

The Pennsylvania State University

The Graduate School

**MULTISCALE MODELING OF FIBER REINFORCED MATERIAL USING THE
EMBEDDED FINITE ELEMENT METHOD**

A Dissertation in
Mechanical Engineering

by

Valerie Martin

© 2023 Valerie Martin

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2023

The dissertation of Valerie Martin was reviewed and approved by the following:

Reuben Kraft
Associate Professor of Mechanical and Biomedical Engineering
Thesis Advisor
Chair of Committee

Francesco Costanzo
Professor in the Department of Mechanical Engineering and the Department of
Engineering Science and Mechanics

Parisa Shokouhi
Professor in the Department of Engineering Science and Mechanics

Charles E. Bakis
Distinguished Professor of Engineering Science and Mechanics

Robert J. Kunz
Professor of the Department of Mechanical Engineering
Associate Department Head for Graduate Studies

ABSTRACT

The purpose of this dissertation is to investigate an efficient finite element model for fiber reinforced composites to use in the design of armors with complex shapes, incorporating multi-scale modeling, high strain rate behavior, and damage by utilizing the embedded element method. Particularly, the focus is on modeling Dyneema®, a composite consisting of Ultra High Molecular Weight Polyethylene (UHMWPE) fibers set in a polymer matrix. The embedded element method is used to model bundles of fibers as truss elements embedded in the matrix material modeled by standard continuum elements. The truss elements can support both tensile and compressive loads, but not buckling. A large part of this effort involves modifying the embedded element method to make it more conducive to modeling such composites with large fiber volume fractions. To this end, an central difference finite element code is introduced that is programmed to correct the volume redundancy in the embedded element method on an algorithmic level. It is shown that this code is accurate when compared to a commercial code and correctly removes the volume redundancy. Additionally, the embedded element method is implemented in the commercial code Abaqus to model ballistic impact test. The method is shown to be able to capture plate back face deformation, delamination, indirect tension failure, and fiber snap back.

The main research questions can be summarized as follows:

1. Can a truss element mesh be easily created for curved or complex geometry?
2. Can the embedded element method be modified to correct the volume redundancy and eliminate the need for material property smearing?

3. Does using embedded truss elements to model fiber bundles provide accurate predictions of material behavior on small scales (tension/compression tests) and larger scales (full plate impact)?

TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES.....	xiii
ACKNOWLEDGEMENTS	xiv
Chapter 1 Literature Review and Background Information	1
1.1 Fiber Reinforced Composites	1
1.1.1 Description.....	1
1.1.2 Uses in Armor.....	2
1.1.3 Curved Plates	4
1.2 History of the Embedded Element Method.....	6
1.3 Finite Element Modeling of Fiber Reinforced Composites.....	9
1.4 Thesis Statement.....	11
Chapter 2 MATLAB Script for Embedded Fiber Creation.....	12
2.1 Flat Plate.....	12
2.2 Curved Plates	15
2.3 Limitations	17
Chapter 3 Embedded Element Finite Element Code.....	19
3.1 Addition of Neo-Hookean Truss Elements.....	20
3.2 Addition of Embedded Element Method.....	21
3.2.1 Tying the Displacement Degrees of Freedom	23
3.2.2 Interaction of Forces.....	25
3.3 Volume Redundancy Correction	28
3.3.1 Numerics of Additional Mass, Internal Force, Kinetic Energy	28
3.3.2 Proof of Volume Redundancy	30
3.3.3 Method of Volume Redundancy Correction.....	31
3.4 Verification of FLaGSHyP Modified.....	32
3.5 Proof of Volume Redundancy Impact.....	34
3.5.1 Specific Effects of Redundancy	36
3.6 Validation with Multiple Host Elements.....	43
3.7 Tension/Compression Comparison to Experiment.....	44
3.7.1 Literature Data	45
3.7.2 Embedding Elements.....	46
3.7.4 Material Properties.....	48
3.7.5 Results and Discussion	51

3.8 Speed Updates	55
3.9 Conclusion	59
Chapter 4 Finite Element Impact Models	61
4.1 Steel Plate Models	61
4.1.1 Mesh Convergence	63
4.1.2 Mild Steel Plates	67
4.1.3 Element Erosion	72
4.2 Embedded Element Modeling	73
4.2.1 Initial Tests	73
4.2.2 Karthikeyan Plate Impact	77
4.3 Using the Embedded Element Method to Simulate Full Impact	79
4.3.1 Methods	80
Plate impact experiments	80
CT scans and results	81
4.3.2 Results	89
4.3.3 Discussion	96
4.4 Curved Plate	98
4.5 Conclusions	102
Chapter 5 Conclusions and Future Work	104
5.1 Conclusions	104
5.1.1 MATLAB Script for Embedded Fiber Creation	104
5.1.2 Embedded Element Finite Element Code	105
5.1.3 Embedded Element Finite Element Code and Finite Element Impact Models	105
5.2 Future Work	107
5.2.1 Volume Redundancy Guidelines Paper Concept	107
5.2.2 Move FFlagSHyP Modified to a Faster Language	109
5.2.3 More Experimental Validation	109
References	112
Appendix A Derivation of a One-Dimensional Neo-Hookean Truss Element Material Model	118
Appendix B Modified getForceEffective Subroutine	122
Appendix C Mild Steel: Johnson-Cook Material Properties	123
Appendix D	
Information on the GitHub Repository for FFlagSHyP Modified (FM)	124

LIST OF FIGURES

Figure 1- 1: Schematic of $0^\circ/90^\circ$ oriented layers of fibers in a composite, from Liu et al.[7].....	2
Figure 1- 2: Images of the darted pinwheel pattern and the compression press used to mold fiber reinforced composite helmets from Li et al. [1].....	5
Figure 1- 3: The different size scales of modeling fiber reinforced composites	10
Figure 2- 1: a) Fibers defined in MATLAB based on the cylinder cross section. b) Cross section of the cylinder model imported into Abaqus from the MATLAB output. c) Whole cylinder part as imported into Abaqus, with truss elements in a $0/90^\circ$ orientation	13
Figure 2- 2: Progression of creating embedded truss elements in a cylinder-shaped part. ...	14
Figure 2- 3: The MATLAB script can create truss element definitions for complex shapes.....	14
Figure 2- 4: Illustration of the Curved – Flat – Curved mapping process to include truss element in a curved plate.....	15
Figure 2- 5: An illustration of the process to interpolate between the two radii curved plate arcs and the calculation of z coordinates in the curved space from the flattened space	16
Figure 2- 6: A curve plate created and meshed in Abaqus along with a rendering of the embedded truss elements added by the MATLAB script	17
Figure 3- 1: Effects of damping showed on a four element cylinder in tension. a) With no damping, the model is very unstable. b) Damping added to the cylinder model, but the damping coefficient is too high, and the system is still unstable. c) With the damping applied, the wild oscillations in the stress are eliminated.....	20
Figure 3- 2: Stress results from Abaqus and FFlagSHyP for a neo-Hookean truss element in tension using some contrived material types. Several different materials were tested to show that the final version of FFlagSHyP works for different properties.	21
Figure 3- 3: Illustration of a fiber structure embedded in matrix material.	22
Figure 3- 4: Relation between host element, embedded node, and natural space, and the algorithm for Newton-Rapson method used to approximate an inverse relation.	25

