

RESEARCH ARTICLE

Open Access



An energy-based study of the embedded element method for explicit dynamics

Valerie A. Martin¹, Reuben H. Kraft^{1*} , Thomas H. Hannah¹ and Stephen Ellis²

*Correspondence:
rhk12@psu.edu

¹Department of Mechanical Engineering, Penn State University, University Park, USA

²Los Alamos National Lab, Los Alamos, USA

Abstract

The embedded finite element technique provides a unique approach for modeling of fiber-reinforced composites. Meshing fibers as distinct bundles represented by truss elements embedded in a matrix material mesh allows for the assignment of more specific material properties for each component rather than homogenization of all of the properties. However, the implementations of the embedded element technique available in commercial software do not replace the material of the matrix elements with the material of the embedded elements. This causes a redundancy in the volume calculation of the overlapping meshes leading to artificially increased stiffness and mass. This paper investigates the consequences in the energy calculations of an explicit dynamic model due to this redundancy. A method for the correction of the redundancy within a finite element code is suggested which removes extra energy and is shown to be effective at correcting the energy calculations for large amounts of redundant volume.

Keywords: Finite element analysis (FEA), Fibers, Polymer-matrix composites, Embedded elements

Introduction

The embedded element method is a superposition technique in finite element analysis (FEA) where two independent element meshes are superimposed on one another. One mesh is embedded in the other by tying the degrees of freedom of the embedded nodes to the degrees of freedom of the host elements. The embedded element method has evolved both in its implementation and application since it originated and has been used under many different names. Initially, it was used as a localization method, similar to the extended finite element method, where the element shape functions were modified to include embedded regions that could accommodate highly localized strain fields [1,2]. Later, these embedded regions influenced the creation of higher order elements [3–5] as well as separate refined meshes that replaced a coarse mesh in localized regions [6,7]. This technique was used in various ways to model woven fiber reinforced composites, from multi-scale localization models [8] to microscale models of fiber regions meshed with continuum elements embedded in a matrix material [4,9,10]. Rather than using a solid element mesh, some authors used truss or spring elements to represent the stiffness of fibers embedded in continuum element mesh of a matrix material [11–13]. This has been

a popular method for modeling rebar reinforced concrete as well [14–17]. In the last few years, embedded truss elements have also been used to model bio-material such as white matter in the brain [18–20].

However, in the implementation of the embedded element method, the volume of the host elements is not modified to “make room” for the embedded elements. The volumes of the two meshes occupy the same space. If this is unaccounted for, it causes internal energy and kinetic energy to be miscalculated by double counting volume of the embedded mesh [20,21]. The amount of redundant volume is equal to the volume of the embedded mesh so in some cases, such as in reinforced concrete, the redundancy is considered negligible. However, as the volume of the embedded regions increases the additional mass and stiffness of the redundant volume need to be addressed. There have been several different methods introduced to account for the redundant volume [2,4,8–10,21]. The most common way is, rather than using the matrix material’s true properties, to use an effective medium. This is found from taking measured properties (stiffness, density) of the total composite and subtracting the stiffness of the embedded elements [4,9,19,21,22]. Some authors have used modified integration schemes, averaging or ignoring overlapping integration points [2,10,23]. However, these methods were created for elastic problems, and none for time-dependent dynamic problems or explicit finite element techniques. In a paper on modeling brain matter, Garimella et al. [20] proposed a modification to the explicit finite element method to internally correct the volume redundancy that examined the stress response but did not analyze energy.

We are extending Garimella’s work to study energy to investigate the possibility of using the embedded element method of FEA to create meso-scale models of fiber reinforced composites, particularly for ballistic applications. If fibers could be modeled as an independent mesh of truss elements, it would simplify mesh generation while maintaining a distinction between the fiber and matrix material, which is useful for meso or macro-scale modeling [6,21]. By using a continuum material to represent the composite’s matrix and embedded line elements to represent bundles of fibers, the orthotropic nature of the material would be naturally captured. It could also have the ability to model curved plates without the complexity of defining local material axes. The element mesh is easy to create, yet retains distinction between the fiber and matrix components for meso-scale analysis. An example of this type of mesh is shown in Fig. 1. Since fiber reinforced composites have a large fiber volume fraction (87% is common for ballistic composites), it is important to take into account the effects of the volume redundancy in the embedded element method in order to get accurate measures of the amount of energy absorbed and dispersed by the material.

This paper investigates the energy terms affected by volume redundancy in the embedded element method and how the volume redundancy effects problem solutions, particularly in dynamic problems with high fiber volume fraction materials.

Methods

Energy and the finite element method

In the finite element method for structural mechanics, we wish to find the displacement field of a structural system given the system’s initial position and a set of boundary conditions on the displacement and/or stress on a surface of the system. The displacement

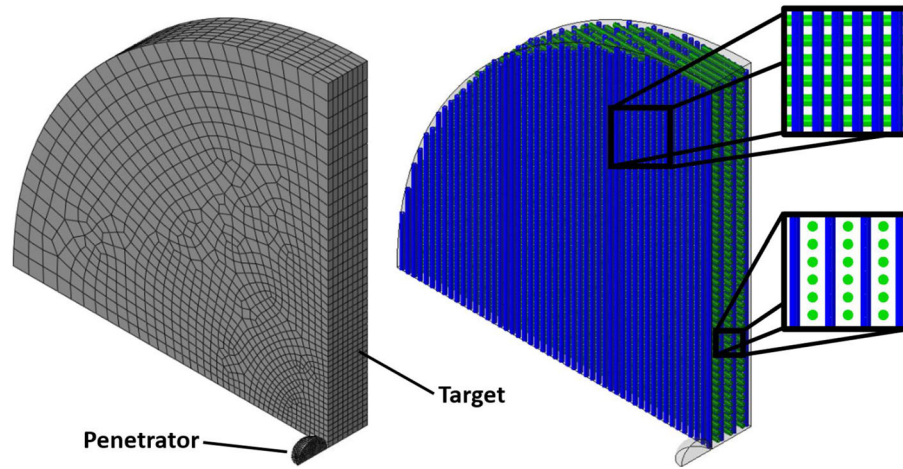


Fig. 1 Example of a 1/4 disk meshed with embedded truss elements to represent fiber bundles layered in a 0/90° in a composite

field can then be used to calculate stress and strain from the strain displacement equation and a constitutive equation. Normally, this problem results in solving the partial differential stress equilibrium equation. The principle of virtual work allows the solution of the problem via integrals. As an expression of the conservation of energy, the principle of virtual work states that the work done on the system by external forces is equal to the total amount of internal energy stored in the system plus the kinetic energy and any dissipated energy. Starting with a system with known boundary conditions, the principle of virtual work can be derived from conservation of energy as follows. For an elastic structural problem with negligible amounts of heat transfer involved and no energy dissipation, the internal energy is equal to the strain energy of the structure, which is found by Eq. (1).

$$W_{int} = \int C_{ijkl} u_{i,j} u_{k,l} dV \tag{1}$$

Here u_i is the displacement field of the system and C_{ijkl} is the material stiffness tensor. The kinetic energy and the work done by external forces can be expressed as Eqs. (2) and (3).

$$W_{kin} = \int \rho \frac{d^2 u_i}{dt^2} u_i dV \tag{2}$$

$$W_{ext} = \int \rho b_i u_i dV + \int_{S_2} t_i u_i dA \tag{3}$$

In these equations, ρ is the material density, b_i is a vector representing body forces (such as gravity), and t_i is the vector of the surface traction (applied force per unit area) on the surface S_2 . The total system potential energy is equal to the summation of these work terms.

