Embedded finite elements for modeling axonal injury

Harsha T. Garimella[1,3,+], Ritika R. Menghani[1,+], Jesse I. Gerber[1], Srikumar Sridhar[4], Reuben H. Kraft[1,2,*]

[1] Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, University Park, USA 16801

[2] Department of Biomedical Engineering, The Pennsylvania State University, University Park, USA 16801

[3] Present address: Computational Medicine and Biology Division, CFD Research Corporation, 701 McMillian Way NW, Huntsville, AL USA 35806

[4] Department of Computer Science and Engineering, The Pennsylvania State University, University Park, USA 16801

[+] Co-first authors

[*]Corresponding Author, Email: reuben.kraft@psu.edu

1 Abstract

The purpose of this paper is to propose and develop a large strain embedded finite element formulation that can be used to explicitly model axonal fiber bundle tractography from diffusion tensor imaging of the brain. Once incorporated, the fibers offer the capability to monitor tract-level strains that give insight into the biomechanics of brain injury. We show that one commercial software has a volume and mass redundancy issue when including embedded axonal fiber and that a newly developed algorithm is able to correct this discrepancy. We provide a validation analysis for stress and energy to demonstrate the method.

Keywords:
embedded element, volume redundancy, mass redundancy, force redundancy, finite element, brain tissue anisotropy

2 Introduction

The brain consists of a complex network of axonal fibers that can be observed through the use of magnetic resonance diffusion tensor imaging (MR-DTI) [1]–[3]. These fibers can be considered the mesoscopic level of the brain [4], smaller than the organ and larger than the individual cells. **Figure 1** illustrates the white matter fiber tracks in the brain of a healthy subject. The physics and mechanics of axonal fiber tracts is important in various fields including understanding axonal injury [5], [6], predicting the source of electrical signals [7], [8], and learning about brain structure-function relationships [9]–[11]. Recently, we have been using the embedded element approach to understand and predict axonal injury due to impact or blast loading to the head [12]–[15].

The embedded element method, a mesh superposition technique, is a method that allows for an explicit inclusion of the axonal fiber network into finite element models. This technique facilitates the inclusion of multiple fibers per finite element. Here, an independent axonal fiber mesh is developed and coupled to the conventional finite element head model using the embedded element constraint. When the two meshes are coupled together, the embedded mesh loses its degrees of freedom and adopts the degrees of freedom of the host mesh. This leads to an affine projection of strains but can be non-affine if slip is introduced[16]. Besides this, the host material is also reinforced by the stiffness of the embedded mesh – thus incorporating the effects of directional stiffening into the finite element model response. In summary, this method enables us to:

- incorporate an explicit model of the diffusion tractography in the finite element head model,
- real time tracking of the mechanical deformation of the anatomically significant axonal fiber tracts,
- automatic stiffening of brain tissue based on the fiber direction, and
- incorporation of exact fiber directions without any averaging procedures.

Demonstrating the biomechanical application of this technique, Garimella et al. [12] developed a high-resolution FE head model to investigate the evolution of axonal injury. This technique is different from the approaches that use transverse isotropic brain tissue constitutive models in the sense that modifications are made at the element level, whereas in the phenomenological models, modifications are made at the material (constitutive) model level[17]. However, one of the major limitations of using this method is the issue of material redundancy. This results in an overestimation of mass and stiffness in the model resulting in a potential underestimation of strains – thus underestimating the possibility of axonal injury.

Over the years, there have been different studies addressing the issue of volume redundancy in mesh superposition methods. For example, Fish [18], Fish and Belytschko [19] introduced the S version of this method and extended this for large deformation problem by tackling the excess stiffness in the model using an averaged integration scheme (area fraction). They assumed that the location of the embedded domain is unimportant and unknown. Subsequently, domain superposition techniques (DST) were introduced by Jiang et al. [20], Ohyama et al. [21] and Tabatabaei et al. [22], [23] where the excess stiffness in the model is addressed at a material level i.e., stiffness of the embedded domain was modified to account for the excess stiffness. A subsequent post-processing algorithm was introduced into the methodology to account for the stress corrections arising due to the use of modified material model. Then there was the M^3 method introduced by Zako et al. [24] who defined a stress-correlation matrix in the overlapped regions to accommodate the excessive stiffness; and the Independent Mesh Method (IMM) introduced by Iarve et al. [25] where the shape functions in the overlapped regions were ignored. Most of these studies have used meshes (host and embedded) of the same dimension while implementing this technique. Also, in most of these studies, size of the embedded mesh is comparable with host mesh making

it possible to neglect the quadrature points (thus shape functions) occupied by the embedded regions. But, in our case, we are trying to model the fibers as truss elements embedded into three-dimensional hexahedral elements, and the embedded fibers are smaller than the host finite elements. Besides this, most of the formulations were shown using stiffness matrices making it mandatory to use a linearized finite element formulation.

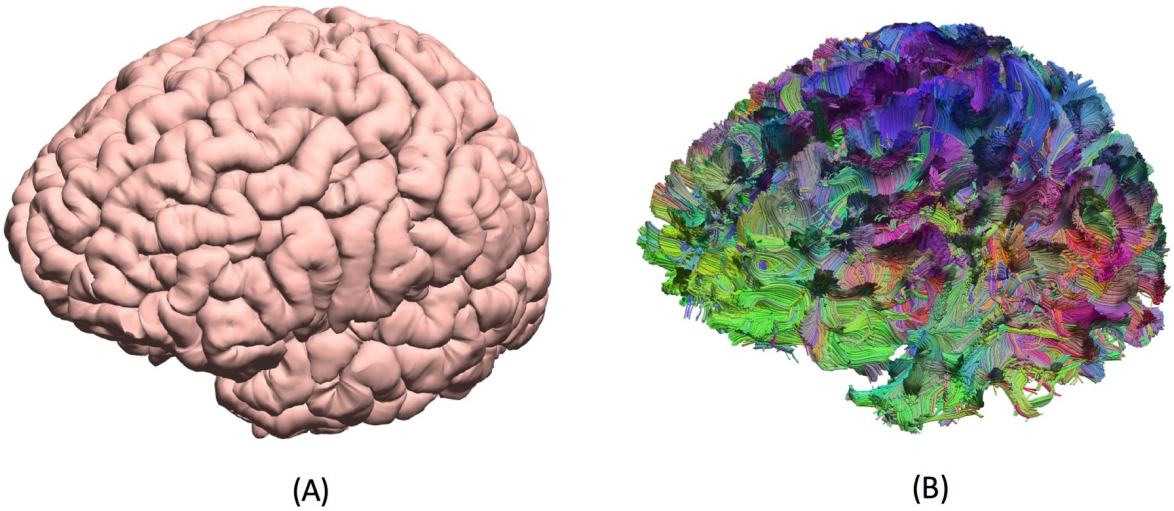

Figure 1. (A) Rendering of brain cortical surface; (B) rendering of white mater fiber tracks. Both images were created using the Cortical Surface Extraction and Diffusion Pipeline toolkits in BrainSuite [26].

Therefore, in this paper, we propose to develop the novel force vector based finite element formulation (eliminating the need for linearization), corresponding finite element library and validate the formulation. The novelty of this paper lies in (a) its ability to handle embedded elements that are smaller than a single host element (similar to that of Fish and Belytschko's [19] 'sub-h' method), (b) its capability of accurately handling the location and orientation of the embedded elements (using the inverse mapping idea developed by Elwi et al. [27]), (c) its ability in handling volume redundancy without having to modify material properties of the embedded domain or add additional integration points. This formulation proposed here was developed in a large strain domain such that it can be used for various biomechanical simulations (e.g., high-resolution human head finite element models with embedded axonal fiber tracts). This paper provides a detailed finite element algorithm and provides access to an open-source finite element library (under development) with the implementation of these algorithms.

The layout of this paper is as follows. The "Methods" section starts with the basic finite element formulation of a solid mechanics problem and subsequently presents the mathematics of the embedded element method in a large strain (finite kinematics) domain. This is followed by the "Results" section where we show the effects of volume redundancy on the model response, propose a new finite element algorithm to address volume redundancy and show the results before and after. This paper concludes with a discussion on our results and our future vision for this finite element library. Some of the implementation detail is included in the Appendix section of the paper.