Figure 3- 5: Explicit time integration flowchart from Nonlinear Finite Elements for Continua and Structures by Belytschko et al. [64].	26
Figure 3- 6: Subroutine for internal force calculation from Garimella et al. [40].	26
Figure 3- 7: Calculation of the host element internal force vector with the influence of embedded elements and volume redundancy correction.	27
Figure 3- 8: a) How volume redundancy is created in the embedded element method. b) How the system should be correctly modeled.	28
Figure 3- 9: a) Unit cube models containing 0, 2, and 25 embedded elements. b) Applied displacement boundary conditions for the embedded element models.	31
Figure 3- 10: Algorithm suggested by Garimella et al. [40] to address mass and force redundancy.	32
Figure 3- 11: Raw energy data comparison from FFlagSHyP 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 can remove this extra energy and returns the same results as a solid cube.	34
Figure 3- 12: Internal strain energy plotted with displacement of the 0 fiber, 2 fiber, 10 fiber, and 25 fiber models with an applied displacement boundary condition. Although all the models should be equivalent systems, adding embedded elements increases the amount of internal strain energy.	35
Figure 3- 13: Energy plots from the 0 fiber, 2 fiber, 10 fiber, and 25 fiber analyses with the applied displacement boundary condition. The addition of embedded elements affects all types of energies.	36
Figure 3- 14: 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.	39
Figure 3- 15: 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 high strain rates. FM Corrected can bring the error in energy down to less than 10% for all velocities.....	42
Figure 3- 16: a) 8 hexahedral elements with the 4 embedded truss elements b) Abaqus and FFlagSHyP results for the 8 host and 4 embedded element models for tension and compression tests.....	44
Figure 3- 17: Finite element models with boundary conditions generated for this study....	45
Figure 3- 18: Definition of variables in the mapping between host element natural space and the model global coordinates.	47
Figure 3- 19: Examples of truss element stacking in the tension test model. A) shows the matrix material mesh used for all embedded element models. The truss meshes were made by assigning b) 5,000, c) 1,000, and d) 100 fibers per truss. The cross-sectional areas reflect the equivalent cross-sectional area of the total number of fibers.	48
Figure 3- 20: Overall model response of the embedded finite element models for the tension and compression tests when a consistent truss element size and varying the truss and matrix material behavior.	50
Figure 3- 21: Different packing arrangements for embedded truss elements in a cube of material. The original arrangement, used in this work, only gives a fiber volume fraction of 52.9%, which is much less than the typical 83% found in Dyneema®. By reducing the buffer space between the trusses, and by changing the stacking arrangement to $[0^\circ/90^\circ]$ increases the fiber volume fraction to 78.0% and 92.0%, respectively. One of these latter two packing arrangements will be used in future work.	52
Figure 3- 22 : When the truss and matrix element material is kept constant, changing the cross-sectional area and number of truss elements does not significantly affect the model response if the truss volume fraction is kept relatively constant.	54
Figure 3- 23: Comparison of the progress of an embedded element compression model in FFlagSHyP where run time was capped at 36 hours. A small amount of mass scaling was used in the FFlagSHyP model.	56
Figure 3- 24: Wallclock time needed to run an 800 element model with embedded elements using different numbers of parallel cpus for before speed edits to FFlagSHyP (blue circles), after edits (green stars), and removing the saved stress data to reduce overhead (red x's).....	58
Figure 4- 1: Boundary conditions for the initial test plate problem, based plans for future ballistic impact tests with clamped edges.	62

Figure 4- 2: Johnson Cook material model used in a ballistic model with and without plasticity.....	63
Figure 4- 3: Points used for extraction of stress data in the mesh convergency study.	64
Figure 4- 4: Three of the meshes used in the mesh convergence study. a) Mesh8 (3,700 elements) b) Mesh16 (?) c) Mesh26 (180,000 elements).....	65
Figure 4- 5: Stresses at Points 1, 2, and 3 (locations shown in Figure #) for the five different mesh sizes. Mesh8 is the coarsest and Mesh 26 is the finest mesh with the most elements.....	66
Figure 4- 6: Three other metrics of convergence that were used in the mesh refinement study. Changes in hole radius, penetration depth, and change in projectile mass become minimal in meshes larger than Mesh16.	67
Figure 4- 7: Mild Steel plate Abaqus model, boundary conditions, mesh, and Johnson Cook material definition.....	69
Figure 4- 8: Stress, plastic equivalents stress, and strain rate results from the completed mild steel ballistic model.	70
Figure 4- 9: Three different meshes of the steel plate. The top and bottom rows represent two snapshots in time.	71
Figure 4- 10: Stress strain results of experimental tensile tests from Iqbal[70] and tensile tests that were simulated in Abaqus at different strain rates.....	72
Figure 4- 11: Ballistic impact model of the steel plate and projectile. The amount of element erosion is compared between three different mesh sizes.....	73
Figure 4- 12: Dyneema® embedded element mesh from three different views.....	74
Figure 4- 13: Embedded element elastic model of Dyneema® shows similar deformation patterns as the ballistic tests	75
Figure 4- 14: The back face deformation shape that we refer to as tenting.....	76
Figure 4- 15: Back face deformation of plate impact test. a) the initial iteration of a 1/2" plate with an orthotropic matrix material. b) 1/4" " plate with an orthotropic matrix material) 1/4" plate with an isotropic matrix material. This model gives the most realistic looking symmetrical tenting effect	77
Figure 4- 16: Set up used in Karthikeyan's impact experiment[71] and the finite element mesh used to create the embedded element model.....	78

Figure 4- 17: Results of the embedded element models of Karthikeyan's impact experiment[71]. Both the 100 and the 500 fibers per truss cases produce good results.....	79
Figure 4- 18: Experimental test fixture (left) and a close up of the clamping rings which fit into the fixture (right).....	80
Figure 4- 19: X Y view (left) and X Z view (right) of the 8 inch diameter by 1 inch thick HB26 plate when impacted by a 3/8 inch diameter steel sphere traveling at 1028 m/s at impact.	82
Figure 4- 20: a) A single layer of matrix material b) quarter symmetry finite element model with 20 matrix layers connected by cohesive contact constraints and the clamp boundary conditions (red) and symmetry boundary conditions imposed on the model.	84
Figure 4- 21: Host element matrix mesh of 177,600 hexahedral elements.....	85
Figure 4- 22: Progression of creating embedded truss elements in a cylinder-shaped part. .	87
Figure 4- 23: Truss element meshes for the 5581 FpT and 1380 FpT meshes.....	87
Figure 4- 24: Running the embedded element model with truss elements and with beam elements produced visually similar results, with the beam elements showing more early failure and less matrix delamination than the trusses. The beam element model takes.....	90
Figure 4- 25: When compared to CT images of the impact experiment, the embedded element model shows a similar shear plug failure zone and the fiber snap back although the projectile penetration distance is shorter.	92
Figure 4- 26: a&b) Examples of the deformation cone created by the stretching of the principal fibers from other impact experiments. c) Front and (d) back face stress patterns on the fully mirrored plate reveal the characteristic deformation cone associated with Dyneema® and other 0°/90° composites.....	93
Figure 4- 27: Sketch of the indirect tension mechanism from Liu et al[13]. with added description of the mechanisms involved.....	94
Figure 4- 28: Tensile/compressive stresses in the truss elements and shear stress between the matrix layers shows characteristics of the indirect tension mechanism which facilitates the tensile failure of the truss elements early in the impact process.	95
Figure 4- 29: Geometric definitions of the extruded cross section of the curved plate.	99