$$E_{total} = W_{int}(u_i) - W_{ext}(u_i) + W_{kin}(u_i) \tag{4}$$

According to the principle of minimum total potential energy, the system will move to an equilibrium position that will minimize the total potential energy. One way to tell if a system is at equilibrium is to perturb it by some amount and see if the system returns to the original state. By perturbing the potential energy equation by some virtual displacement field δu_i , we get the virtual work equation with the virtual values of internal, kinetic, and

external energy. An equilibrium solution will be one that satisfies this equation for all kinematically admissible virtual displacements [24–26].

$$\int C_{ijkl}u_{i,j}\delta u_{k,l} dV - \int \rho b_i\delta u_i dV - \int_{S_2} t_i\delta u_i dA + \int \rho \frac{d^2u_i}{dt^2}\delta u_i dV = 0 \quad (5)$$

The virtual work equation is then solved by discretizing the system into a set of nodes. The displacement field between the nodes is assumed to be a linear interpolation of the nodal displacements. This done by using interpolation functions, often called shape functions, $N^a(x_j)$. The displacement field $u_i(x_j)$ is then defined by Eq. (6).

$$u_i(x_j) = \sum_{a=1}^n N^a(x_j) u_i^a \quad (6)$$

The summation is over all nodes in the model, where u_i^a is the displacement at node a. Substituting this definition of u_i into the virtual work equation gives a linear system that can be expressed in matrix form as in Eq. (7).

$$M_{ab}u_{i,tt}^b + (K_{aibk}u_k^b - F_i^a)\delta u_i^a = 0 \quad (7)$$

$$K_{aibk} = \int C_{ijkl} \frac{\partial N^b}{\partial x_j} \frac{\partial N^a}{\partial x_l} dV \quad (8)$$

$$F_i^a = \int \rho b_i N^a dV + \int_{S_2} t_i N^a dA \quad (9)$$

$$M_{ab} = \int \rho N^b N^a \frac{d^2u_i^b}{dt^2} \delta u_i^a dV \quad (10)$$

In this system, K is known as the element stiffness matrix, F (when multiplied by u_i^a) is a vector of applied forces, and M is the element mass matrix. K and M are precomputed based on known material properties and interpolation functions [24].

Implementation of the embedded element method

In the embedded element method, one finite element mesh is superimposed on another. The embedded mesh is constrained to follow the motion of the other mesh (the host mesh) by tying the degrees of freedom of the embedded nodes to the degrees of freedom of the host elements. Additionally, the stiffness of the embedded mesh resists the motion of the host mesh, thus creating a coupling system between the two meshes.

Tying the displacement degrees of freedom

If slip between the host and embedded nodes is not allowed, then the displacement degrees of freedom of the embedded elements can be tied to the host element motion. Displacements anywhere in the host element are interpolated from the displacement of the host element nodes via the element shape functions, usually denoted in matrix form as N . For the embedded element method, this same interpolation can be used to find the displacement of the embedded nodes. This is represented by Eq. (11).

$$\underline{u}_i = N^a_H(\underline{\xi}_i)\underline{u}_H^a \quad (11)$$

where \underline{u}_i is the displacement of the embedded node, $\underline{\xi}_i$ is the natural coordinates of the embedded node, and \underline{u}_H^a are the known displacements of the host element nodes. Figure 2 shows a two-dimensional equivalent of the scenario: the host element with nodal coordinates \underline{x}_H^a and embedded node (located at \underline{x}_i) are shown on the right as they are

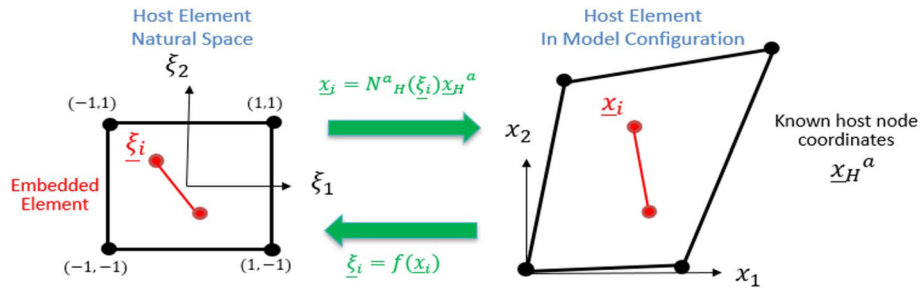


Fig. 2 Definition of variables in the mapping between host element natural space and the model global coordinates

Box 6.1 Flowchart for explicit time integration	Subroutine <i>getforce</i>
<ol style="list-style-type: none"> 1. Initial conditions and initialization: set \mathbf{v}^0, σ^0, and initial values of other material state variables; $\mathbf{d}^0 = \mathbf{0}, n = 0, t = 0$; compute \mathbf{M} 2. <i>getforce</i> 3. Compute accelerations $\mathbf{a}^n = \mathbf{M}^{-1}(\mathbf{F}^n - \mathbf{C}^{damp}\mathbf{v}^{n-1/2})$ 4. Time update: $t^{n+1} = t^n + \Delta t^{n+1/2}, t^{n+1/2} = \frac{1}{2}(t^n + t^{n+1})$ 5. First partial update nodal velocities: $\mathbf{v}^{n+1/2} = \mathbf{v}^n + (t^{n+1/2} - t^n)\mathbf{a}^n$ 6. Enforce velocity boundary conditions: if node J on Γ_v; $\mathbf{v}_J^{n+1/2} = \bar{\mathbf{v}}_J(\mathbf{x}_J, t^{n+1/2})$ 7. Update nodal displacements: $\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t^{n+1/2}\mathbf{v}^{n+1/2}$ 8. <i>getforce</i> 9. compute \mathbf{a}^{n+1} 10. Second partial update nodal velocities: $\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + (t^{n+1} - t^{n+1/2})\mathbf{a}^{n+1}$ 11. Check energy balance at time step $n+1$: see (6.2.14–18) 12. Update counter: $n \leftarrow n+1$ 13. Output; if simulation not complete, go to 4. 	<ol style="list-style-type: none"> 0. Initialization: $\mathbf{F}^0 = \mathbf{0}, \Delta t_{crit} = \infty$ 1. Compute global external nodal forces \mathbf{f}_{ext}^n 2. Loop over elements e <ol style="list-style-type: none"> i. GATHER element nodal displacements and velocities ii. $\mathbf{f}_e^{int,n} = \mathbf{0}$ iii. Loop over quadrature points ξ_Q <ol style="list-style-type: none"> 1. if $n=0$, go to 4 2. compute measures of deformation: $\mathbf{D}^{n-1/2}(\xi_Q), \mathbf{F}^n(\xi_Q), \mathbf{E}^n(\xi_Q)$ 3. compute stress $\sigma^n(\xi_Q)$ by constitutive equation 4. $\mathbf{f}_e^{int,n} \leftarrow \mathbf{f}_e^{int,n} + \mathbf{B}^T \sigma^n \bar{w}_Q J \mathbf{1}_Q$ END quadrature point loop iv. Compute external nodal forces on element, $\mathbf{f}_e^{ext,n}$ v. $\mathbf{f}_e^n = \mathbf{f}_e^{ext,n} - \mathbf{f}_e^{int,n}$ vi. Compute Δt_{crit}^e, if $\Delta t_{crit}^e < \Delta t_{crit}$ then $\Delta t_{crit} = \Delta t_{crit}^e$ vii. SCATTER \mathbf{f}_e^n to global \mathbf{F}^n 5. END loop over elements 6. $\Delta t = \alpha \Delta t_{crit}$

Fig. 3 Explicit time integration flowchart from Nonlinear Finite Elements for Continua and Structures by Belytschko et al. [27]

in the global finite element model. The natural space of the host element, which includes the nodes of the embedded element is shown on the left. ξ_i are the coordinates of the embedded node in the natural space of the host element. Equation (11) adds another constraint equation to the combined finite element model, which reduces the total degrees of freedom to only the degrees of freedom associated with the host mesh.