3 Methods

The mathematical formulation of the embedded finite element can be developed from the very fundamental momentum equation (strong form of a mechanics problem). The strong form consists of the momentum balance equation, boundary conditions and the continuity conditions:

Momentum Balance:

$$\nabla_0 \cdot \mathbf{P} + \rho_0 \mathbf{b} = \rho_0 \ddot{\mathrm{u}} \tag{1}$$

Boundary Conditions:

$$\mathbf{e}_i \cdot \mathbf{n}^0 \cdot \mathbf{P} = \mathbf{e}_i \cdot \bar{\mathbf{t}}_0 \text{ on } \Gamma^0_{t_i} \tag{2}$$

$$u = \bar{u}_i \text{ on } \Gamma^0_{u_i} \tag{3}$$

$$\Gamma^0_{t_i} \cup \Gamma^0_{u_i} = \Gamma^0 \text{ for } i = 1 \text{ to } n_{SD} \tag{4}$$

$$\Gamma^0_{t_i} \cap \Gamma^0_{u_i} = 0 \text{ for } i = 1 \text{ to } n_{SD} \tag{5}$$

Interior Continuity Conditions:

$$\left\| n^0_j \mathbf{P}_{ji} \right\| = 0 \text{ on } \Gamma^0_{int} \tag{6}$$

Here ρ_0 is the material density, $\mathbf{P}$ represents the 1st Piola-Kirchhoff Stress tensor, $\mathbf{b}$ is the body force vector, $\mathbf{u}$ is the displacement vector, t_i is the traction. $\Gamma^0_{t_i}$ is the set of boundaries on which a traction boundary condition was applied while $\Gamma^0_{u_i}$ is the set of boundaries with prescribed displacement boundary conditions. All the above equations constitute the strong form of a mechanics-based finite element problem. Most of the preliminary mathematical governing equations of solid mechanics shown in the following sections are extracted from Belytschko et al. [28]. Even though these equations are standard, they are presented here for reader's convenience as they would lay the foundation for the mathematics of the embedded elements.

The corresponding weak form is derived using the above strong form and the following trial and test function spaces, $\delta \mathrm{u}(\mathrm{X}) \epsilon u_0 \; and \; \mathrm{u}(\mathrm{X}, \mathrm{t}) \epsilon u$, respectively, where u is the space of kinematically admissible displacements and u_0 is the same space with the additional requirement that the displacements vanish on displacement boundaries. Substituting these expressions into the strong form and integrating over the domain, we obtain:

$$\int_{\Omega_0} (\delta \mathbf{F}^T : \mathbf{P} - \rho_0 \delta \mathrm{u} \cdot \mathbf{b} + \rho_0 \delta \mathrm{u} \cdot \ddot{\mathrm{u}}) d\Omega_0 - \sum_1^{n_{SD}} \int_{\Gamma^0_{t_i}} (\delta \mathrm{u} \cdot \mathbf{e}_i)(\mathbf{e}_i \cdot \mathbf{t}_i^{-0}) d\Gamma^0 = 0 \tag{7}$$

where $\mathbf{F}$ is the deformation gradient, and Γ^0 represents the boundaries of the domain (on which force and displacement boundary conditions were applied). The above weak form can be rewritten using the principle of virtual work as shown below. If $u \in U$

$$\delta \mathbf{W}^{int}(\delta \mathrm{u}, \mathrm{u}) - \delta \mathbf{W}^{ext}(\delta \mathrm{u}, \mathrm{u}) + \delta \mathbf{W}^{kin}(\delta \mathrm{u}, \mathrm{u}) = 0 \; \forall \; \delta \mathrm{u} \in u_0 \tag{8}$$

where

$$\delta \mathbf{W}^{int} = \int_{\Omega_0} \delta \mathbf{F}^T : \mathbf{P} d\Omega_0 \tag{9}$$

$$\delta \mathbf{W}^{ext} = \int_{\Omega_0} \rho_0 \delta \mathrm{u} \cdot \mathbf{b} \, d\Omega_0 + \sum_1^{n_{SD}} \int_{\Gamma^0_{t_i}} (\delta \mathrm{u} \cdot \mathbf{e}_i)(\mathbf{e}_i \cdot \mathbf{t}_i^{-0}) d\Gamma^0 \tag{10}$$

$$\delta \mathbf{W}^{kin} = \int_{\Omega_0} \rho_0 \delta \mathrm{u} \cdot \ddot{\mathrm{u}} \, d\Omega_0 \tag{11}$$

All the terms in the above equation are virtual energies where $\delta \mathbf{W}^{kin}$ is the virtual energy term originating from kinetic forces, $\delta \mathbf{W}^{int}$ is the virtual energy term originating from internal forces (stresses) and $\delta \mathbf{W}^{ext}$ is the virtual energy term originating from external body forces and traction. Once the finite element mesh is created, the above energy terms can be further modified into a vector form as follows:

$$\delta \mathbf{W}^{int} = \delta \mathrm{u}^T \mathbf{f}^{int} \tag{12}$$

$$\delta \mathbf{W}^{ext} = \delta \mathrm{u}^T \mathbf{f}^{ext} \tag{13}$$

$$\delta \mathbf{W}^{kin} = \delta \mathrm{u}^T \mathbf{f}^{kin} \tag{14}$$

where $\delta u^T = [\delta u_1\ \delta u_2\ \delta u_3 \ldots \delta u_n]$. Here, δu represents the nodal displacement vector, n is the number of nodes, $\mathbf{f}^{kin}$ is the kinetic/inertial nodal force vector, $\mathbf{f}^{int}$ is the internal nodal force vector and $\mathbf{f}^{ext}$ is the external nodal force vector. Therefore, the final discretized equation can be written as follows:

$$\mathbf{f}^{kin} = \text{M}\ddot{\text{u}} = \mathbf{f}^{ext} - \mathbf{f}^{int} \tag{15}$$

3.1 Mathematics of the embedded element methods

In this technique, we have two domains - matrix and fiber. Due to the overlap between these domains there is an additional mass and internal nodal force (stiffness) in the system.

3.1.1 Mass

The overall mass (system mass) is the addition of the mass of the host domain (matrix) and mass of the embedded fibers (fiber). This calculation is carried out at the element level, and the modified element mass will be scattered to form a global modified mass matrix.

$$\mathbf{m}_{e,effective} = \mathbf{m}_{e,matrix} + \mathbf{m}_{e,fiber} \tag{16}$$

where $\mathbf{m}_{e,effective}$ is the effective element mass, $\mathbf{m}_{e,matrix}$ is the matrix element mass and $\mathbf{m}_{e,fiber}$ is the total mass of the embedded fibers.

3.1.2 What is causing excess mass in the system?

In the above equations, $\mathbf{m}_{e,matrix}$ is the mass matrix computed over the matrix domain. When the embedded elements are smaller than the host elements (such as trusses embedded in a hexahedral element), accounting for the exact matrix volume is not possible. Therefore, a new algorithm, to account for the exact volume of the matrix (and corresponding mass), is required.

3.1.3 Internal Nodal Forces (stiffness)

The internal nodal force contribution from the embedded fibers is added to the internal nodal force vector of the matrix resulting in a global internal nodal force vector.

$$\mathbf{f}_e^{int} = \mathbf{f}_{e,matrix}^{int} + \mathbf{f}_{e,fiber}^{int} \tag{17}$$

where $\mathbf{f}_{e,matrix}^{int}$ is the internal nodal force vector contribution from the matrix domain, $\mathbf{f}_{e,fiber}^{int}$ is the internal nodal force vector contribution from the fiber domain and $\mathbf{f}_e^{int}$ is the effective nodal force contribution. The above equation shows that the net nodal force is a combination of the nodal forces from the matrix domain and nodal forces from the fiber domain. Assuming no slip between the matrix and fiber domains, terms in the above equation can be re-written as,

$$\mathbf{f}_{e,matrix}^{int} = \int_{matrix} \mathbf{B}^T \boldsymbol{\sigma}\, d\Omega_0 \tag{18}$$

$$\mathbf{f}_{e,fiber}^{int} = \int_{fiber} \mathbf{B}^T \boldsymbol{\sigma}\, d\Omega_f \tag{19}$$

where **B** is the strain-displacement matrix, and $\boldsymbol{\sigma}$ is the Cauchy stress tensor.

3.1.4 What is causing excess force in the system?

In the above equations, $\mathbf{f}_{e,matrix}^{int}$ is the internal nodal force vector computed over the matrix domain. However, when the embedded elements are smaller than the host elements (such as trusses embedded in a host hexahedral element) accounting for the exact matrix volume is not possible. Therefore, new algorithms to account for the exact volume of the matrix are needed.

4 Results

4.1 Effects of Volume Redundancy

In this section, we have considered a simple test scenario - a single hexahedral element with a single embedded fiber of 1mm diameter; this number is an approximation of the 1.12 ± 0.8 mm reported in a study of axonal dimensions of a guinea pig optic nerve [29]. Both the matrix and the fiber domains are described using the same material properties. Since both are described using the same material properties, the resultant system should behave as a single homogenous matrix element (with no embedded fibers).