- Figure 4- 30:** a) Fiber mesh embedded in the curved plate model b) hexahedral mesh of the curved plate matrix material c) close up comparision of the the mesh density of the host mesh and the embedded truss mesh..... 100
- Figure 4- 31:** Stress magnitudes of the impact on the curved plate from two different angles. The characteristic defomration from the $0^\circ/90^\circ$ can clearly be seen in the top image. 101
- Figure 4- 32:** Stress magnitudes part way through pennetration veiwed from the underside of the curved plate. From the stress magnitues, it is observed that the curved fibers (alligned in the x direction) take very little of the stress load, while the flat fibers (alligned in the y direction) support most of the load..... 102
- Figure A- 1:** Stress results of a single truss element in Abaqus and FFlagSHyP where FFlagSHyP uses an assumed 1D neo-Hooke material formulation..... 111
- Figure A- 2:** Stress results from Abaqus and FFlagSHyP for a neo-Hookean truss element in tension. Several different material types were tested to show that the final version of FFlagSHyP works for different properties..... 111
- Figure A- 3:** Abaqus stress and reaction force for a truss element in tension. In this case, the reaction force should be equal to the element stress times the current cross-sectional area. Abaqus uses the assumption that the material is incompressible ($J=1$) for this..... 111

LIST OF TABLES

Table 3- 1: Fiber Volume Fraction Dependency Tests.....	37
Table 3- 2: Strain Rate Dependency Tests.....	41
Table 4- 1: Johnson Cook flow stress parameters and damage evolution parameters used in the steel plate impact model	63
Table 4- 2: Preliminary ballistic impact testing data from Los Alamos National Lab taken November 2019.....	68
Table 4- 3: Mass loss of the steel plate and the projectile for each mesh.....	73
Table 4- 4: Material properties used for the embedded element model of Dyneema®[4]....	74
Table 4- 5: Cohesive contact properties for the matrix layers.	84
Table 4- 6: Material properties for isotropic matrix material and fibers.	88
Table 5- 1: Material properties used in our previous embedded element models for Dyneema®	108
Table 5- 2: Material properties for steel reinforced concrete taken from Garg et al.[29]	108
Table 5- 3: Material properties for brain white matter taken from Garmella et al.[80]. Note that the matrix material is modeled as a Mooney Rivlin material and the fibers as Ogden.....	108
Table C- 1: Material properties used for the Abaqus model of the rectangular mild steel plates.....	111

ACKNOWLEDGEMENTS

I would like to acknowledge the funding and support of Los Alamos National Laboratory. This work would not have been possible without their ever-present technical guidance and advice. I would like to thank the members of my committee for taking the time and effort to serve on this committee. My lab mates Tom, Ritika, Zac, Ann, Oishwarya, and Tara have been a constant support and a source of fun to make working bearable. Thank you to my friends Nina, Meg, Kayla, Bryan, Micah, Lydia, Yz, Rose, Kaley, Kaitlyn, Pat, Sarah, and many others for being there for me, whether we were playing board games, rock climbing, working, staying up late at Wegmans, or listening to me complain about being tired. Friends like you make grad school enjoyable. Thank you to my family for believing in me, supporting me, and helping me move several times. My cat, Tux, who has delivered cuteness and companionship for my years of grad school – he is truly the best and cutest cat. Thank you to Charli XCX for writing banger music that I wrote this entire dissertation to. Finally, to Devon, who showed up in my last year and loved me through my difficulties and has never stopped trying to convince me that all of this has not been the work of an imposter.

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. 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.

Chapter 1

Literature Review and Background Information

1.1 Fiber Reinforced Composites

1.1.1 Description

Ultra-high molecular weight polyethylene (UHMWPE) fiber is thermoplastic fiber in the polyolefin family that is made up of extremely long chains of polyethylene. It was first produced using a gel-spinning process by DSM® (Netherlands) in the late 1970s [1], [2]. Fibers are made of small fibrils of material twisted together to make fibers between 10-20 μm in diameter[2], [3]. The current producers of UMMWPE fibers are DSM® based in the Netherlands under the brand name Dyneema, and Honeywell based in the USA with the brand name Spectra®[1], [3]. The UHMWPE fiber is highly anisotropic with extremely high toughness and high impact strength due to the very strong carbon bond in the longitudinal direction and weak van der Waals interactions that hold the fiber together in the radial direction[1], [2]. This high tensile strength (3.4 GPa) combined with the low density (980 kg/m^3) of the fibers makes them an excellent material choice for making ropes, sails, tear resistant fabrics, and ballistic impact protection[3], [4].

Sheets of UHMWPE can be made by weaving fibers together to create a woven sheet or by creating layers of unidirectional (UD) fibers held together by a matrix resin. These layers are usually about 60 μm thick[3]. The woven sheet can be left as a flexible fabric or made into a solid composite sheet by the addition of resin matrix material. The UD fiber layers can also be stacked together in alternating patterns to form different types of composite layups. Most popular are the $0^\circ/90^\circ$ layups where layers are rotated 90° relative to the previous layer to create a quasi-orthotropic

composite material. **Figure 1- 1** shows a schematic of the $0^\circ/90^\circ$ layup. Sheets of the fiber and thermoplastic matrix material are bound together by a high-pressure low temperature consolidation process. The heating of the composite facilitates the distribution of the resin across the fibers and between plies as well as removes air pockets that would otherwise become voids in the material[4]–[6]. The matrix material, usually polyurethane, serves to hold the fibers together and give the material rigidity and some shear strength[4].

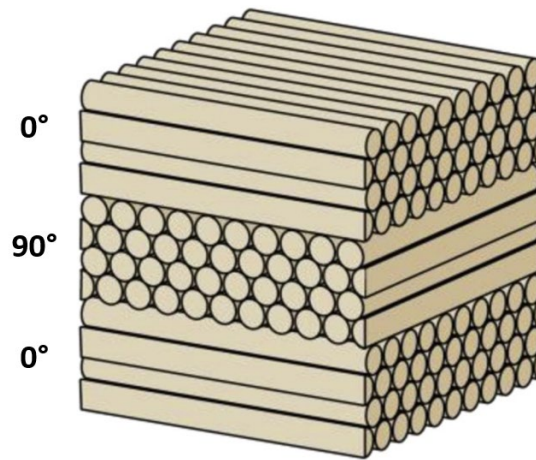


Figure 1- 1: Schematic of $0^\circ/90^\circ$ oriented layers of fibers in a composite, from Liu et al. [7].

This dissertation will focus on Dyneema® and the cross-ply $[0^\circ/90^\circ]$ laminate grade HB26. This material uses a matrix of polyurethane (PUR) which is reinforced by a gel-spun fiber known as grade SK76. This composite contains about 85% fibers by volume. Specifically, we look at the potential of modeling Dyneema® using the embedded finite element method, discussed in further detail in Section 1.2.

1.1.2 Uses in Armor

Armor systems are designed to protect against a variety of threats including shock waves, fragments, and direct fire while also optimizing for low weight, cost, and volume of the material.

In addition to these requirements, armor needs to be designed as a specific category of armor: personnel or body armor, which includes helmets and vest inserts, vehicle armor, or aircraft armor. These different classes change what “optimal” means and there is no best armor for all categories[6]. Effectiveness of armor is often measured by a metric called the ballistic limit or V50 of the armor. V50 is the projectile velocity at which the projectile will penetrate the armor for 50% of shots at that velocity. Because of the cost of running ballistic tests, which are always destructive, characterization experiments are usually limited to 20 or fewer shots[8]. However, for modern armors V50 may no longer be sufficient to characterize the armor[8]. Another important characteristic of armor is the back face deformation (BFD)[1]. This is the amount deformation from the original back face of the armor is created after the armor has successfully captured a projectile. Significant back face deformation can still cause damage (potentially lethal damage in the case of personnel armor) to whatever the armor is protecting[1], [9], [10]. The allowable BFD is different for different armor applications but requires much additional ballistic testing to quantify.