Interaction of forces

The explicit finite element method does not calculate stiffness directly, rather, the element stresses are calculated based on the deformation, then those stresses are used to find the internal forces on the element nodes. Figure 3 shows the general algorithm for the implementation of explicit dynamics [27].

Embedded elements contribute to the stiffness of the host elements via the combined internal forces of the embedded and host elements on the host nodes. The easiest way to incorporate this is to calculate the internal forces of the embedded elements exactly as they would be in a normal finite element program, then distribute the internal forces that are concentrated at the embedded nodes to the host element. The distribution is done using the host element shape functions which serve as weighting functions so a host node will get more force if it is closer to an embedded node and less force if it is further away. Using the element internal forces concentrated at the nodes also makes the method more versatile, since the embedded element mesh can be any size and still utilize the same distribution code. A version of *getForce_effective* that uses this method is included in Fig. 4.

FLAGSHYP - getForceEffective

1. if computing f_{net}^0 , replace “n + 1” with “0” throughout subroutine
2. Initialize: f_{net}^{n+1}
3. Update global external nodal forces f_{ext}^{n+1}
4. Loop over host elements n_h
 - a. GATHER element nodal displacements and velocities
 - b. Initialize: $f_{int,h}^{n+1}$
 - c. Loop over quadrature points n_{ζ_h}
 - i. if computing f_{net}^0 , go to (c)iv.
 - ii. Compute deformation measures $F^n(\zeta_h)$, $E^n(\zeta_h)$
 - iii. Compute stress: $\sigma^n(\zeta_h) = f(F^n(\zeta_h))$
 - iv. $f_{int,h}^{n+1} \leftarrow f_{int,h}^{n+1} + (B^T \sigma^n w_q)_\zeta$
 - d. END quadrature point loop.
 - e. Loop over all the embedded fibers n_f
 - i. if computing f_{net}^0 , go to (e)v.D.
 - ii. Compute deformation measures of the fiber: $F^n(\eta_f)$
 - iii. Calculate the stress and internal force of the fiber using the fiber material properties: $\sigma_f = f(F^n(\eta_f), \rho_f)$, $f_{int,f}^{n+1} = \sigma_f a_f$
 - iv. Calculate the correction stress and force of the fiber using the host material properties: $\sigma_{fc} = f(F^n(\eta_f), \rho_h)$, $f_{int,c}^{n+1} = \sigma_{fc} a_f$
 - v. Loop over fiber nodes $x_{f,i}$
 - A. Get fiber node coordinate $(\zeta_{x_{f,i}})$ in the iso-parametric coordinate system of the host element using Newton-Raphson algorithm.
 - B. Distribute the fiber internal force on node $x_{f,i}$ to the host nodes x_h using the values of the host element shape functions at $x_{f,i}$ as weight values $f_{int,f}^{n+1} = N(\zeta_{x_{f,i}}) f_{int,f}^{n+1}$
 - C. Distribute the correction force “on” node $x_{f,i}$ to the host nodes x_h using the values of the host element shape functions at $x_{f,i}$ as weight values $f_{int,c}^{n+1} = N(\zeta_{x_{f,i}}) f_{int,c}^{n+1}$
 - D. $f_{int,h}^{n+1} \leftarrow f_{int,h}^{n+1} + f_{int,f}^{n+1} - f_{int,c}^{n+1}$
 - 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, $f_{ext,h}^{n+1}$
 - h. Compute $f_{net,h}^{n+1} = f_{ext,h}^{n+1} - f_{int,h}^{n+1}$
 - i. SCATTER $f_{net,h}^{n+1}$ to global f_{net}^{n+1}

Fig. 4 Modified algorithm for the getforce subroutine that includes the interaction between the embedded element internal forces and the host element

Energy and the embedded element method

The classic embedded element technique superimposes an embedded mesh onto a host mesh without changing the volume of the host mesh to account for the space that is now occupied by the embedded mesh. Figure 5 shows an illustration of this. The finite element method is based on energy methods in which the internal energy of each element is calculated by an approximation of the integral of the strain energy over the element volume. Similarly, mesh mass is calculated as a volume integral of the material density. Having extra volume leads to increased mass and strain energy. This causes changes in the internal energy and the kinetic energy of the system and therefore influences the system solution [20].

The energy issues can be shown by how the model’s energy terms are calculated. The total energy terms are the sums of the energies of the host element and of the embedded element. We will refer to the host elements as the matrix elements and the embedded elements as fiber elements. The internal energy, W_{int} , Eq. (12), is found by integrating the strain energy over the volumes of the two meshes, the kinetic energy, W_{kin} , Eq. (13), is an integral of momentum times velocity over the volume, and the external work, W_{ext} , Eq. (14), is the summation of the body forces on the two meshes plus any applied traction

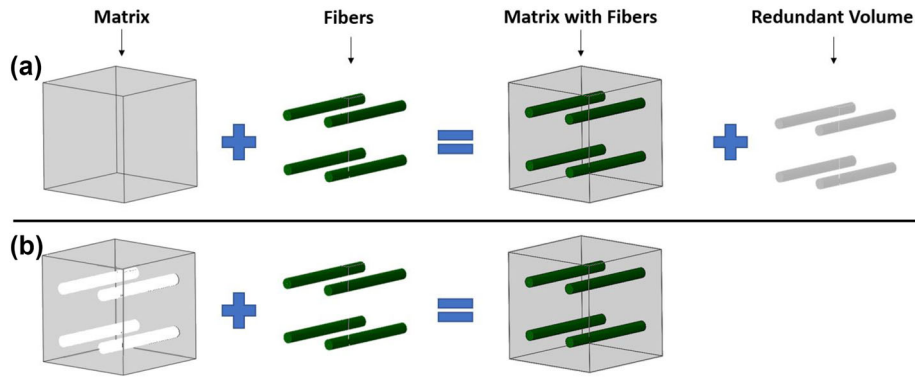


Fig. 5 **a** shows how the volume redundancy is created in the embedded element method. **b** shows how to correctly model the system

forces.

$$W_{int} = \underbrace{\int_M C_{ijkl} u_{i,j} u_{k,l} dV_M}_{\text{Matrix Strain Energy}} + \underbrace{\int_F C_{ijkl} u_{i,j} u_{k,l} dV_F}_{\text{Fiber Strain Energy}} \quad (12)$$

$$W_{kin} = \underbrace{\int_M \rho_M \frac{d^2 u_i}{dt^2} u_i dV_M}_{\text{Matrix Kinetic Energy}} + \underbrace{\int_F \rho_F \frac{d^2 u_i}{dt^2} u_i dV_F}_{\text{Fiber Kinetic Energy}} \quad (13)$$

$$W_{ext} = \underbrace{\int_M \rho_M b_i u_i dV_M}_{\text{Matrix External Load}} + \underbrace{\int_F \rho_F b_i u_i dV_F}_{\text{Fiber External Load}} + \underbrace{\int_{S_2} t_i u_i dA}_{\text{Body Force Load}} \quad (14)$$

In these equations, u_i is the displacement field of the system and C_{ijkl} is the material stiffness tensor, ρ is the material density, b_i is a vector representing body forces (such as gravity), and t_i is the vector of the surface traction (applied force per unit area) on the surface S_2 . The subscript M refers to material properties of the matrix and volume integrals over the matrix volume. Likewise, F refers to material properties of the fiber elements and volume integrals over the fiber volume.