Figure 2 shows the role of volume redundancy in underestimating the axonal strains. In this scenario, we have compared a matrix finite element (hexahedral) with an embedded fiber to a normal homogeneous matrix cube with no fiber. The fiber was described using the same material properties as that of the matrix element - ensuring that the cube with embedded fiber is equivalent to the cube with no fiber. However, when both the configurations were subjected to the same loading conditions, the resulting strains are significantly different - emphasizing the role of volume redundancy in changing the model's strain predictions. Here, we have used a hyperelastic material model to describe the matrix and fibers.

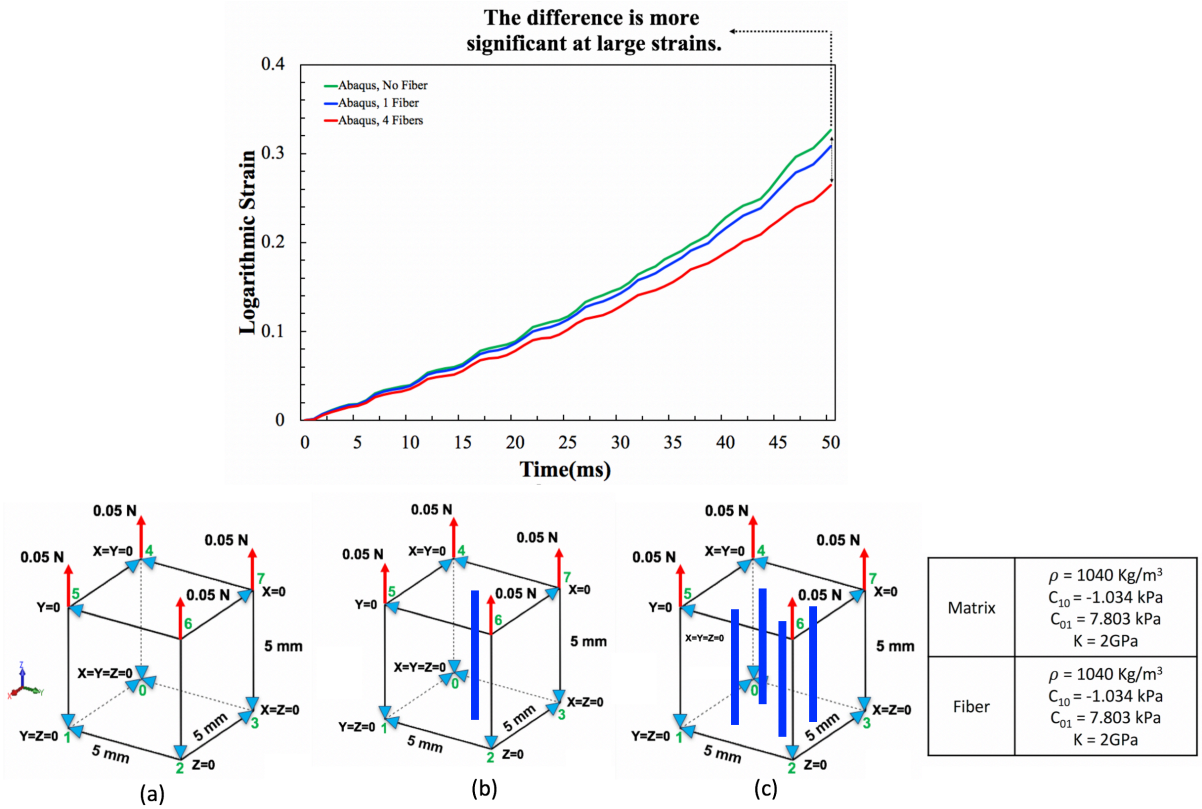

Figure 2. Effect of volume redundancy when the simulations are carried out in ABAQUS. (a) Case-1 shows a homogenous 5 mm cube with Mooney-Rivlin hyperelastic material properties, (b) Case-2 shows a homogenous cube with an embedded fiber of 1mm diameter and (c) Case-3 shows a homogenous cube with 4 embedded fibers of 1 mm diameter each– where both the cube and fiber are described using the same (Mooney-Rivlin hyperelastic) material properties. As the fiber is described using the same material properties as that of the host cube, the systems with the embedded fiber (Case-2 and Case-3) are equivalent to the system with no fiber (Case-1). However, when all the systems were subjected to the same loading conditions (0.2 N total), they resulted in different strains – pointing out to the effect of volume redundancy in the simulation. The redundancy effects are more significant at large strains.

4.1.1 How can we address the mass and force redundancy?

The algorithm used in addressing the issue of volume redundancy for the particular scenario of embedded element being smaller than host elements can be schematically visualized using **Figures 3 and 4**:

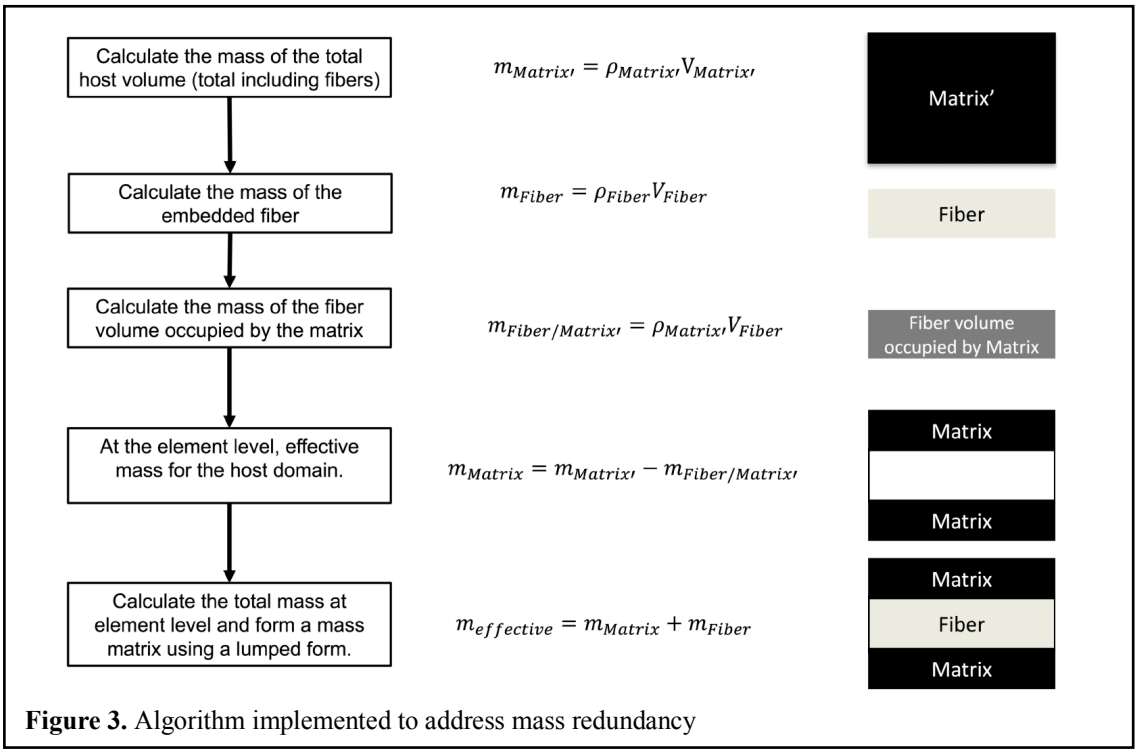

Figure 3. Algorithm implemented to address mass redundancy

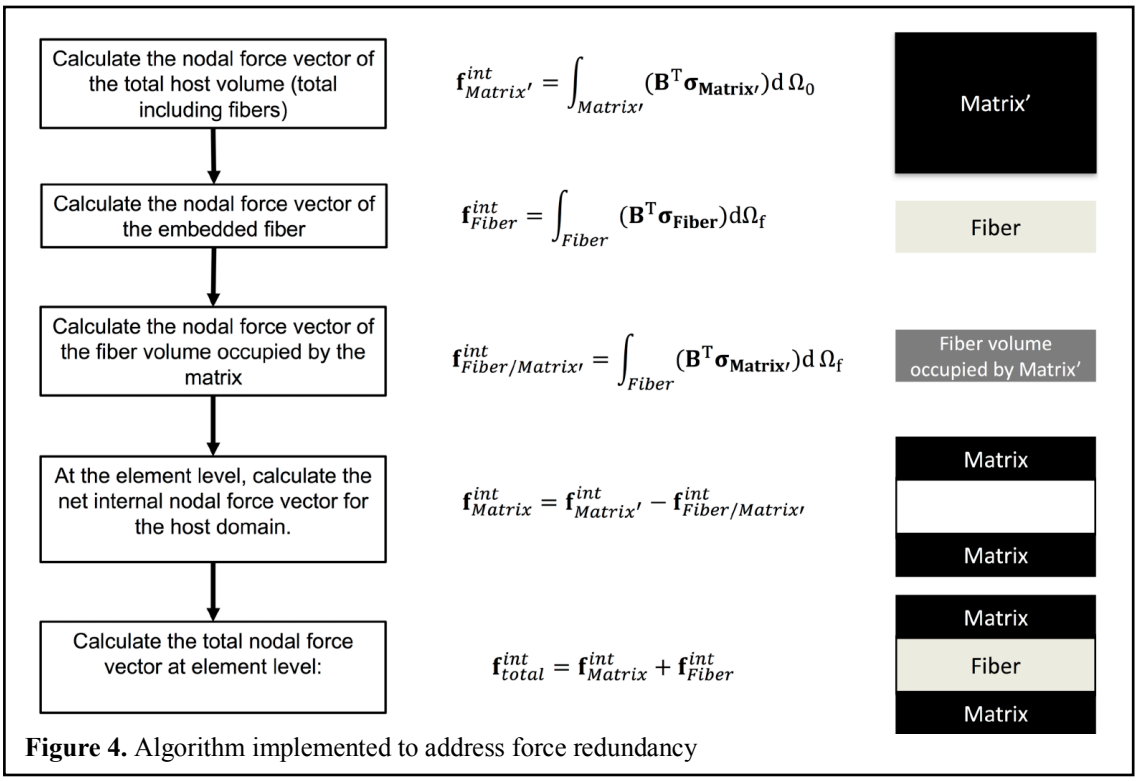

Figure 4. Algorithm implemented to address force redundancy

4.1.2 Finite element algorithm - Addressing Mass and Force Redundancy

In this section, we have modified the explicit time integration FEM algorithm presented in Belytschko Box 6.1 [28] to develop a new algorithm that can implement the embedded element algorithm while addressing volume redundancy. Note that this algorithm was developed in the context of fibers embedded in three-dimensional host finite elements the overall FE algorithm (with explicit time integration scheme) used in developing the embedded element supported finite element library. Box 2 shows the mass subroutine - i.e., calculating the effective mass.

Box 1 Flowchart - FEM with Embedded Fibers - While addressing Volume Redundancy (Newmark - β time integration - explicit central difference method):

1. Initial conditions and initialization – set $\mathbf{v}^0$, $\boldsymbol{\sigma}^0$, and initial values of other material state variables; $\mathbf{d}^0 = 0$, $n = 0$, $t = 0$;
2. Compute $\mathbf{M}$ using *getMassEffective*.
3. Compute $\mathbf{f}_{net}^0$ using *getForceEffective*.
4. Compute accelerations: $\mathbf{a}^0 = \mathbf{M}^{-1}(\mathbf{f}_{net}^0)$ Calculate time step: $\Delta t = \alpha \Delta t_{critical}$ (α is the reduction factor).
   Update time: $t^{n+1} = t^n + \Delta t$
5. Set $\gamma = \frac{1}{2}$ and $\beta = 0$ (Explicit central difference scheme). Partial update of nodal velocities:
   $\tilde{\mathbf{v}}^{n+1} = \mathbf{v}^n + (1-\gamma)\Delta t \mathbf{a}^n$
6. Update Nodal Displacements: $\mathbf{d}^{n+1} = \tilde{\mathbf{d}}^{n+1} + \beta \Delta t^2 \mathbf{a}^{n+1} = \mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2}(1-2\beta)\mathbf{a}^n + \beta \Delta t^2 \mathbf{a}^{n+1}$
7. Enforce displacement boundary conditions
8. Compute $\mathbf{f}_{net}^{n+1}$ using *getForceEffective*.
9. Compute accelerations: $\mathbf{a}^{n+1} = \mathbf{M}^{-1}(\mathbf{f}_{net}^{n+1})$
10. Second partial update of nodal velocities: $\mathbf{v}^{n+1} = \tilde{\mathbf{v}}^{n+1} + \gamma \Delta t \mathbf{a}^{n+1}$
11. Enforce velocity boundary conditions
12. Calculate energies: Internal Energy (W_{int}), External Work (W_{ext}), Kinetic Energy (W_{kin})
    a. $W_{int}^{n+1} = W_{int}^n + \frac{1}{2}\Delta \mathbf{d}^T \left(\mathbf{f}_{int}^n + \mathbf{f}_{int}^{n+1}\right)$ where $\mathbf{f}_{int}$ is the internal nodal force vector
    b. $W_{ext}^{n+1} = W_{ext}^n + \frac{1}{2}\Delta \mathbf{d}^T \left(\mathbf{f}_{ext}^n + \mathbf{f}_{rec}^n + \mathbf{f}_{ext}^{n+1} + \mathbf{f}_{rec}^{n+1}\right)$ where $\mathbf{f}_{ext}$ is the external nodal force vector and $\mathbf{f}_{rec}$ is the nodal reaction force vector.
    c. $W_{kin}^{n+1} = \frac{1}{2}\left(\mathbf{v}^{n+1}\right)^T \mathbf{M}\mathbf{v}^{n+1}$
    d. $\mathbf{f}_{rec}^{n+1} = \mathbf{f}_{int}^{n+1} + \mathbf{M}a^{n+1}$ is the reaction force on the nodes on which no external forces were applied.
13. Check energy balance at time step $n+1$: $|W_{kin} + W_{int} - W_{ext}| \leq \varepsilon_{energy} max(W_{ext}, W_{int}, W_{kin})$
14. Update counter $n \leftarrow n+1$
15. If simulation is not complete goto step 5.

Box 2. Subroutine - *getMassEffective* - A lumped mass calculation was used.

1. Loop over elements n_e
    a. For each element calculate the volume: v_e
    b. Calculate the mass of each element: $m_e = \rho_e v_e$.
    c. Loop over all the embedded fibers: n_f
        i. For each fiber, calculate the volume v_f using length l_f and area of cross section a_f
        ii. Calculate the mass: $m_f = \rho_f l_f a_f$
        iii. Calculate the correction mass: $m_c = \rho_e l_f a_f$
    d. Calculate the new effective element mass: $m_{eff} = m_e + m_f - m_c$
    e. Divide the effective mass equally among all element nodes – form the nodal mass vector.
    f. Use the nodal mass vector to form the element mass matrix $\mathbf{M}_e$ (diagonal mass matrix).
    g. Scatter the mass matrix to update the global mass matrix $\mathbf{M}$.
2. END loop over the elements n_e

Box 3. Subroutine - *getForceEffective*

1. if computing $\mathbf{f}_{net}^{0}$, replace "$n+1$" with "0" throughout subroutine
2. Initialize: $\mathbf{f}_{net}^{n+1} = 0$
3. Update global external nodal forces $\mathbf{f}_{ext}^{n+1}$
4. Loop over elements n_e
   a. GATHER element nodal displacements and velocities
   b. Initialize: $\mathbf{f}_{e}^{int,n+1} = 0$
   c. Loop over quadrature points n_{ξ}
      i. if computing $\mathbf{f}_{net}^{0}$, go to (c)iv.
      ii. Compute deformation measures $\mathbf{F}^{n}(\xi)$, $\mathbf{E}^{n}(\xi)$
      iii. Compute stress: $\boldsymbol{\sigma}^{n}(\xi) = f\left(\mathbf{F}^{n}(\xi)\right)$
      iv. $\mathbf{f}_{e}^{int,n+1} \leftarrow \mathbf{f}_{e}^{int,n+1} + (B^{T}\sigma^{n}\overline{w}_{q}J)_{\xi}$
   d. END quadrature point loop.
   e. Loop over all the embedded fibers n_f
      i. if computing $\mathbf{f}_{net}^{0}$, go to (e)iii.E.
      ii. Calculate the displacements of the embedded nodes from the host nodal displacements: $u_{embed} = [\mathbf{N}]u_{host}$. Shape function matrix is calculated at the embedded nodes.
      iii. Loop over the quadrature points of the fiber n_{ξ_f}
         A. Calculate the fiber quadrature point coordinates $(\xi_{f,iso-host})$ in the iso-parametric coordinate system of the host element using Newton-Raphson algorithm.
         B. Compute deformation measures at these fiber quadrature points: $\mathbf{F}^{n}\left(\xi_{f,iso-host}\right)$
         C. Calculate the fiber stress measure in the host coordinate system: $\boldsymbol{\sigma}_{f}^{n} = f(\mathbf{F}^{n}\left(\xi_{f,iso-host}\right))$ using the fiber material model.
         D. Calculate the correction stress measure in the host coordinate system: $\boldsymbol{\sigma}_{c}^{n} = f(\mathbf{F}^{n}\left(\xi_{f,iso-host}\right))$ using the host element material model.
         E. $\mathbf{f}_{e}^{int,n+1} \leftarrow \mathbf{f}_{e}^{int,n+1} + (B^{T}\boldsymbol{\sigma}_{f}^{n}\overline{w}_{f,q}J_{f} - B^{T}\boldsymbol{\sigma}_{c}^{n}\overline{w}_{f,q}J_{f})_{\xi_f}$ where $J_f = \frac{l_f}{2}a_f$ is the fiber volume Jacobian.
      iv. END loop over the quadrature points of the embedded fibers.
   f. END loop over the embedded fibers.
   g. GATHER external nodal forces on the element, $\mathbf{f}_{e}^{ext,n+1}$
   h. Compute $\mathbf{f}_{e}^{net,n+1} = \mathbf{f}_{e}^{ext,n+1} - \mathbf{f}_{e}^{int,n+1}$
   i. SCATTER $\mathbf{f}_{e}^{net,n+1}$ to global $\mathbf{f}_{net}^{n+1}$