Before the introduction of UHMWPE, steel and ceramics were the most common armor materials. Steel has high strength and high V50 but is quite heavy. Ceramics stop projectiles and absorb the energy by fracturing, making them poor defense against multiple sequential impacts. Both materials are still widely used in armor, but the strength to weight ratio of UHMWPE along with its ability to take multiple impacts makes it ideal, particularly for body armor where weight is important for the mobility of the wearer.

Although woven UHMWPE fabrics have excellent ballistic protection performance, their ballistic protection greatly depends on the weave density, which limits its applications. The nonwoven unidirectional (UD) fabrics consisting of multi-ply with $0^\circ/90^\circ$ or $0^\circ/90^\circ \pm 45^\circ$ orientations can overcome the deficiency and have become preferred for armor designs[1], [11]. The typical deformation and failure mechanisms of $[0^\circ/90^\circ]$ plates of UHMWPE can be divided

into three phases. The initial stage of impact is usually referred to as a shear plugging stage, where the strain rate is very high and fibers directly under the projectile fail with little delamination damage to allow the projectile to create a tunnel into the material[6]. However, as shown by Attwood et al., O'Msta et al., and Liu et al., the actual failure mechanism is an indirect tension mechanism which causes the tensile failure of the fibers, not a shear failure[3], [5], [12], [13]. In the second stage of impact, the projectile has slowed enough that principal fibers stretch instead of break, causing significant strain and delamination between layers of the composite. In the final stage, a combination of bending and membrane stretching resulting in tensile ply fracture on the back surface can cause the plate to fail entirely allowing the projectile to completely penetrate the plate, or all the projectile energy is absorbed by continued fiber stretching, fiber breakage, and delamination, and the projectile motion is halted[3], [6].

To optimize armor for so many different threat protections and use case scenarios, it is important to have a prediction tool to be used in the design process to minimize the amount of ballistic testing needed to produce a new armor package. Because of the complexity of the failure mechanisms involved, ballistic testing will always be necessary, but these tests are costly and time consuming. Computer modeling of UHMWPE is critical both for understanding failure mechanisms, effects of various design parameters, and creating high-quality cost-effective armor packages[6], [8], [11].

1.1.3 Curved Plates

While traditional armor plates are flat inserts that go into a holding vest, curved plate armor conforms to the shape of the wearer for increased coverage, comfort, and mobility[14], [15]. Curvature in composite armor is already found in helmets as well as aircraft armor[1], [16], [17]. There is evidence that the ballistic limit and maximum back face deformation of curved composite

plates are dependent on the radius of curvature[18]–[21]. This makes modeling ballistic impacts of curved UHMWPE plates important for future armor package development. Helmets are also an example of curved armor. The orientation and architecture of UHMWPE composite helmets has been shown to affect the material response after the peak BFD and delamination damage[1], [9], [10].

Fiber reinforced helmets are also widely used but difficult to model because of fiber orientation from the manufacturing process[1], [16]. A common method is to layer darted pinwheel patterns of the composite material and then press the layers into the helmet shape, as shown in **Figure 1- 2** from Li et al. [1]. Curved composite plates can be more simply made by creating the $[0^\circ/90^\circ]$ layup the same way a flat plate is started, but then bend into a curved configuration during the time the matrix resin sets[17], [18]. The embedded element method could be adapted to work for either of these types of curved plates by layering the embedded trusses in the same orientation as the fibers in the material. The orientation of the fibers in these plates may make the usual orthotropic material assumptions less accurate, and the embedded element method would be an easy way to model the correct material orientations. Additionally, there is evidence that the ballistic limit and maximum back face deformation are dependent on the radius of curvature[18]–[21]. Curvature actually increases the ballistic limit up to a certain, optimal, radius of curvature for a particular plate thickness[18], [21] so having an accurate model of curved composite plates would be useful for finding the optimal thickness for the desired curvature.

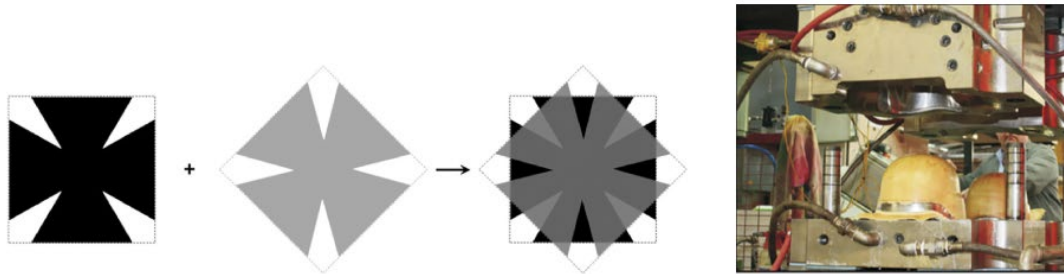


Figure 1- 2: Images of the darted pinwheel pattern and the compression press used to mold fiber reinforced composite helmets from Li et al. [1].

1.2 History of the Embedded Element Method

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[22], [23]. Fish and Markolefas adapted this into a superposition method that they called the s-version of the finite element method (s for “superposition”) to be used for capturing high stress gradients on the free edges of composite laminates[24]. Instead of modifying the element equations, a patch of higher order elements was overlaid on the finite element mesh in the locations where high strain gradients would occur. A modified numerical quadrature scheme is used to calculate the resulting stiffness of the combined elements[25].

Cox et al. [26] developed the Binary Method for modeling woven fiber reinforced composites. Instead of having a homogenized material with refined strain regions, two components of the composite were modeled by different element types. 1D line elements represented the axial stiffness of the fibers while the transverse properties of the composite were represented in an ‘effective medium’ meshed with 3D continuum elements. The two meshes were coupled together by their displacement degrees of freedom[26], [27]. Utomo used a similar method of truss elements in an effective medium to model a unidirectional reinforced composite in a model for ballistic impact[4]. Embedding truss or spring elements into a medium of continuum elements has also been a popular method for modeling reinforced concrete[28]–[31].

The M-cube method, proposed by Zako et al. applied the superposition technique to woven composites. The domain of interest was discretized into micro, meso, and macro scale models and the total stiffness matrix was created from the superposition of the three models[32]. As an attempt to further simplify previous methods by eliminating iterative procedures, Jiang et al. created, the domain superposition technique for woven composites[33]. Fiber tows and matrix material were meshed separately with continuum elements and superimposed with displacement coupling equations. It was later modified to be used for braided composites[34]. Iarve et al.'s Independent Mesh Method is similar to the domain superposition technique, but rather than directly couple the two full meshes, parts of the matrix mesh covered by the tow mesh were discarded. Connections between the tow and remaining matrix elements were then enforced by a penalty function[35]. While one of the benefits of embedded element method is that the host mesh does not need to conform around the embedded sections, some authors suggest meshing techniques that do this in order to ensure a smoother transition in the strain mapping throughout the model[36], [37].