To avoid volume redundancy, the integral over the matrix volume should not include the volume occupied by the fibers. Equations (15)–(17) include an extra term to remove the effects of the redundancy. However, in many finite element programs (e.g. Abaqus and LS-Dyna) these integrals are calculated via Gauss Quadrature, so the simplest way to account for the volume occupied by the embedded fiber is to subtract the volume integral of the matrix energy density over the fiber volume.

If the fibers and matrix have the same material properties, both integrals over the fiber volume cancel and what remains are the equations for only the matrix energy, which would be the same situation as having the matrix elements without any embedded elements.

$$W_{int} = \underbrace{\int_M C_{ijkl} u_{i,j} u_{k,l} dV_M}_{\text{Matrix Strain Energy}} - \underbrace{\int_F C_{ijkl} u_{i,j} u_{k,l} dV_F}_{\text{"Extra" Matrix Strain Energy}} + \underbrace{\int_F C_{ijkl} u_{i,j} u_{k,l} dV_F}_{\text{Fiber Strain Energy}} \quad (15)$$

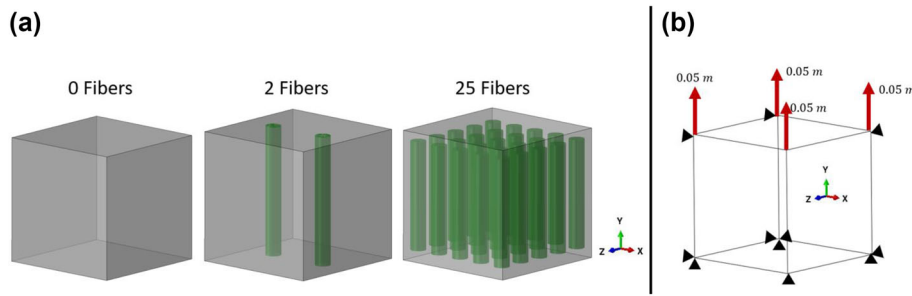


Fig. 6 **a** Unit cube models used containing 0, 2, and 25 embedded elements. **b** Applied displacement boundary conditions for the embedded element models

$$W_{kin} = \underbrace{\int_M \rho_M \frac{d^2 u_i}{dt^2} u_i dV_M}_{\text{Matrix Kinetic Energy}} - \underbrace{\int_F \rho_M \frac{d^2 u_i}{dt^2} u_i dV_F}_{\text{"Extra" Matrix Kinetic Energy}} + \underbrace{\int_F \rho_F \frac{d^2 u_i}{dt^2} u_i dV_F}_{\text{Fiber Kinetic Energy}} \quad (16)$$

$$W_{ext} = \underbrace{\int_M \rho_M b_i u_i dV_M}_{\text{Matrix External Load}} - \underbrace{\int_F \rho_M b_i u_i dV_F}_{\text{"Extra" Matrix External Load}} + \underbrace{\int_F \rho_F b_i u_i dV_F}_{\text{Fiber External Load}} + \underbrace{\int_{S_2} t_i u_i dA}_{\text{Body Force Load}} \quad (17)$$

Volume redundancy is not corrected in most commercial codes. This is acknowledged [28], leaving it up to the analyst to account for the redundant mass. When the embedded element method is used in quasi-static models, the redundancy can be addressed by reducing the density and stiffness of the host or embedded material [4, 9, 13, 17, 29]. For dynamic applications, this is unacceptable, since it will alter the wave speed of the material and any internal energy calculations that may be used in material damage models. The negative effects of volume redundancy in embedded elements can be shown by a simple test. A model of a unit cube with embedded truss elements ought to behave as if it was a single homogeneous cube if the cube and the embedded truss elements have the same material properties (i.e., the embedded element should have no effect on the model). Comparing this model with an identical cube with no embedded truss, any differences will be due to the volume redundancy. Four versions of a unit cube model were created with 0, 2, 10 or 25 embedded truss elements to represent bundles of fibers. All of the truss elements had the same cross-sectional area so increasing the number of trusses would increase their volume fraction. Twenty-five truss elements result in a fiber volume ratio of 0.5. Examples of the models are shown in Fig. 6a.

The simulations were run using the dynamic explicit solver in Abaqus with the boundary conditions shown in Fig. 6b. Abaqus was chosen as a representative commercial code because it is widely used in industry. A displacement boundary condition of 0.05 m was applied to the four nodes of the positive y face, while the nodes of the negative x, y, and z faces were pinned in those directions.

Proof of volume redundancy impact

In the initial Abaqus tests, the embedded elements and the host elements are the same material, according to the rule of mixtures there should be no difference between the stiffness and behavior of the models even as more embedded elements are added. Figure 7 shows a plot of the strain energy of each Abaqus model as a function of displacement. For

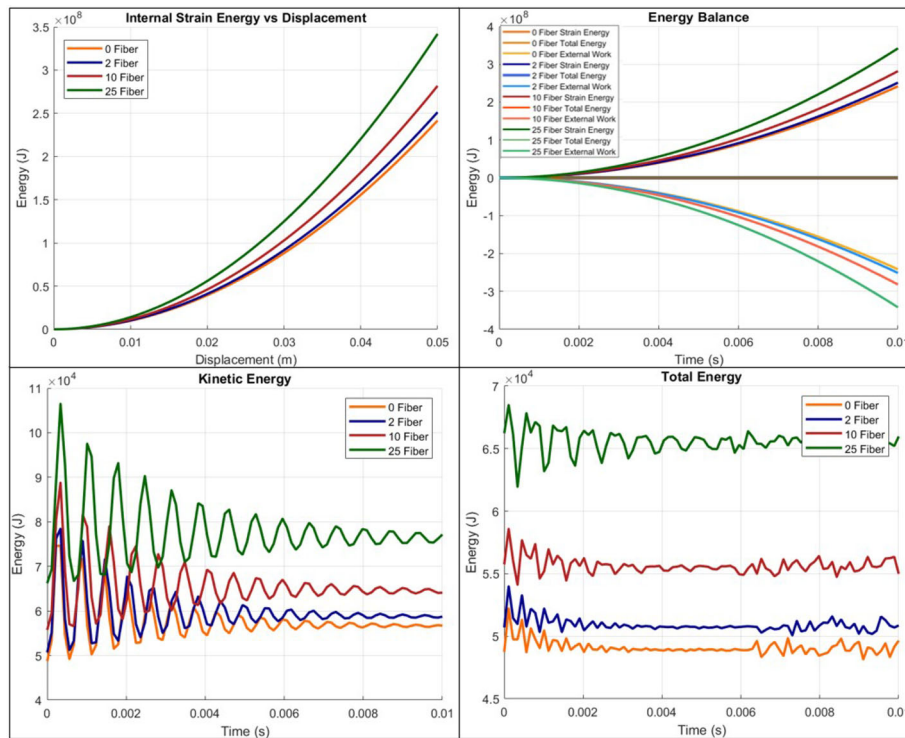


Fig. 7 Energy plots from the 0 fiber, 2 fiber, 10 fiber, and 25 fiber analyses with the applied displacement boundary condition. Although all of the models should be equivalent systems, adding embedded elements effects all types of energies

a given displacement, adding more embedded elements increases the amount of stored internal energy.

Figure 7 also shows a few other types of energy plots. The energy balance shows that in all cases energy is conserved. The difference between the models is that the total amount of energy increases with the number of embedded elements. The increase in kinetic energy with increase in fiber elements shows that the redundancy in the mass matrix calculation does have an effect. The applied displacement ensures that all three models will have the same velocity as a function of time. Therefore, the only difference in their kinetic energies is due to their differences in mass. In a low strain rate analysis such as this, incorrect kinetic energies have little effect on the final result, but might be a significant problem in problems with large velocities or high strain rates.