4.2 Validation of the code for large strain embedded element problems

Figure 5 shows the comparison of logarithmic strains before and after activating the volume redundancy correction in EEMA. This figure also shows the comparison of EEMA's behavior to that of ABAQUS -- improving our confidence in the FE library. The main takeaways from this validation effort include:

1. When both cases were subjected to the same loading conditions in ABAQUS, the results are not the same -- thus pointing out to the volume redundancy issue in ABAQUS (same takeaway as that of the **Figure 2**).
2. Before the volume redundancy was corrected, logarithmic strain calculated from EEMA is same as that of the ABAQUS results -- strengthening our confidence in the FE library (EEMA). This also serves as a validation of the code for large strain finite element analysis.
3. After the volume redundancy corrections, EEMA's results match well with the Case-1 results in ABAQUS -- demonstrating that EEMA can accurately handle the volume redundancy corrections (even in the large strain domain).

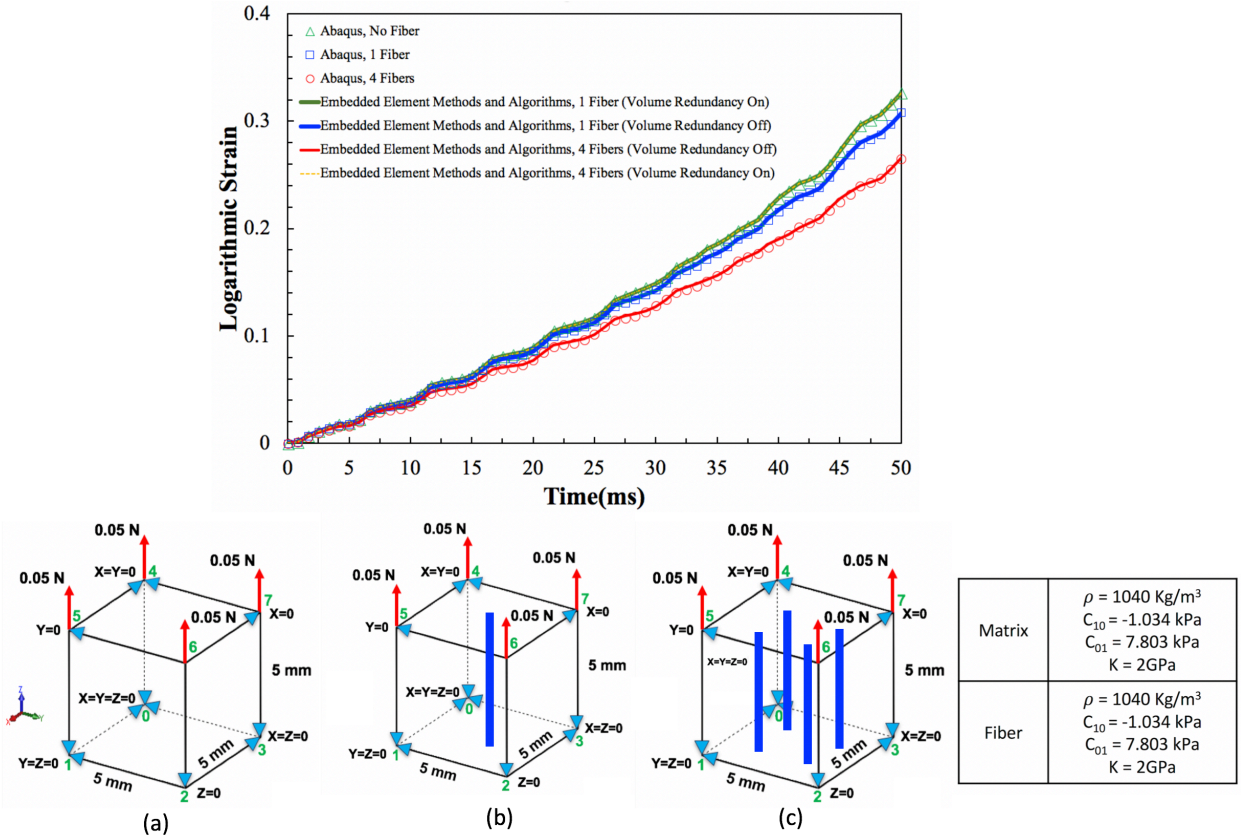

Figure 5. Figure shows the EEMA's behavior (Logarithmic strain vs. time) compared to that of ABAQUS. In this figure, (a) shows Case-1 where we have a single homogeneous cube with Mooney-Rivlin hyperelastic material properties (b) shows Case-2 where we have a single homogenous cube with embedded fiber and (c) shows case-3 where we have a single homogenous cube with 4 embedded fibers. The embedded fibers were provided with the same material properties as that of the host cube around it. This ensures that all cases are equivalent systems and should show same strains under the same loading conditions. All cases were subjected to uniaxial extension under a total loading of 0.2 N. The multiple takeaways from the results include: (1) When all cases were subjected to the same loading conditions in ABAQUS, the results are not the same – thus pointing out to the volume redundancy issue in ABAQUS (same takeaway as that of the Figure 1); (2) Before the volume redundancy was corrected, the logarithmic strain calculated from EEMA was the same as the result from ABAQUS – strengthening our confidence in the FE library (EEMA); (3) After the volume redundancy corrections, EEMA's results match well with the Case-1 results in ABAQUS – demonstrating that EEMA efficiently handles the volume redundancy corrections (even in the large strain domain).

For the example shown in **Figure 5**, there is a significant strain mismatch between the corrected and uncorrected cases. The uncorrected case could be viewed as being artificially stiff. This artificial stiffening effect becomes more pronounced with increased fiber volume fraction and greater matrix stiffness, relative to the fiber stiffness. **Figure 6** expands upon **Figure 5**, by presenting four additional cube sizes for comparison, with all results created using EEMA. A typical brain mesh consists of hexagonal elements with edge lengths ranging from 1 to 7 mm. Therefore, the sizes shown in **Figure 6** are of practical interest in the application of FE human head models. In **Figure 6**, all cube sizes contain a single 1 mm diameter fiber and the force applied to each has been tailored to result in approximately the same final corrected strain. Therefore, **Figure 6** highlights the significant and varying artificial stiffening effect, as seen in the uncorrected cases. As mentioned earlier, the strain mismatch is greater for increased fiber volume fraction, or in this case, decreased cube size. The percent increases in final strain, resulting from the volume redundancy correction, are presented for each cube size in **Figure 6**. As shown, the artificial stiffening effect is more severe for smaller matrix element sizes.