Embedded element techniques have been used for modeling a range of materials. Belytschko and Fish's s-method was developed for modeling shear bands in metals. Others have used the method for reinforced concrete[28]–[31], fiber reinforced composites[26], [27], [35], and, more recently, bio-material such as white matter in the brain[38]–[40]. Fiber reinforced composites made from ultra-high molecular weight polyethylene have been increasingly used in the design of ballistic armor due to their low density and high strength[11], and there is a need for development of a reliable model to both assist with armor design and understand penetration mechanisms. Most embedded element models of composites have been for woven composites, often these are microscale models with the goal of understanding how the weave pattern carries stress[33]–[36],

[41], [42]. It has been shown that unidirectional and cross-ply composites perform better in ballistic armor applications than woven composites[43]–[45].

Most models for unidirectional and cross-ply composites have been based on a homogenization approach, where either layers of the composite or the entire composite is modeled as a homogenous continuum[4], [6], [7], [44], [46]–[55]. These models have worked well for modeling quasi-static or elastic events. However, in order to model something like a ballistic impact, models must include damage mechanics. The homogenization model makes defining the different types of damage (fiber failure, delamination, matrix cracking) and interpreting the damage modes predicted by the model difficult. Using the embedded element technique, the definition of these failure mechanisms could be easier since each material is defined independently, rather than averaged into a continuum model[56]. Additionally, 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[25], [41].

However, in the implementation of the embedded element method, the volume of the host elements are not modified to “make room” for the embedded elements. The volumes of the two meshes occupy the same space. In cases where this is left uncorrected, it causes internal energy and kinetic energy to be miscalculated by double counting volume of the embedded mesh[40], [41]. Fiber reinforced composites typically have a high fiber volume ratio, so the redundant volume adds unacceptable amounts of additional stiffness and mass. There have been several different methods introduced to account for the redundant volume[23], [32]–[35], [41]. 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[33], [34], [36], [39], [41]. Some authors have used modified

integration schemes, averaging or ignoring overlapping integration points[23], [35]. Vanalli modified the host elements' energy to include the stiffness of randomly distributed fibers[28]. Radtke used a method similar to XFEM where an enrichment function was added to the host elements to represent fibers. This method could also model fiber slip[57]. However, these methods were all created for elastic problems, and none for time dependent dynamic problems. In a paper on modeling brain matter, Garimella 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 and was limited in application.

1.3 Finite Element Modeling of Fiber Reinforced Composites

Modeling fiber reinforced composites is difficult due to the complexity of the materials involved and the many failure modes and our lack of understanding about exactly how each mode initiates and which are important to include in a particular model[48]. Models are built on one of three scales: micro-scale, where the individual fiber and matrix are explicitly modeled and meshed separately, meso-scale, where the properties of individual plies are modeled and stacked together and macro-scale, where the composite is modeled as a continuum and the properties are homogenized into an orthotropic material[48], [58]. These three size scales are illustrated in **Figure 1- 3**.

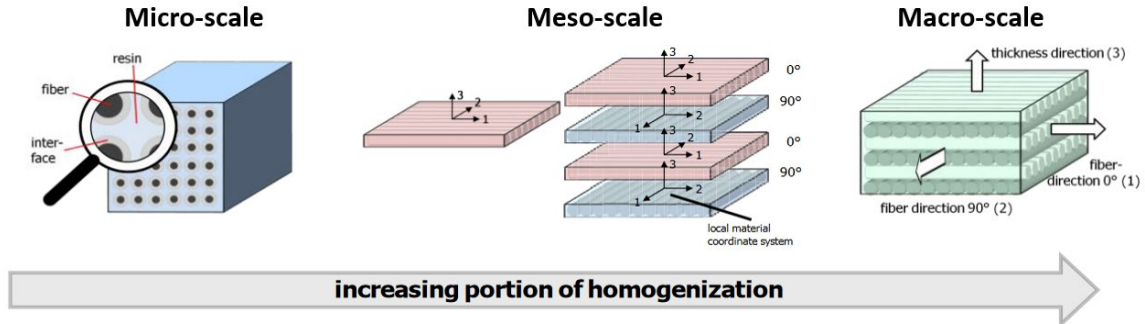


Figure 1- 3: The different size scales of modeling fiber reinforced composites.

Micro-scale modeling is useful for investigating the physics of the interactions between the fiber and matrix materials but is computationally expensive for modeling a large system. In macro-scale methods, the behavior of the entire $[0^\circ/90^\circ]$ composite is approximated as an orthotropic material[51]–[55], [59], [60]. Using this method, it's easy to apply the material definition to a large or complicated mesh and the computation time is reduced by assuming the composite is a single continuous material rather than two types of combined material. These aspects make the total homogenization method good for large scale models, like for ballistic plate impact, but as you go to a smaller length scale the assumption that the material behaves as a continuum will break down and the fiber and matrix materials need to be meshed separately for more accuracy. In this total homogenization method, damage and failure are usually defined using classic composite failure measures such as Hashin damage or a custom combination of maximum directional stress[11], [53], [54]. This type of modeling makes delamination, an important damage mechanism, difficult to identify[4], [37], [46]. Meso-scale modeling addresses this by modeling distinct layers of material. Layers are chosen either to be only a 0° material layer that is rotated to get the 90° , or homogenized as a full $[0^\circ/90^\circ]$ ply similar to the total homogenization method[13], [44], [46], [47], [49], [50]. These layers are stacked and held together either by cohesive elements or a cohesive contact layer that will be removed once the delamination damage criteria is met, allowing the layers to peel apart as they do in the real composite[11].

Multi-scale modeling of composites attempts to combine aspects of all three size scales. A homogenized macro-scale mesh is used as the main mesh in a large-scale analysis[61]. The homogenized material properties are used to calculate a first estimate of the stress in specific locations. The material loads are then mapped to a micro-scale unit cell model where the interaction between the fibers and the matrix are considered, and their specific stresses, internal forces, and failure initiation criteria are calculated. These final values are then mapped back to the main homogenized mesh to continue the analysis. This can be a good method for combining the strengths of different length scales of modeling, but also can be difficult to program and computationally expensive.

1.4 Thesis Statement

We hypothesize that the embedded element method of FEA could be used to create meso-scale and macro-scale models of fiber reinforced composites, particularly for ballistic applications. 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 would also have the ability to model curved plates. This element mesh is easy to create yet retains distinction between the fiber and matrix components for meso-scale analysis. Since fiber reinforced composites have a large fiber volume fraction, 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.

Chapter 2

MATLAB Script for Embedded Fiber Creation

Using the embedded element method to model fiber bundles requires defining a truss (or beam) element mesh that precisely fits inside of the matrix material elements. This process would be tedious to do by hand if there are many truss elements to create, but the process is well suited for automation. Initial work on such an automation process has focused on the ability to take an existing input file from the existing finite element software Abaqus (usually generated from Abaqus's well established user interface called Abaqus CAE) and add layers of truss elements to represent fiber bundles in a $0/90^\circ$ orientation. This results in a new input file that can be immediately run by Abaqus's solver to obtain results. Abaqus has an established input file format which lends itself towards automation, but the principles used to read the file and determine the locations of the truss elements can later be used to automate other types of finite element solver input files.

2.1 Flat Plate

Initial work simplified the problem further, assuming that the matrix material part would have a constant cross-sectional area and have a specific orientation with respect to Abaqus's underlying Cartesian coordinate system. The Abaqus input file must also specify a node set that contains all the nodes on the external boundary of the part. With these limitations, a MATLAB script was written that calculated the endpoints of two node truss elements based on the shape of the part's cross section. These truss elements created a single layer along the bottom surface of the part. The layers are then rotated 90° and copied along the entire part to create the $0/90^\circ$ layers. An example of this is shown in **Figure 2- 1** based on a simple cylinder-shaped part that was created in

Abaqus. This method worked well for simple shapes but would not work if the part was not symmetric. While this was a good start, the script needed to be more robust.