Method of volume redundancy correction

Garimella [20] suggested a way to correct the volume redundancy in the algorithm for dynamic explicit finite element analysis. The implementation of Garimella’s method, shown in Fig. 8, was modified to make the incorporation into the embedded element method easier. While Garimella used a calculation of the embedded element internal force in the host element’s natural coordinate system to find the amount of force needed to correct for the internal energy of the redundant volume, we chose to use a method similar to the force interaction method described earlier. Here, the internal force of the redundant volume is calculated as if it were a truss element made of the host element material as the embedded element is oriented in the global coordinate system. This results in

Box 3. Subroutine - *getForceEffective*

1. if computing f_{net}^0 , replace “ $n + 1$ ” with “0” throughout subroutine
2. Initialize: f_{net}^{n+1}
3. Update global external nodal forces f_{ext}^{n+1}
4. Loop over elements n_e
 - a. GATHER element nodal displacements and velocities
 - b. Initialize: $f_{int,e}^{n+1}$
 - c. Loop over quadrature points n_ζ
 - i. if computing f_{net}^0 , go to (c)iv.
 - ii. Compute deformation measures $F^n(\zeta)$, $E^n(\zeta)$
 - iii. Compute stress: $\sigma^n(\zeta) = f(F^n(\zeta))$
 - iv. $f_{int,e}^{n+1} \leftarrow f_{int,e}^{n+1} + (B^T \sigma^n \bar{w}_q)_\zeta$
 - d. END quadrature point loop.
 - e. Loop over all the embedded fibers n_f
 - i. if computing f_{net}^0 , go to (e)iii.E.
 - ii. Calculate the displacements of the embedded nodes from the host nodal displacements:
 $u_{embed} = [N]u_{host}$. Shape function matrix is calculated at the embedded nodes.
 - iii. Loop over the quadrature points of the fiber n_{ζ_f}
 - A. Calculate the fiber quadrature point coordinates (ζ_f) in the iso-parametric coordinate system of the host element using Newton-Raphson algorithm.
 - B. Compute deformation measures at these fiber quadrature points: $F^n(\zeta_f)$.
 - C. Calculate the fiber stress measure in the host coordinate system:
using the fiber material model.
 - D. Calculate the correction stress measure in the host coordinate system: $\sigma^n(\zeta_f) = f(F^n(\zeta_f))$ using the host element material model.
 - E. $f_{int,e}^{n+1} \leftarrow f_{int,e}^{n+1} + (B^T \sigma_f^n \bar{w}_{f,q} J_f - B^T \sigma_c^n \bar{w}_{f,q} J_f)_{\zeta_f}$ 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. $f_{ext,e}^{n+1}$
 - h. Compute $f_{net,e}^{n+1} = f_{ext,e}^{n+1} - f_{int,e}^{n+1}$
 - i. SCATTER $f_{net,e}^{n+1}$ to global f_{net}^{n+1}

Fig. 8 Algorithm suggested by Garimella [20] to address internal energy redundancy

the correction force which is distributed to the host nodes using the host element shape functions, the same way as the embedded element force. By using the standard calculation for the internal force of the embedded element in the global coordinate system, we avoid any confusion in attempting to convert the one-dimensional axial stress of a truss element to a stress measure that would need to exist in three dimensions. Additionally, the accuracy of the force transfer is now directly correlated with the embedded element mesh density. Increasing the mesh density of the embedded nodes will correlate with approaching the true, non-discretized, constraint condition between the fibers and the mesh where they are bonded along the entire length of the fiber. The differences between Garimella’s [20] method and the method used here are shown in the flowchart of Fig. 4 in step 4.a.e.v.A–D and Fig. 8 steps 4.e.i–iv.

The mass correction was carried out as shown in the flowchart of Fig. 9, this is the same algorithm as used in Garimella [20]. This only requires calculating the mass of the redundant volume by multiplying the volume of the embedded mesh by the density of the host mesh. This mass is then subtracted from the total mass of the host element before it is scattered to the lumped mass matrix.

To assess the accuracy of this correction, it was incorporated into a dynamic explicit code implemented in Matlab. This code is a modified version of Bonet et al.’s [30] FLagSHyP code that was written to go along with their textbook. When this work started only the static implicit version of FLagSHyP was available, so a dynamic explicit solver modified to include the volume redundancy correction was added. The algorithm suggested by

Subroutine - getMassEffective

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 using length l_f and cross-sectional area a_f
 - ii. Calculate the fiber mass: $m_e = \rho_f l_f a_f$
 - 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 nodal mass vector
 - f. Use the nodal mass vector to form the element mass matrix M_e (diagonal mass matrix)
 - g. Scatter mass matrix to global mass matrix M
2. END loop over the elements n_e

Fig. 9 Algorithm suggested by Garimella [20] to address mass redundancy

Garimella was implemented so that the user can choose to use the correction if desired. The resulting FE code will be referred to as FLagSHyP Modified (FM). When the volume correction feature is in use, the code will be referred to as FM Corrected. This code can be found on GitHub at https://github.com/Valerie96/flagshyp/tree/Flagshyp2_element_types.

Results**Verification of FLagSHyP Modified**

To verify FLagSHyP Modified, a single continuum element in tension was compared with an identical element but with an embedded truss. The truss had the same material properties as the host element, and so should have no effect on the total energy or stiffness. This test case was also run in Abaqus. Since the truss element and the host element have the same material properties, they should produce exactly the same response. As shown previously, this is not the case for the Abaqus models. However, the results in Fig. 10 show that with the volume redundancy correction algorithm turned on, FM was able to reproduce the results for a single solid element when the truss and the host materials were identical.

Specific effects of redundancy

To explore the amount of difference the volume redundancy makes in different cases, several more applied displacement models were run to create plots of energy vs. fiber volume fraction and energy vs. strain rate. The redundant volume should affect the internal and kinetic energies. With extra volume the model can store more internal energy for a given displacement. The extra volume also comes with extra mass, which will show up as increased kinetic energy. As the fiber volume fraction increases, the amount of redundant volume increases and should make these energy differences larger. Increasing the strain rate will increase the nodal velocities. In this case, the strain energy should not be changing but the kinetic energy, which depends on velocity, will increase.