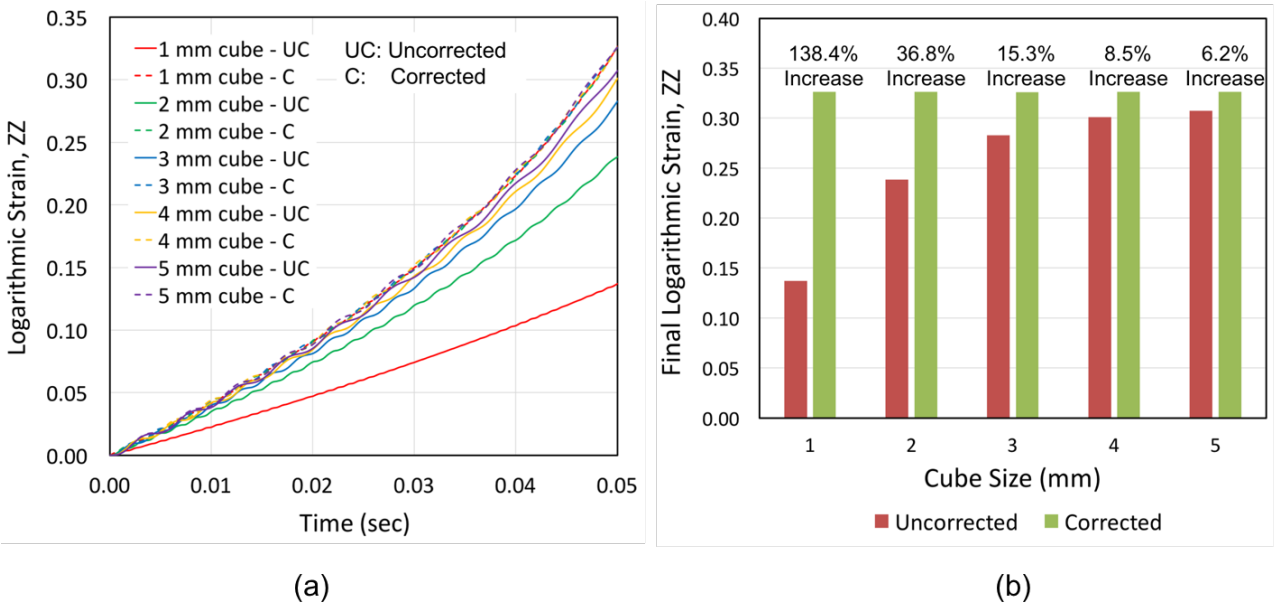

Figure 6. Figure shows logarithmic strain results for 5 cubes of different sizes. Each cube contains the same fiber of 1mm diameter, the fiber material is the same as the cube. As the cube size gets smaller, the effect of the volume redundancy is more pronounced since there is a higher overlap of volume

5 Discussion

In this paper, efforts were made to address the issue of volume redundancy - a limitation associated with the embedded element technique (in ABAQUS). Novel finite element algorithms were developed to address the corresponding mass and nodal force redundancies that arise from this mesh superposition technique. The new finite element algorithms included element-level corrections that can accurately account for correct mass and internal nodal force contributions. Simultaneously, a corresponding general-purpose finite element library was developed that can handle this modified embedded element technique. This library was developed in C++ and is made available to the public through GitHub, https://github.com/PSUCompBio/compbio.

There have been some excellent attempts to deal with volume redundancy in the past. For example, The Integrated Mesh Method [25] deals with volume redundancy by modifying the shape functions, the Domain Superposition Technique [20] seems to modify material parameters to adjust for fiber contributions, and the Tabatabaei Method [23] addresses volume redundancy using a stiffness correction. However, these methods may not be well suited for the complex fiber bundle networks like is observed in brain diffusion tractography and in applications employing explicit dynamics. In addition, in the applications where these previous methods are applied, the size of the embedded region or fibers is comparable to the host region, whereas when modeling the brain the fibers are a lot smaller than the matrix. In our approach we effectively subtract off unwanted, overlapping contributions to mass and force. It is a very practical approach.

The embedded element approach developed here constrains the fibers to the matrix in an affine transformation, i.e., perfect bonding with no slip. The end points of each fiber are constrained to the faces of the matrix element it is embedded in and hence the motion of the fiber only depends on the motion of the matrix element as can be seen at step e. ii. of **Box 3**. One advantage of this approach is that complex networks of DTI fiber bundles can be represented with the embedded algorithms and any anisotropic effects that the underlying tissue may have can be incorporated naturally into the model, as opposed to using constitutive models, such as a transversely isotropic law. However, one disadvantage of the method presented here is that non-affine deformations of the fiber tract bundles, such as fiber/matrix slip, is not included with the current implementation. However, it is possible to include slip between the fiber and matrix by modifying the fiber displacement calculation to be [16]:

$$u_{fiber} = u_{embed} + u_{slip} \tag{20}$$

However, modeling slip would imply that the fibers now have their own degrees of freedom thereby increasing computation cost and tortuosity of the code. This is reserved for future work.

This new formulation is extensive; as seen in **Boxes 1** and **3**, there are calculations that loop over every element, within every timestep. Hence, as the mesh size decreases and the number of elements increase, the runtime increases significantly. To overcome this issue, parallel programming and multithreading has been added to the embedded element code. Using the maximum number of threads that a machine can support (hardware concurrency) the total elements in the mesh are divided across all threads. Hence all the calculations in the *getforce* function are carried out in parallel. Since the *getforce* calculation is the most time-intensive area of the code the parallelization gives a significant speed up. With the success of the parallelization of the *getforce* function, efforts are now being directed to parallelize more parts of the code to achieve even higher speed ups.

The capability we have added here is important to the vision we foresee in individual-specific multiscale modeling of the brain. For military loading, this vision is schematically represented in **Figure 7**. The core technology to be developed is a new multiscale method to model brain tissue that will transform how neuroimaging is used in engineering simulations by explicitly representing fiber tractography in models. Furthermore, we anticipate the explicit inclusion of fiber tractography will be critical in a number of subtopics of brain science [30]. To this end, we suggest the concept of a "*Digital Brain*" (**Figure 7**) that serves as a hub for collecting, analyzing, and disseminating data on a brain.

We see computer models of brain injury as a potential gold standard for computational medicine, where simulations serve as a repository or "hub" to maintain and translate data to clinical applications and to advance experimental injury mechanics. Advancements have the real potential to change the paradigm of how neurotrauma is diagnosed and monitored. Kinematic neuroimaging, such as elastography [31]–[33],

are far from being wearable and mobile, so simulations remain the *only* way to "see" how external forces to the head (i.e., impacts or blasts) are translated inside the brain to tissue strains, enabling a noninvasive evaluation of mild TBI. This would have a transformative impact in quantitative TBI diagnostics and monitoring.

Figure 7. Our long-term vision is that brain simulations are used in conjunction with wearable sensors to provide a history-dependent analysis of structural and functional changes in brain tissue. This schematic shows how the *Digital Brain* sits permanently in the cloud always listening for an impact or blast measurement from a wearable sensor. When an impact occurs, a simulation is automatically initiated, and analysis results are compiled and disseminated as needed.

The addition of an embedded element approach for modeling axonal fiber bundles also contributes to the emerging field of precision computational medicine, which is just beginning to transform healthcare. Furthermore, the composite modeling techniques could extend to other fibrous materials such as ligaments and muscle, and the new multiscale computational method is not limited to structural mechanics, as it can be applied to functional bioelectrics.

6 Conclusion

The embedded element method is a useful approach for modelling computational biomechanics models. ABAQUS, a commercial software, provides this feature; however, by overestimating the system volume it overestimates the internal forces generated during loading conditions leading to inaccurate results. The development of this new approach to embedded element modelling takes care of this volume redundancy by subtracting the excess volume and hence the excess force.

7 Acknowledgement

The authors gratefully acknowledge the support provided by CFDRC, Inc. under a sub-contract funded by the Department of Defense, Department of Health Program through contract W81XWH-14-C-0045, the U.S. Army Research Laboratory (ARL) at the Aberdeen Proving Grounds under contract W15P7T-10-D-D416, the Defense Health Agency under the contract W81XWH-14-C-0045 and the US Department of Defense through the contracts W15P7T-10-D-D416, W81XWH-17-C-0216, DOTC-17-01-INIT0086. The authors also thank the Pennsylvania State University Institute for Cyberscience for providing the computational resources required for this work.

8 Appendix

8.1 How does our element level corrections compare to the transverse isotropic constitutive modeling?

In this section, we will look into some of the implementation details of selected areas of this finite element formulation -- that are specific to the embedded element technique.

8.2 Transformation Matrix

This matrix can be used to transform a tensor from one coordinate system to another coordinate system. Let's assume that the first coordinate system is represented by the basis (m_1, m_2, m_3) and the second coordinate system is represented by the basis (e_1, e_2, e_3), then the transformation matrix to transform a tensor from first coordinate system to the second coordinate system can be defined as:

$$\mathbf{T} = \begin{bmatrix} m_1 \cdot e_1 & m_1 \cdot e_2 & m_1 \cdot e_3 \\ m_2 \cdot e_1 & m_2 \cdot e_2 & m_2 \cdot e_3 \\ m_3 \cdot e_1 & m_3 \cdot e_2 & m_3 \cdot e_3 \end{bmatrix} \tag{21}$$

i.e., if a tensor in the first coordinate system is $\boldsymbol{\sigma}_m$ and the same tensor in the second coordinate system is $\boldsymbol{\sigma}_e$, then

$$\boldsymbol{\sigma}_c = \mathbf{T}\boldsymbol{\sigma}_m\mathbf{T}^T \tag{22}$$

This matrix is frequently used in transforming a tensor of interest from fiber to matrix coordinate systems and vice-versa.

8.3 Adding embedded fiber mass to the host element mass:

For implementation purposes, an element level direct lumped mass matrix was used. The below-listed procedure explains the detailed calculation of global mass matrix using the host as well as the embedded element information.