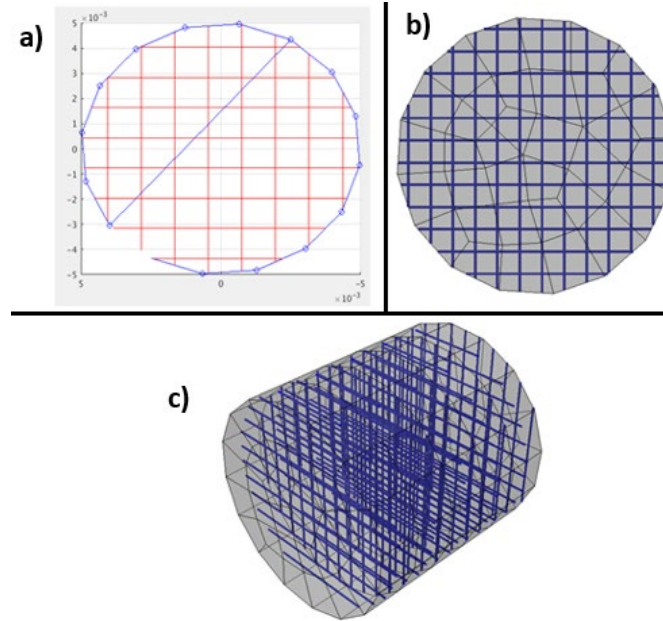


Figure 2- 1: a) Fibers defined in MATLAB based on the cylinder cross section. b) Cross section of the cylinder model imported into Abaqus from the MATLAB output. c) Whole cylinder part as imported into Abaqus, with truss elements in a 0/90° orientation.

A new MATLAB script was written to handle more complex shapes. Instead of looking at only one cross section, this code works by reading in the nodes and element connectivity of the external boundary of a part, and recreating the surface in MATLAB (Figure 2- 2a). A grid of points in and around the part are defined by the script. A function then determines which points in this set are located inside of the surface (Figure 2- 2b). From these points, the script determines which ones are closest to the edges of the part, these are designated as truss endpoints (Figure 2- 2c). Finally, the trusses are defined by drawing lines between the endpoints, and these lines are subdivided into truss elements depending on the user defined mesh density (Figure 2- 2c). This method works for complex shapes, such as the tensile test specimen and the quarter circular plate in Figure 2- 3. The contact surfaces and embedded element constraint needed to run a simulation in Abaqus are

automatically defined so the new input file can be submitted to the Abaqus solver without any additional work by the user.

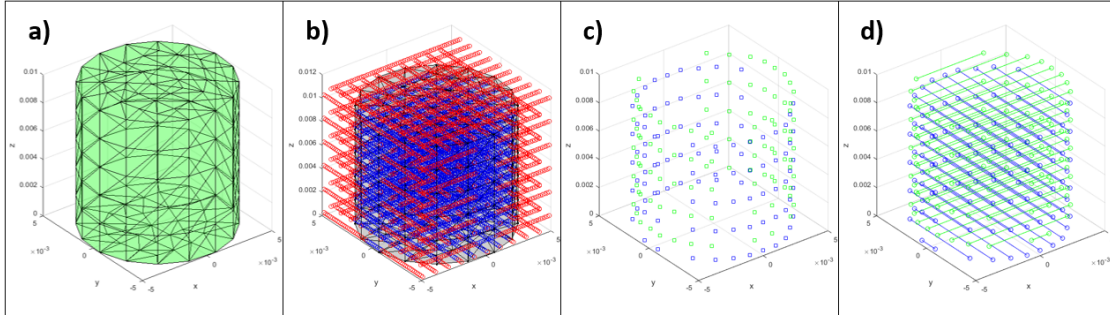


Figure 2- 2: Progression of creating embedded truss elements in a cylinder-shaped part.

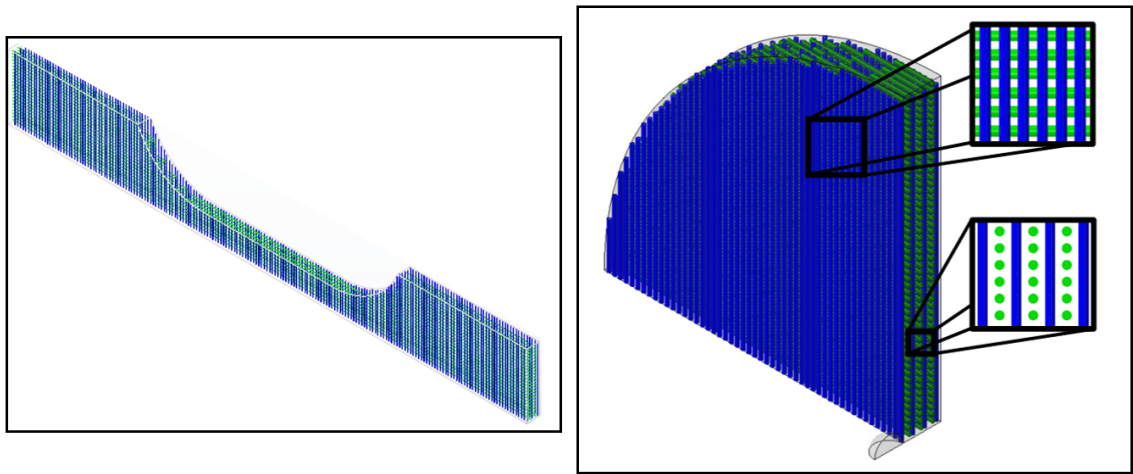


Figure 2- 3: Examples of truss element meshes created by the MATLAB script.

The size of the truss elements are determined by assigning a number of fibers to be represented by a single truss element, similar to the process in Utomo et al [4]. The cross-sectional area is then calculated by Equation 2-1, where n is the number of fibers assigned to a truss and D_{fiber} is the fiber diameter, assuming a fiber diameter of 17 microns (which is common for Dyneema fibers). A larger number of fibers per truss (FpT) yields less, but larger, truss elements for a courser mesh, while a smaller number creates a finer mesh. The MATLAB script then stacks truss elements in $0^\circ/90^\circ$ layers based on the outline of the shape. Trusses are not

placed more than $\frac{1}{2}$ a truss element diameter away from the edge of the part to avoid trusses that fall outside of the part.

$$A_{truss} = n \frac{\pi}{4} D_{fiber}^2 \quad \text{Eq. 2-1}$$

2.2 Curved Plates

Part of the draw of using the embedded element method to model composites is its ability to model more complex fiber orientations. A particular application of this is modeling curved plates that could be used for human conformal armor or vehicle armor. To model the composite correctly, the truss elements representing the fibers need to conform to the curved shape of the plate. Previously, a MATLAB script was developed to add $[0^\circ/90^\circ]$ fibers to any shape. A simple way to reuse the current script for the curved plate geometry is to convert the curved plate geometry to a flat shape, use the old script to add $[0^\circ/90^\circ]$ truss elements, and then convert this flat shape, along with the truss elements, back to the curved geometry. **Error! Reference source not found.** shows a graphic of this process.

