Delta V_{Ventricles}$) due to tumour volume (V_{tumour}).

Given the location of the tumour, the ventricular deformation and associated volume loss, displayed in *Table 6.3*, is likely to correspond to the leakage of CSF. In reality CSF may leak between the left and right ventricles, however it is also known to leak out of the ventricles completely, particularly under static deformation (Rando and Fishman, 1992). Furthermore, large tissue deformation is apparent, particularly for the 4118 *mm*^{3} tumour, as shown in *Figure 6.7,* with local displacements of up to 9.66 *mm.* For this example 14.5% of the brain mesh experienced displacements greater than 5 *mm*.

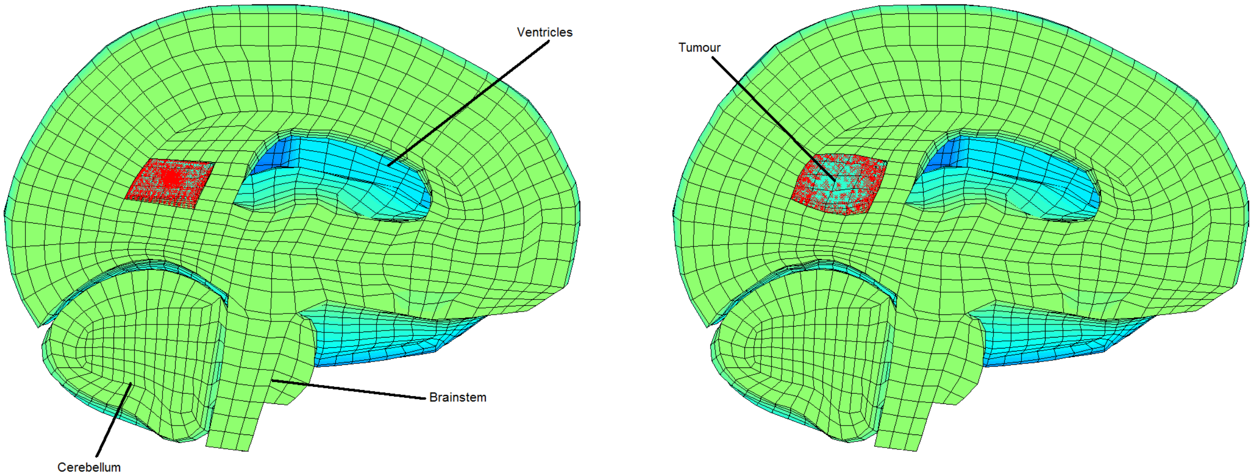

Figure 6.7: Undeformed brain cross-section (left). Deformed brain cross-section due to 4118 mm^3 tumour (right).

This follows on from Clatz, Bondiau, *et al.* (2004), which declared volume variation within the ventricles and large tissue deformation in response to brain tumour growth mass effect. It should be noted that simulated tumour growths larger than 14000 *mm*^{3} on the given mixed-mesh reduced the reliability of the method as the resulting deformation to the Finite Element region became too large. A greater EFG domain would be required surrounding the tumour affected region in order to simulate larger growths accurately.

7. Conclusion

A new coupling method has been proposed to combine the Finite Element and Element Free Galerkin methods for modelling the non-linear soft tissue deformation of the brain in response to tumour growth. The method was verified against FE commercial software and a validated EFG simulator on a number of different mixed meshes. All results were very accurate, easily falling within the 0.85 *mm* error tolerance, corresponding to the working resolution of an MRI scan. Simple analytical tumour growths were conducted on a comprehensive brain mesh. The tumour's close proximity to the ventricles caused observable volume changes, which may involve leakage of CSF. Furthermore large tissue displacements were noted, with a significant portion of the brain undergoing moderate deformation. In reality this may have a detrimental effect on the cell metabolism and function of the brain, altering the stress distribution and blood flow. Further investigation into realistic tumour growth models and implementation of a brain-skull contact algorithm would increase the reliability of the results. Ultimately this would become beneficial for both clinical prognosis and operation planning as well as for simulated training applications.

References

Bathe, K. J. (1996). Finite Element Procedures. Upper Saddle River, New Jersy, USA, Prentice-Hall Inc.

Belytschko, T., Krongauz, Y., *et al.* (1996). "Meshless Methods: An Overview and Recent Developments." Computer Methods in Applied Mechanics and Engineering 139: 3-47.

Bourgeois, G., Magnin, M., *et al.* (1999). "Accuracy of MRI-guided stereotactic thalamic functional neurosurgery." NEURORADIOLOGY 41(9): 636-645.

Clatz, O., Bondiau, P., *et al.* (2004). Brain Tumour Growth Simulation, INRIA.

Flanagan, D. P. and T. Belytschko (1984). "Eigenvalues and Stable Times Steps for the Uniform Strain Hexahedron and Quadrilateral." Journal of Applied Mechanics 51: 35-40.

Fries, T. P. and H. G. Matthies (2003). Classification and Overview of Meshfree Methods. Informatikbericht 2003-03. Technical University Braunschweig, Brunswick, Germany, Institute of Scientific Computing.

Hallquist, J. O. (2006). LS-DYNA Theory Manual. California, Livermore Software Technology Corporation.

Horton, 2006. "Mesh Free Methods For Soft Tissue Deformation", Report # ISML/02/2006.

Horton, 2007. "Mesh Free Methods For Soft Tissue Deformation II", Report # ISML/01/2007.

Horton, Wittek, *et al.*, (2007) "Subject-specific biomechanical simulation of brain indentation using a meshless method" MICCAI 2007.

Joldes, G. R., A. Wittek, *et al.* (2007). "An efficient hourglass control implementation for the uniform strain hexahedron using the Total Lagrangian formulation." Communications in Numerical Methods in Engineering. Early View.

Li, S. and W. K. Liu (2004). Meshfree Particle Methods. Berkeley, California, Springer.

Miller, K., Chinzei, K., *et al.* (2000). "Mechanical properties of brain tissue in-vivo: experiment and computer simulation." Journal of Biomechanics 33(11): 1369-1376.

Miller, K., Joldes, G., *et al.* (2007). "Total Lagrangian Explicit Dynamics Finite Element Algorithm for Computing Soft Tissue Deformation." Communications in Numerical Methods in Engineering 23: 121-134.

Miller, K., Taylor, Z., *et al.* (2005). "Towards Computing Brain Deformations for Diagnosis, Prognosis, and Neurosurgical Simulation." Journal of mechanics in medicine and biology 5(1): 105.

Rando, T. A. and Fishman, R. (1992). "Spontaneous intracranial hypotension " Neurology 42: 481.

Urbach, H., Binder, D., *et al.* (2007). "Correlation of MRI and histopathology in epileptogenic parietal and occipital lobe lesions." Seizure 16(1): 608-614.

Wittek, A., Dutta-Roy, T., *et al.*, (2008). "Subject-specific non-linear biomechanical model of needle insertion into brain" Computer Methods in Biomechanics and Biomedical Engineering 11: 135-146.

Wittek, A., Miller, K., *et al*. (2006). "Patient-specific model of brain deformation: Application to medical image registration." Journal of Biomechanics 40: 919-929.