1. Calculate the host element mass using the volume and density information
2. Calculate the total fiber volume embedded in this particular host element
3. Calculate the fiber mass using its volume (Step-2) and using the material model information.

4. Calculate the effective element mass by adding the host and embedded element masses.
5. Divide this effective element mass among all the host element nodes equally and form an element level diagonal mass matrix.
6. Scatter this element level mass matrix into a global mass matrix and repeat this step for all the elements.

8.4 Deformation in the embedded elements

Since the displacements of the embedded nodes are assumed to be a function of the displacements of the host nodes, it can be shown that the deformation gradient of the embedded elements will be same as that of the host element.

$$\mathbf{F}_{host} = \frac{\partial \mathrm{x}_{host}}{\partial \mathrm{X}_{host}} = \frac{\partial (\mathrm{X}_{host} + \mathrm{u}_{host})}{\partial \mathrm{X}_{host}} = \mathbf{I} + \frac{\partial \mathrm{u}_{host}}{\partial \mathrm{X}_{host}} \tag{23}$$

But, since $\mathrm{u}_{embed} = [\mathbf{N}]\mathrm{u}_{n,host}$ and $\mathbf{B} = \frac{\partial [\mathrm{N}]}{\partial \mathrm{X}_{host}}$,

$$\frac{\partial \mathrm{u}_{host}}{\partial \mathrm{X}_{host}} = \mathbf{B}\mathrm{u}_{n,host} = \frac{\partial \mathrm{u}_{embed}}{\partial \mathrm{X}_{host}} \tag{24}$$

Therefore, we can see that the above equation 22 can be extended such that

$$\mathbf{F}_{host} = \mathbf{I} + \frac{\partial \mathrm{u}_{host}}{\partial \mathrm{X}_{host}} = \mathbf{I} + \frac{\partial \mathrm{u}_{embed}}{\partial \mathrm{X}_{host}} = \mathbf{F}_{embed} \tag{25}$$

In the above equations, $\mathbf{F}_{host}$ represents the deformation gradient of the host element, $\mathbf{F}_{\boldsymbol{embed}}$ represents the deformation gradient of the embedded element, x_{host} represents the current configuration of the host element, $\mathrm{X}_{\boldsymbol{embed}}$ represents the reference configuration of the host element, u_{host} represents the continuum displacement of the host domain, $\mathrm{u}_{n,host}$ represents the nodal displacements of the host domain, u_{embed} represents the nodal displacement vector of the embedded elements, **B** represents the strain displacement matrix and [**N**] represents the shape function matrix of the host element.

8.5 Adding the nodal force contribution from the embedded fibers back to that of the host elements:

Adding the nodal force vector of the truss element to the host parent element is not straight forward and involves a complicated procedure. The overall methodology adopted here can be classified into multiple steps:

1. Calculating integration points
2. Calculating the internal nodal force vector

Calculating integration points: This step involves a calculating the integration points and moving between coordinate systems. Note that this step in explained in the context of truss elements embedded in hexahedral host elements. Here, we have three coordinate systems. First one is the global coordinate system (coordinate system in which the host and embedded meshes are defined). The second one is the iso-parametric coordinate system of the host element. And the third one is the iso-parametric coordinate system of the embedded fiber. We will designate these systems as G (global system), LH (iso-parametric coordinate system of the host element) and LE (iso-parametric coordinate system of the embedded fiber), respectively for our future discussion. Since we will be integrating over the fiber volume to calculate the internal nodal force vector contribution from the embedded fibers and adding it to the nodal force vector from the host element, we need the integration points of the embedded fibers calculated in the LH coordinate system (iso-parametric coordinate system of the host element). This can be done using an iterative solver. In our library, we have currently implemented the Newton-Raphson solver for this

purpose. More information on the mapping of integration points or the Newton-Raphson solving can be obtained from the dissertation work by Hartl [16].

Calculating the internal nodal force vector of the embedded fiber: Once the fiber integration points are calculated in the host element coordinate system, the nodal force vector can be calculated by integrating over the fiber volume using the following equation:

$$\mathbf{f}_{e,fiber}^{int} = \int_{fiber} \mathbf{B}^T \boldsymbol{\sigma}\, d\Omega_f = \sum_{n_\xi} \mathbf{B}^T \boldsymbol{\sigma} * wts(\xi) * jacobian_{fiber} \quad (26)$$

In the above equation, n_ξ represents the number of Gauss quadrature points, and $wts(\xi)$ represents the Gaussian weights of the particular quadrature point.

8.6 Calculating the stress along the fiber

As seen in the above sub-section, both the host and embedded elements will have the same deformation gradient. Therefore, the Cauchy stress tensor calculated using the deformation gradient and the embedded fiber material properties will be transformed into the fiber coordinate system (a unit vector along the fiber axis forms the basis of this coordinate system), and the corresponding fiber stress will be calculated.

$$\boldsymbol{\sigma}_{fiber} = f(\mathbf{F}, \text{Fiber material model}) \quad (27)$$

where **F** is the deformation gradient and $\boldsymbol{\sigma}_{fiber}$ is the fiber Cauchy stress tensor calculated in the host element coordinate system.

$$\boldsymbol{\sigma}_{fiber-axis} = \mathbf{T} \cdot \boldsymbol{\sigma}_{fiber} \cdot \mathbf{T}^T \quad (28)$$

8.7 Calculating the strain along the fiber

The deformation gradient tensor can be used to calculate the strain projected along the fiber using the following equations \ref{fiber-strain-eqn}:

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} \quad (29)$$

$$\lambda_{fiber} = \sqrt{a_0 \cdot \mathbf{C}a_0} \quad (30)$$

$$\varepsilon_{fiber} = log\,(\lambda_{fiber}) \quad (31)$$

where **C** is the right Cauchy-Green deformation tensor, λ_{fiber} is the stretch along the fiber, a_0 is the unit vector along the fiber direction and ε_{fiber} is the strain along the fiber.

8.8 EEMA Energy Plot

Since embedded element technique has been formulated using the finite element method, it would be essential for the energy to be conserved. **Figure 7** shows a system and its corresponding energy plot. As expected, the total energy of the system remains zero throughout the span of the simulation.

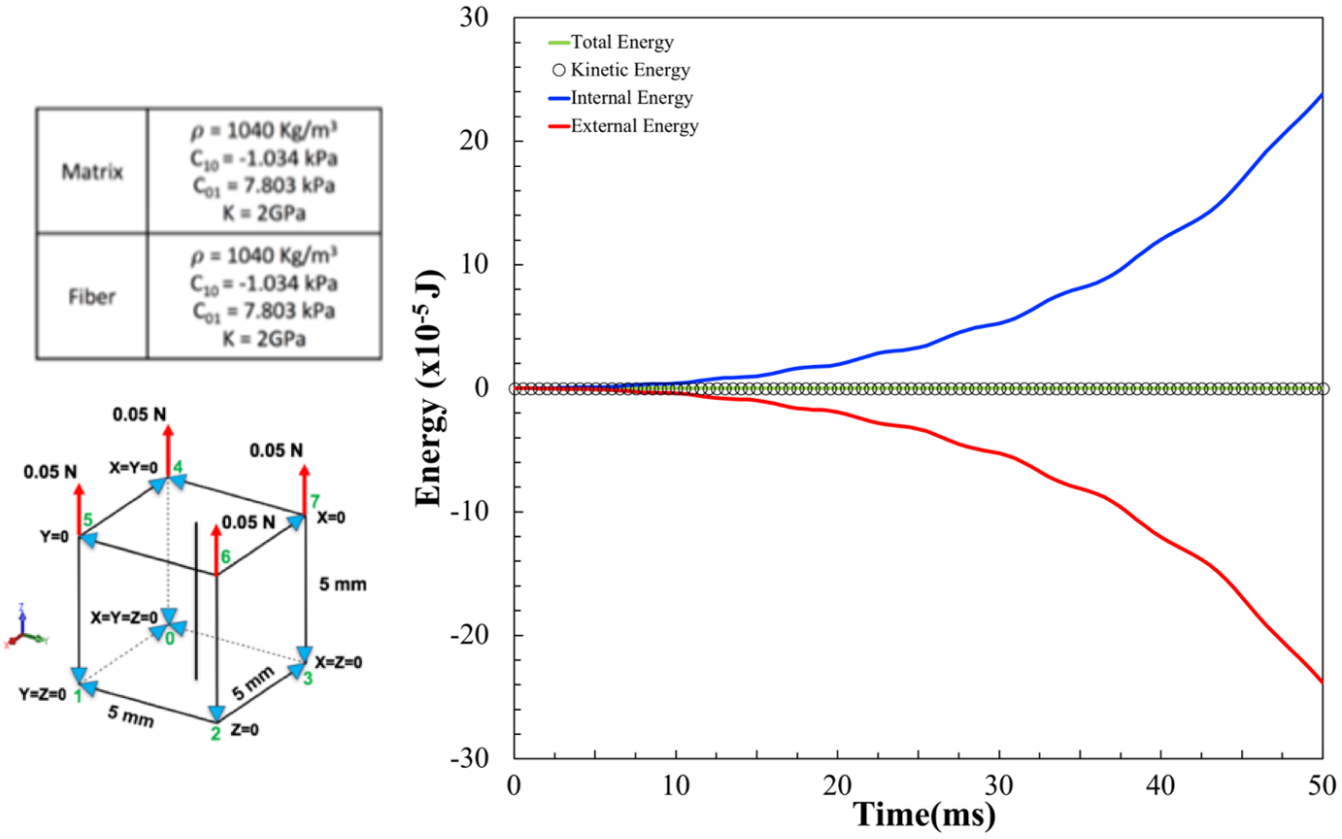

Figure 8. Energy Plot for a Cube with one Fiber (Addressing Volume Redundancy)

9 References

[1] S. Mori and J. Zhang, "Principles of diffusion tensor imaging and its applications to basic neuroscience research," *Neuron*, vol. 51, no. 5, pp. 527–539, 2006.