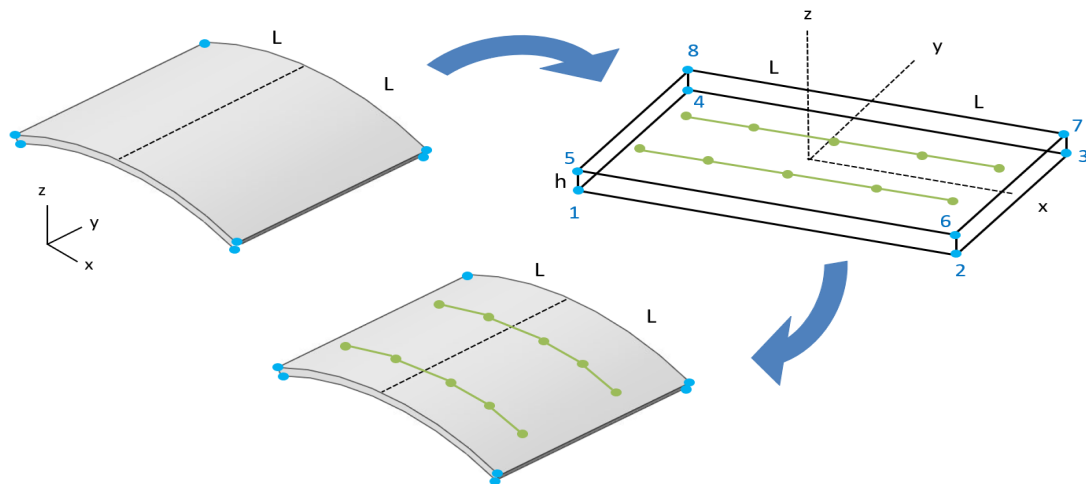


Figure 2- 4: Illustration of the Curved – Flat – Curved mapping process to include truss element in a curved plate.

The curved plate is mapped to a flat shape by creating a parallelogram using the arc length of the plate as the length and the plate thickness as the height. Then the distances to each of the 8 corners of the curved plate are used to map the rest of the points on the curved plate to the flat plate. Truss elements are added to the flattened shape based on the truss diameter and truss mesh density specified by the user. The node locations of these truss elements are mapped back to the curved shape. The reverse mapping for the x and y directions (defined in Figure 2-4) is done by measuring the distance to the 8 corner nodes of the flat shape and transferring that measurement to the corners of the curved shape. The curvature in the z direction is added by interpolating between the two radii of the top and bottom of the curved plate. These radii, along with the concentric center of both arcs can be calculated based on the curvature of the plate. Then, the z coordinate of any node in the flattened space can be used to interpolate between the two radii and find the correct z coordinate for the node in the curved space. This process is illustrated in Figure 2- 5.

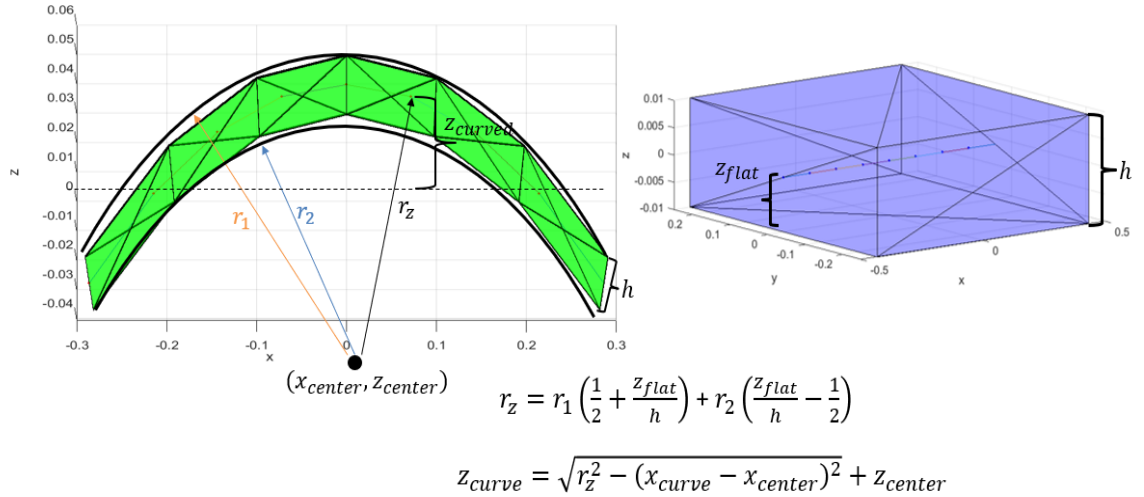


Figure 2- 5: An illustration of the process to interpolate between the two radii curved plate arcs and the calculation of z coordinates in the curved space from the flattened space.

The completed MATLAB script reads an Abaqus input file of the desired curve geometry including boundary and loading conditions that will be used in the simulation, adds truss elements

in a $[0/90^\circ]$ configuration, and writes a new Abaqus input file that can be run with no further edits by the user. Figure 2- 6 shows an example of a curved plate made in Abaqus and the truss elements added by MATLAB as they are rendered in Abaqus's output database reader. The MATLAB script has several limitations on the types of curved shapes it can manage. The curved plate must be made from an extruded curved shape, which must be the area between two concentric circles (such as the shape shown on the left in Figure 2- 5). Additionally, the plate needs to be oriented so that the global z direction defined by Abaqus is perpendicular to the curvature direction and the x and y directions are along the other major axes of the plate. The method used in this script could be expanded to more types of curved plates but would require adjustments. However, the shape being used in this model will be sufficient for demonstrating the embedded element method in a curved plate.

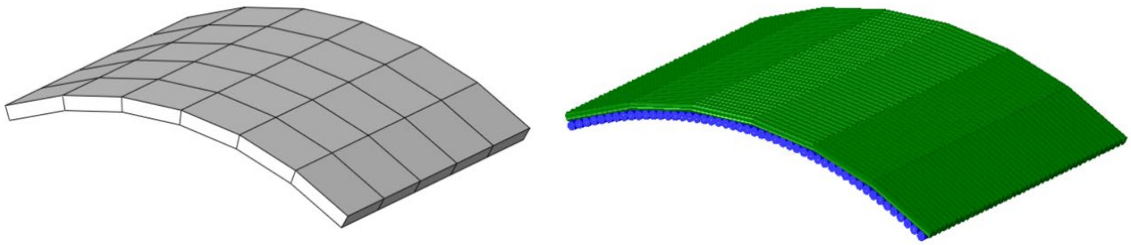


Figure 2- 6: A curve plate created and meshed in Abaqus along with a rendering of the embedded truss elements added by the MATLAB script.

2.3 Limitations

The code still has a few drawbacks. The original part still must be oriented such that one surface is on the plane $z = 0$ and the user must define a node set of the outside surface while creating the Abaqus input file. Using this point method for determining the truss endpoints (illustrated in **Figure 2- 2a**) means that the truss endpoints will never quite reach the part boundary. The number of points in the point cloud can be increased by the user, but using a larger number of points slows the code down significantly. With a modest number of points, the distance between

the edges of the part and the start of the fibers is small and likely has negligible effects on any simulation results. Additionally, the code can only handle parts that are meshed with hexahedral elements. Adding the ability to recognize multiple element types should be straightforward but has not been done yet. For the embedded mesh to be used in the custom embedded element implementation discussed later, the truss elements can span no more than two host elements, and the user must make sure they set the truss element discretization length accordingly. Some finite element codes will remesh embedded elements so that embedded elements do not cross host element boundaries, but creating an efficient software further divide the truss mesh was outside the scope of this work. In addition to these limitations, for the curved plate version the plate can only have one direction of curvature (not a spherical section) and must be made of two concentric arcs for the mapping process to work. The curved plate also must have a constant thickness and the edges of the plate need to be perpendicular to the two curves.