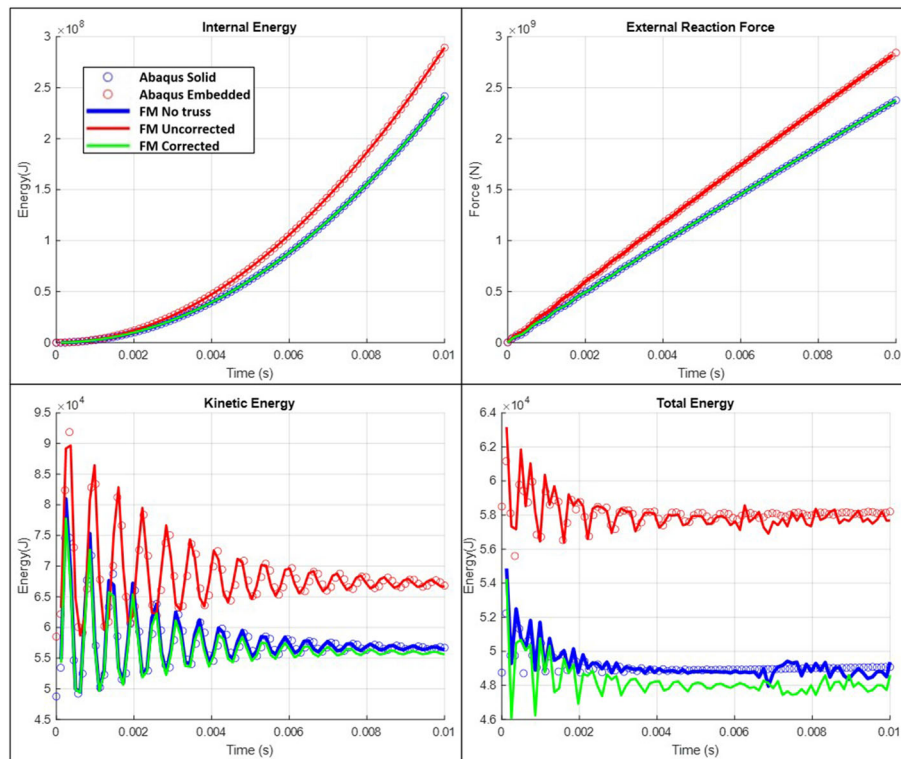


Fig. 10 Raw energy data comparison from FLagSHyP Modified verification tests. A solid element is compared with an identical model with the addition of an embedded element. The embedded element was assigned the same material properties as the host element so that the system equivalent behavior should be the same as the solid element model. The volume redundancy in the embedded element method causes increases in internal energy, stiffness, and kinetic energy. The correction method shown is able to remove this extra energy and returns the same results as a solid cube

Fiber volume fraction effects

The tests on fiber volume fraction involved running the displacement simulation in both FM and Abaqus with zero, two, ten, and twenty-five embedded truss elements, where the host and embedded elements had the same material type. The truss elements all had the same cross-sectional area so adding more trusses increased their volume fraction. Additionally, the embedded element simulations were rerun in FM, but this time with the volume redundancy correction turned on. Figure 12 shows a schematic of the set up and Table 1 has the masses and details about the applied displacement and strain rate for each of the test cases. To compare the effect of the embedded elements, the internal and kinetic energy of each model was compared with the solid element model to get a percent difference in energy. Abaqus embedded element models were compared to the Abaqus solid element model and FM embedded element models were compared to the FM solid element model. With the embedded elements having the same material as the host, they should all be equivalent to a solid element, which is why the zero truss case is used as the reference. These comparisons resulted in a relative difference, or error introduced by added the embedded elements which is shown in the plots in Fig. 11. As expected, as the fiber volume fraction increased, both the total internal kinetic and internal energies increased as well as their relative error. The exception to this was the models using FM Corrected. As shown in Fig. 11, the correction nearly eliminated the error in internal

Table 1 Fiber volume fraction dependency tests

	Truss elements	Volume fraction	Total mass (kg)	Applied displacement (m)	Time increment (s)	Strain rate (1/s)
Abaqus	0	0	7800	0.05	0.01	5
	2	0.04	8112	0.05	0.01	5
	10	0.2	9360	0.05	0.01	5
	25	0.5	11,700	0.05	0.01	5
FM	0	0	7800	0.05	0.01	5
	2	0.04	8112	0.05	0.01	5
	10	0.2	9360	0.05	0.01	5
	25	0.5	11,700	0.05	0.01	5
FM Corrected	2	0.04	7800	0.05	0.01	5
	10	0.2	7800	0.05	0.01	5
	25	0.5	7800	0.05	0.01	5

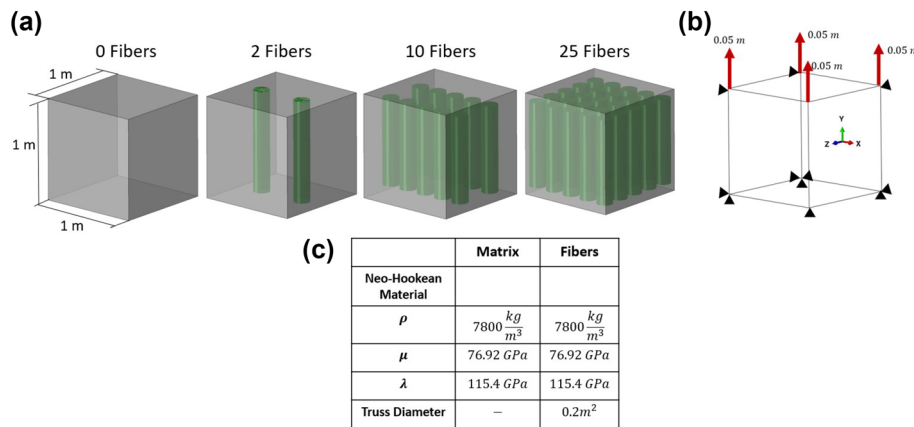


Fig. 11 Comparison of internal and kinetic energy from Abaqus and FM embedded element models with different fiber volume fractions with reference to the energy of solid element cases with no embedded elements. Increasing the fiber volume fraction increases the amount of volume redundancy and increases the error in the system. FM Corrected drastically reduces the errors in energy

energy and drastically reduced the kinetic energy error. Considering that the uncorrected FM models almost exactly match the data from Abaqus, this shows that the volume redundancy is causing error, we know where it is, and we know how to correct it.

Strain rate (velocity effects)

A second set of tests was run to check the dependency of energy on strain rate. These consisted of an applied displacement boundary condition. The total displacement in each case was the same, but the speed at which it was applied was changed to approximate strain rates of 5, 10, 50, and 200 1/s. A reference set of models at each of these strain rates was run in both Abaqus and FM using a solid hex element. Each test was then repeated with 25 embedded truss elements of the same material type as the host element. Finally, the 25 truss element models were repeated again in FM Corrected. Figure 13 shows a schematic of the set up and Table 2 has the masses and details about the applied displacement and strain rate for each of the test cases.

Internal energy was expected to remain constant since the total displacement was unchanged. Kinetic energy was expected to increase due to the increase in applied veloc-

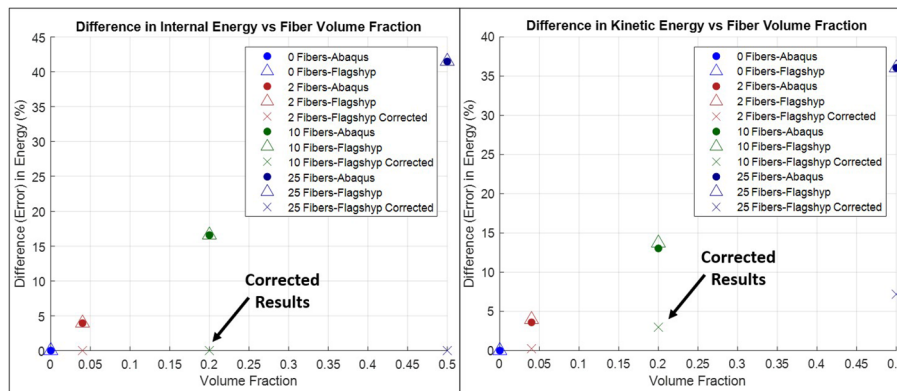


Fig. 12 Set up of the fiber volume fraction dependency tests, **a** images of 0, 2, 10, and 25 fiber meshes, **b** test boundary conditions, **c** matrix and fiber material properties

Table 2 Strain rate dependency tests

	Truss elements	Volume fraction	Total mass (kg)	Applied displacement (m)	Time increment (s)	Strain rate (1/s)
Abaqus	0	0	7800	0.05	0.01	5
		0	7800	0.05	0.002	25
		0	7800	0.05	0.001	50
		0	7800	0.05	0.00025	200
	25	0.5	11,700	0.05	0.01	5
		0.5	11,700	0.05	0.002	25
		0.5	11,700	0.05	0.001	50
		0.5	11,700	0.05	0.00025	200
FM	0	0	7800	0.05	0.01	5
		0	7800	0.05	0.002	25
		0	7800	0.05	0.001	50
		0	7800	0.05	0.00025	200
	25	0.5	11,700	0.05	0.01	5
		0.5	11,700	0.05	0.002	25
		0.5	11,700	0.05	0.001	50
		0.5	11,700	0.05	0.00025	200
FM Corrected	25	0.5	7800	0.05	0.01	5
		0.5	7800	0.05	0.002	25
		0.5	7800	0.05	0.001	50
		0.5	7800	0.05	0.00025	200