[2] Y. Assaf and O. Pasternak, "Diffusion tensor imaging (DTI)-based white matter mapping in brain research: a review," *J. Mol. Neurosci.*, vol. 34, no. 1, pp. 51–61, 2008.

[3] P. Hagmann *et al.*, "Mapping the structural core of human cerebral cortex," *PLoS Biol.*, vol. 6, no. 7, p. e159, 2008.

[4] O. Sporns, G. Tononi, and R. Kötter, "The human connectome: a structural description of the human brain," *PLoS Comput. Biol.*, vol. 1, no. 4, p. e42, 2005.

[5] W. Zhao, Y. Cai, Z. Li, and S. Ji, "Injury prediction and vulnerability assessment using strain and susceptibility measures of the deep white matter," *Biomech. Model. Mechanobiol.*, pp. 1–19, 2017.

[6] C. Giordano, S. Zappalà, and S. Kleiven, "Anisotropic finite element models for brain injury prediction: the sensitivity of axonal strain to white matter tract inter-subject variability," *Biomech. Model. Mechanobiol.*, pp. 1–25, 2017.

[7] S. Makarov, A. P. Leone, and A. Nummenmaa, "Researching Fiber Networks: Computational Modeling of Complex Fibrous Tissue Geometries," *IEEE Pulse*, vol. 8, no. 4, pp. 58–61, 2017.

[8] N. De Geeter, L. Dupré, and G. Crevecoeur, “Modeling transcranial magnetic stimulation from the induced electric fields to the membrane potentials along tractography-based white matter fiber tracts,” *J. Neural Eng.*, vol. 13, no. 2, p. 026028, 2016.
[9] E. Gleichgerrcht, J. Fridriksson, C. Rorden, and L. Bonilha, “Connectome-based lesion-symptom mapping (CLSM): a novel approach to map neurological function,” *NeuroImage Clin.*, vol. 16, pp. 461–467, 2017.
[10] A. T. Toosy, O. Ciccarelli, G. J. Parker, C. A. Wheeler-Kingshott, D. H. Miller, and A. J. Thompson, “Characterizing function–structure relationships in the human visual system with functional MRI and diffusion tensor imaging,” *Neuroimage*, vol. 21, no. 4, pp. 1452–1463, 2004.
[11] D. Werring, C. Clark, G. Parker, D. Miller, A. Thompson, and G. Barker, “A direct demonstration of both structure and function in the visual system: combining diffusion tensor imaging with functional magnetic resonance imaging,” *Neuroimage*, vol. 9, no. 3, pp. 352–361, 1999.
[12] H. T. Garimella and R. H. Kraft, “Modeling the mechanics of axonal fiber tracts using the embedded finite element method,” *Int. J. Numer. Methods Biomed. Eng.*, vol. 33, no. 5, 2017.
[13] H. T. Garimella and R. H. Kraft, “A new computational approach for modeling diffusion tractography in the brain,” *Neural Regen. Res.*, vol. 12, no. 1, p. 23, 2017.
[14] H. T. Garimella, P. Andrzej, and R. H. Kraft, “Do blast-induced skull flexures result in axonal deformation?,” *PloS One*, vol. Accepted. December 20, 2017.
[15] V. Garimella, “An Embedded Element Based Human Head Model to Investigate Axonal Injury,” PhD Thesis, Pennsylvania State University, 2017.
[16] H. Helmut, “Development of a Continuum-Mechanics-Based Tool for 3D Finite Element Analysis of Reinforced Concrete Structures and Application to Problems of Soil-Structure Interaction,” Graz University of Technology.
[17] H. T. Garimella, H. Yuan, B. D. Johnson, S. L. Slobounov, and R. H. Kraft, “A two-fiber anisotropic constitutive model of human brain with intravoxel heterogeneity of fiber orientation using diffusion spectrum imaging (DSI),” in *ASME 2014 International Mechanical Engineering Congress and Exposition*, 2014, p. V003T03A011–V003T03A011.
[18] J. Fish, “The s-version of the finite element method,” *Comput. Struct.*, vol. 43, no. 3, pp. 539–547, 1992.
[19] J. Fish and T. Belytschko, “Elements with embedded localization zones for large deformation problems,” in *Computational Structural Mechanics & Fluid Dynamics*, Elsevier, 1988, pp. 247–256.
[20] W.-G. Jiang, S. R. Hallett, and M. R. Wisnom, “Development of domain superposition technique for the modelling of woven fabric composites,” *Mech. Response Compos.*, pp. 281–291, 2008.
[21] D. Ohyama, T. Kurashiki, Y. Watanabe, Y. Fujita, and M. Zako, “Estimation of mechanical behavior of braided composites based on mesh superposition method,” in *18th Int. Conf. on Composite Materials, Jeju island, Korea*, 2011.
[22] S. Tabatabaei, S. V. Lomov, and I. Verpoest, “Assessment of embedded element technique in meso-FE modelling of fibre reinforced composites,” *Compos. Struct.*, vol. 107, pp. 436–446, 2014.
[23] S. Tabatabaei and S. V. Lomov, “Eliminating the volume redundancy of embedded elements and yarn interpenetrations in meso-finite element modelling of textile composites,” *Comput. Struct.*, vol. 152, pp. 142–154, 2015.
[24] M. Zako, T. Kurashiki, F. Kubo, and M. Imura, “A multi-scale analysis for structural design of fibrous composites–M3 method,” in *15th international conference on composite materials (ICCM-15), CD ed., Durban*, 2005.

[25] E. V. Iarve, D. H. Mollenhauer, E. G. Zhou, T. Breitzman, and T. J. Whitney, “Independent mesh method-based prediction of local and volume average fields in textile composites,” *Compos. Part Appl. Sci. Manuf.*, vol. 40, no. 12, pp. 1880–1890, 2009.
[26] D. W. Shattuck and R. M. Leahy, “BrainSuite: An automated cortical surface identification tool,” *Med. Image Anal.*, vol. 6, no. 2, pp. 129–142, Jun. 2002.
[27] A. A. Elwi, “Nonlinear analysis of axisymmetric reinforced concrete structures.,” 2002.
[28] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary, *Nonlinear finite elements for continua and structures*. John wiley & sons, 2013.
[29] J. Guy, E. A. Ellis, K. Kelley, and G. M. Hope, “Spectra of G ratio, myelin sheath thickness, and axon and fiber diameter in the guinea pig optic nerve,” *J. Comp. Neurol.*, vol. 287, no. 4, pp. 446–454, 1989.
[30] H. T. Garimella and R. H. Kraft, “A new computational approach for modeling diffusion tractography in the brain,” *Neural Regen. Res.*, vol. 12, no. 1, pp. 23–26, Jan. 2017.
[31] C. L. Johnson *et al.*, “Local mechanical properties of white matter structures in the human brain,” *NeuroImage*, vol. 79, pp. 145– 152, 2013.
[32] C. A. Guertler, R. J. Okamoto, J. L. Schmidt, A. A. Badachhape, C. L. Johnson, and P. V. Bayly, “Mechanical properties of porcine brain tissue in vivo and ex vivo estimated by MR elastography,” *J. Biomech.*, vol. 69, pp. 10–18, Mar. 2018.
[33] P. . Bayly *et al.*, “Measurement of brain biomechanics in vivo by magnetic resonance imaging,” in *Application of Imaging Techniques to Mechanics of Materials and Structures:Proceedings of the 2010 Annual Conference on Experimental and Applied Mechanics*, vol. 4, T. Proulx, Ed. Springer, 2013, pp. 117–128.