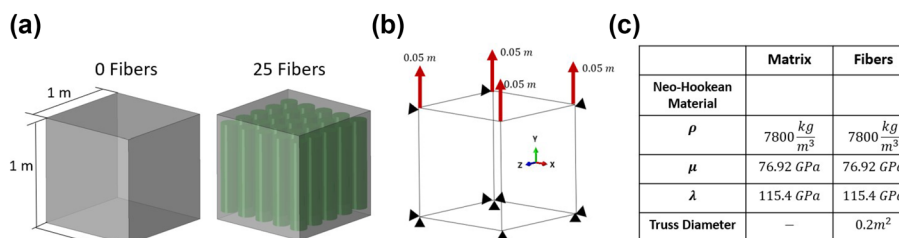


Fig. 13 Set up of the strain rate dependency tests, **a** images of 0 and 25 fiber meshes, **b** test boundary conditions, **c** matrix and fiber material properties

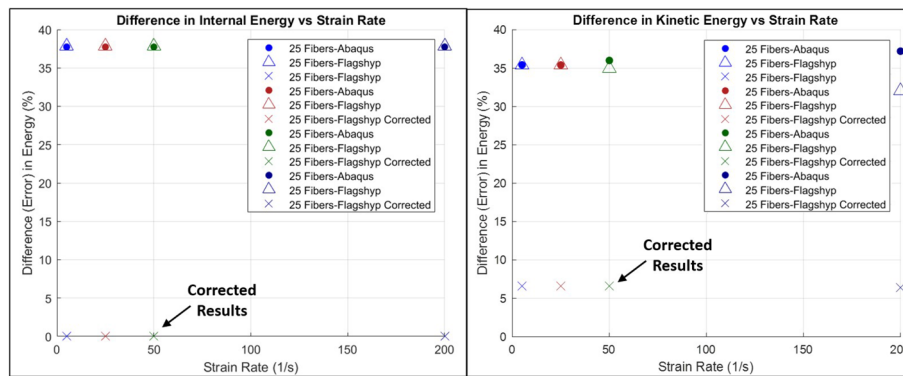


Fig. 14 Comparison of internal and kinetic energy from Abaqus and FM models with twenty-five embedded elements with reference to the energy of a solid element with no embedded elements. The additional strain energy from the volume redundancy is constant with changes in strain rate as expected and is easily corrected for using FM Corrected. The volume redundancy also adds extra mass which ought to increase the error as the nodal velocities increase with the increasing applied strain rate. This effect occurs at very high strain rates. FM Corrected is able to bring the error in energy down to less than 10% for all velocities

ity. Similar to the volume fraction cases, each embedded element model was compared with its' equivalent solid element at the same strain rate. These comparisons resulted in a relative difference, or error introduced by added the embedded elements which is shown in the plots in Fig. 14. Again, FM Corrected was able to completely correct for the error in the internal energy. The relative error in kinetic energy was expected to increase as the strain rate and nodal velocity increased, yet the error remained constant.

Discussion

Energy implications

The volume redundancy is a well-known short coming and is addressed by many authors by modifying the material properties of the host material. Using this volume correction algorithm is a much more robust method because it does not require the estimation of an “effective medium.” When the stiffness of the host material is modified it complicates the stress calculation. This becomes critical where stress wave propagation is important. With our method the true material properties are used and stress calculations are not affected. Correcting the energy terms in the code itself provides a model that more accurately represents the physics of the problem instead smearing the different materials together into a representative material. The modification also ensures the correct calculation of both internal and kinetic energies which become more important as strain rate and fiber volume fraction increase.

Volume fraction effects

The volume fraction of truss element to host element volume has a large impact on the energy terms, since it adds both extra mass and the potential to store more internal energy. Our code is able to eliminate almost all of the error in the internal energy due to the volume redundancy without modification to the material properties themselves. In these tests, internal energy accounts for most of the energy in the system and is important to have accurate in applications involving stress wave propagation. The redundancy correction also reduced the error in kinetic energy to under 10 percent for fiber volume fractions up

to 0.5. Based on the trends shown here, the error in the corrected version will continue to increase with volume fraction. This might only be because, as shown in Figs. 7 and 14, the kinetic energy in these small models oscillates a lot which is a result of small over corrections by the explicit time stepping method, which damps out over time. The kinetic energies used in comparisons here are just the kinetic energy reported at the end of the last time increment so the oscillation in each model adds to the error. This needs to be tested on models with larger numbers of host elements, which will be done in the future.

Strain rate effects

For the strain rate tests, since the amount of volume redundancy was kept constant, the strain rate was expected to impact the kinetic energy by increasing the nodal velocities while having no effect on total internal energy stored since the total displacement remained constant. The effect of strain rate on the error in kinetic energy was small although it did increase slightly with increasing strain rate as expected. Only four different strain rates were tested with the maximum as only 200 1/s. Much larger strain rates were not possible for the small tests (due to errors caused by the coarse model discretization) but could result in increasing error. In the tested cases the corrected code was able to reduce the error in the kinetic energy to below 10%. In these cases the kinetic energy is several orders of magnitude smaller than the internal energy, so having it exactly corrected is less important. Future work will include tests on larger models with more nodes and higher velocities to determine if, or when, the kinetic energy becomes significant.

Limitations of the embedded element method

The volume redundancy code is still in early stages and it has not yet been compared to experimental results with real fiber reinforced material to prove it has better experimental predictions than the standard embedded element method. Another obstacle is that researchers cannot implement this code as a user defined element type or subroutine in commercial codes since it is a modification to the solution process, not the material behavior. The idea is to modify mass and internal energy calculations directly to create a model that more closely matches the physics. We plan to add the volume redundancy calculation to a C based finite element code that will be more efficient than the current Matlab code. It has been brought to our attention before that there may be an issue with the “no slip” condition we propose between the embedded truss elements and the matrix, due to the displacement ties only being located at the embedded node locations. This issue will likely become more apparent as larger strains and plastic deformations are introduced. We expect that as the mesh density of the embedded elements increases, increasing the number of embedded nodes, the relative slipping between the embedded truss elements and matrix elements will decrease. This will be examined in future work.

Conclusion

It has been shown how incorrect volume and mass effect the energy calculations in the embedded element method, which in turn effects the analysis solution. Leaving this uncorrected in commercial codes can cause erroneous results especially in cases with wave propagation. A simple test of a single host element with embedded elements using the same material properties was created and compared to the solution for the same contin-

uum element with no embedded elements. Since the embedded elements were the same material as the continuum element the effective system is equivalent to having no embedded elements at all. Due to the volume redundancies in the calculation, the model with embedded elements could store more internal energy. The volume redundancy also added mass and increases the kinetic energy of the system. The differences in internal energy between the embedded element model and the control continuum element were shown to be accounted for by the additional matrix volume in the calculation. A modified explicit finite element code was shown to be able to correct the effects of the volume redundancy for small models. This technique could be extended to model larger objects with embedded elements, particularly fiber reinforced composites. Next steps for this research should include larger models with different loading types and comparisons against experimental data to determine if this corrected version of the embedded element method will perform better than the standard version in predicting the response of dynamic events in fiber reinforced composites.

Abbreviations

FEA Finite element analysis
FEM Finite element method
FM FLagSHyP modified

Author contributions

VM contributed to the design of the work, the acquisition, analysis, interpretation of data, the creation of new software used in the work, and drafted and revised the paper RK contributed to the conception and design of the work analysis, and draft revision. TH contributed to the draft revision SE contributed to the conception and design of the work. All authors read and approved the final manuscript.

Funding

This work was supported by Triad National Security, LLC which operates Los Alamos National Laboratory under contract 293402. In addition, the authors gratefully acknowledge the support of the Institute for Computational and Data Sciences at the Pennsylvania State University. R.H.K. was partially supported by the National Science Foundation CAREER award under award number 1846059. Any opinions, findings and conclusions expressed in this article are those of the authors and do not necessarily reflect the views of by Penn State University, Triad National Security, LLC, Los Alamos National Laboratory, or the National Science Foundation.

Availability of data and materials

The source code for FLagSHyP Modified can be found at: https://github.com/Valerie96/flagshyp/tree/Flagshyp2_element_types.

Declarations

Competing interests

Reuben Kraft has a financial interest in BrainSim Technologies Inc., a company which could potentially benefit from the results of this research. This interest has been reviewed by Penn State University in accordance with its Individual Conflict of interest policy for the purpose of maintaining the objectivity and integrity in research and is being managed.

Received: 19 January 2022 Accepted: 8 June 2022

Published online: 02 July 2022

References

1. Belytschko T, Fish J, Engelmann BE. A finite element with embedded localization zones. *Computer Methods Appl Mech Eng.* 1988;70(1):59–89.
2. Fish J, Belytschko T. Elements with embedded localization zones for large deformation problems. *Computers Struct.* 1988;78:10.
3. Huynh DBP, Belytschko T. The extended finite element method for fracture in composite materials. *Int J Numer Methods Eng.* 2009;77(2):214–39. <https://doi.org/10.1002/nme.2411>.
4. Jiang WG, Hallett SR, Wisnom MR. Development of Domain Superposition Technique for the Modelling of Woven Fabric Composites. *Computational Methods Appl Sci.* 2008;10:281–91. https://doi.org/10.1007/978-1-4020-8584-0_14.
5. Takano N, Uetsuji Y, Kashiwagi Y, Zako M. Hierarchical modelling of textile composite materials and structures by the homogenization method. *Model Simul Mater Sci Eng.* 1999;7(2):207–31. <https://doi.org/10.1088/0965-0393/7/2/006>.
6. Fish J. The s-version of the finite element method. *Computers Struct.* 1992;43(3):539–47.
7. Fish J, Markolefas S. The s-version of the finite element method for multilayer laminates. *Int J Numer Methods Eng.* 1992;33(5):1081–105. <https://doi.org/10.1002/nme.1620330512>.

8. Zako M, Uetsuj Y, Kurashiki T. Finite element analysis of damaged woven fabric composite materials. *Composit Sci Technol*. 2003;63(3–4):507–16.
9. Ohyama D, Kurashiki T, Watanabe Y, Fujita Y, Zako M. ESTIMATION OF MECHANICAL BEHAVIOR OF BRAIDED COMPOSITES BASED ON MESH SUPERPOSITION METHOD; 2011. p. 6.
10. larve EV, Mollenhauer DH, Zhou EG, Breitzman T, Whitney TJ. Independent mesh method-based prediction of local and volume average fields in textile composites. *Composites*. 2009;40(12):1880–90.
11. Cox BN, Carter WC, Fleck NA. A binary model of textile composites-I. Formulation *Acta Metallurgica et Materialia*. 1994;42(10):3463–79.
12. Yang Q, Cox B. Spatially Averaged Local Strains in Textile Composites Via the Binary Model Formulation. *J Eng Mater Technol*. 2003;125(4):418–25.
13. Heru Utomo BD. High-speed Impact Modelling and Testing of Dyneema Composite, PhD Technische Universiteit Delft; 2011.
14. Vanalli L, Paccola RR, Coda HB. A simple way to introduce fibers into FEM models. *Commun Numer Methods Eng* 2007;24(7):585–603 oi 10.1002/cnm.983
15. Garg AK, Abolmaali A. Finite-Element Modeling and Analysis of Reinforced Concrete Box Culverts. *J Transport Eng*. 2009;135(3):121–8.
16. Kang J, Bolander JE. Event-based lattice modeling of strain-hardening cementitious composites. *Int J Fract*. 2017;206(2):245–61. <https://doi.org/10.1007/s10704-017-0214-2>.
17. Häussler-Combe U, Shehni A, Chihadeh A. Finite element modeling of fiber reinforced cement composites using strong discontinuity approach with explicit representation of fibers. *Int J Solids Struct*. 2020;200–201:213–30.
18. Pan Y, Sullivan D, Shreiber DJ, Pelegri AA. Finite Element Modeling of CNS White Matter Kinematics: Use of a 3D RVE to Determine Material Properties. *Front Bioeng Biotechnol*. 2013. <https://doi.org/10.3389/fbioe.2013.00019/abstract>.
19. Yousefsani SA, Shamloo A, Farahmand F. Micromechanics of brain white matter tissue: A fiber-reinforced hyperelastic model using embedded element technique. *J Mech Behav Biomed Mater*. 2018;80:194–202.
20. Garimella HT, Menghani RR, Gerber JI, Sridhar S, Kraft RH. Embedded Finite Elements for Modeling Axonal Injury. *Ann Biomed Eng*. 2019;47(9):1889–907. <https://doi.org/10.1007/s10439-018-02166-0>.
21. Tabatabaei SA, Lomov SV. Eliminating the volume redundancy of embedded elements and yarn interpenetrations in meso-finite element modelling of textile composites. *Computers Struct*. 2015;152:142–54.
22. Chowdhury NT, Joosten MW, Pearce GMK. An embedded meshing technique (SET) for analysing local strain distributions in textile composites. *Composite Structures*. 2019;210:294–309.
23. Radtke FKF, Simone A, Sluys LJ. A partition of unity finite element method for obtaining elastic properties of continua with embedded thin fibres. *Int J Numer Methods Eng*. 2010;84(6):708–32.
24. Bower A. In: Theory and Implementation of the Finite Element Method: Generalized FEM for Static Linear Elasticity; 2010. http://solidmechanics.org/Text/Chapter8_1/Chapter8_1.php#Sect8_1_1.
25. Bower A. In: Theory and Implementation of the Finite Element Method: The Finite Element Method for Dynamic Linear Elasticity; 2010. http://solidmechanics.org/Text/Chapter8_2/Chapter8_2.php.
26. Bower A. In: Work Done by Stresses; Principle of Virtual Work; 2010. http://solidmechanics.org/Text/Chapter2_4/Chapter2_4.php.
27. Belytschko T, Liu WK, Moran B, Elkhodary KI. *Nonlinear Finite Elements for Continua and Structures*. New York: Wiley; 2014.
28. Systèmes D. Abaqus 2020 Documentation. Abaqus/Constraints/Embedded Elements. Dassault Systèmes; 2020. https://help.3ds.com/2020/english/DSSIMULIA_Established/SIMACAECSRefMap/HelpViewerDS.aspx?version=2020&prod=DSSIMULIA_Established&lang=english&path=SIMACAECSRefMap%2fsimacst-c-embeddelement.htm&ContextScope=all&id=467a17ba7465488d89073dbf5af93ef4.
29. Liu BG, Kandan K, Wadley HNG, Deshpande VS. High strain rate compressive response of ultra-high molecular weight polyethylene fibre composites. *Int J Plasticity*. 2019;122:115–34.
30. Bonet J, Gil AJ, Wood RD. *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*. Cambridge: Cambridge University Press; 2016.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.