Chapter 3

Embedded Element Finite Element Code

For this project, we want to modify the implementation of the embedded element method to eliminate the current volume redundancy. Since we are modifying the implementation itself, we need access to the base code of an explicit dynamic finite element program (since this project's target application is high-rate impact events). A MATLAB-based finite element code called FFlagSHyP (Finite element **L**arge **S**train **H**yperelasto-plastic **P**rogram) from the textbook Nonlinear Solid Mechanics for Finite Element Analysis[62] was chosen to test volume redundancy correction in the embedded element method. A MATLAB based code, while slower than other programming languages, is easy to debug and this code is available free online. At the time of starting the project, FFlagSHyP only had a Newton-Raphson solution method for static (zero acceleration) problems, so a time integration algorithm needed to be added. We chose to do this in the form of a central difference time integration. This extra step was helpful for better understanding how the solution method works. For example, it was discovered that artificial bulk viscosity damping needed to be added to FFlagSHyP to help reduce “ringing” in the system. The documentation for the finite element program Abaqus contains a description for how damping is added as an artificial pressure term[63]. Figure 3- 1 shows the y_y stress of a cylinder model where a tension force is applied in the z direction. Without the artificial damping, the Poisson's effects cause stress waves to reverberate through the model.

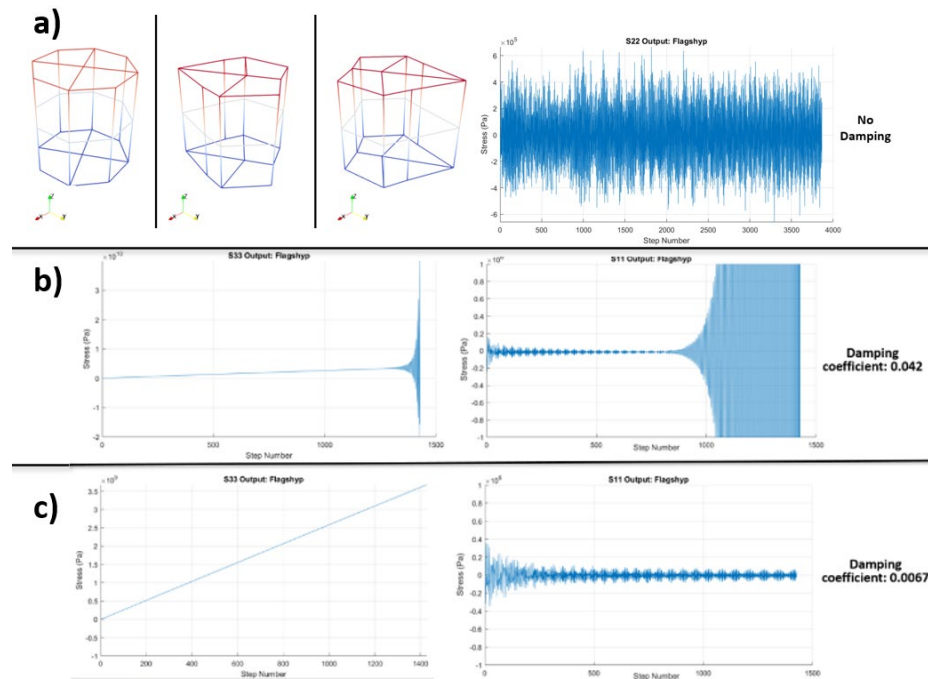


Figure 3- 1: Effects of damping showed on a four element cylinder in tension. a) With no damping, the model is very unstable. b) Damping added to the cylinder model, but the damping coefficient is too high, and the system is still unstable. c) With the damping applied, the wild oscillations in the stress are eliminated.

Throughout the process of programming the dynamic explicit solver and adding the basic embedded element method, the commercial code Abaqus has been used as a standard to verify FlagSHyP's results. Some verification could be done using analytical calculations, but the material models being used are nonlinear and make analytical solutions complicated.

3.1 Addition of Neo-Hookean Truss Elements

The ability to run a model with multiple element types needed to be added to FFlagSHyP since the method of embedded elements we want to do consists of truss elements embedded in a host. We also want to be able to test nonlinear materials, so a method for calculating the one-dimensional truss stress using the generally 3D Neo-Hooke material model was developed. The derivation of this model is included in Appendix A. The final model can recreate the results from

Abaqus as shown in Figure 3- 2 with the exception of incompressible materials, which can be avoided for now.

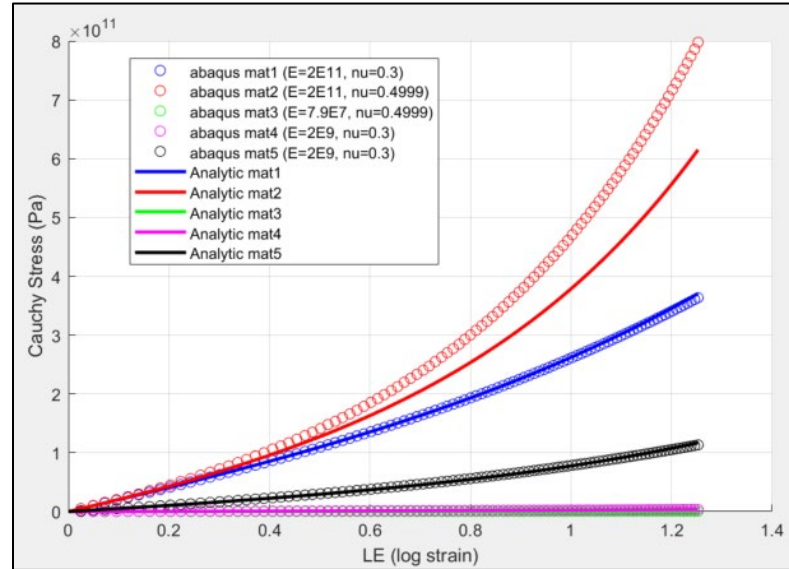


Figure 3- 2: Stress results from Abaqus and FFlagSHyP for a neo-Hookean truss element in tension using some contrived material types. Several different materials were tested to show that the final version of FFlagSHyP works for different properties.

3.2 Addition of Embedded Element Method

Once the explicit dynamic finite element code was working, the ability to use embedded elements was added to FFlagSHyP. For the embedded element method to work, the embedded elements need to move with the displacement of the host elements, and the host element motion needs to be influenced by the presence of the embedded elements. In normal finite element practices, both interactions between elements are achieved by elements sharing nodes with neighboring elements. Forces applied to nodes by the connecting elements determine the motion of the nodes, and the motion of the nodes determines the elements' deformations and therefore the amount of force they apply to the nodes. In the embedded element method, host and embedded elements do not share any nodes, the embedded element nodes are located somewhere inside of the host elements. There needs to be a different mechanism for them to communicate with each other.

In a general framework, the governing equations of the embedded element method are found by finding the stationary point of the energies of both meshes. Let D_F be the fiber domain located in the host body M as illustrated in **Figure 3- 3**. The displacement of a point \underline{X} the fiber is $\underline{u}_F(\underline{X}, t)$ and the displacement of the background host body is $\underline{u}_M(\underline{X}, t)$. The total energy of both meshes includes the sum of the internal and kinetic energies of both meshes minus the external work done on each mesh.

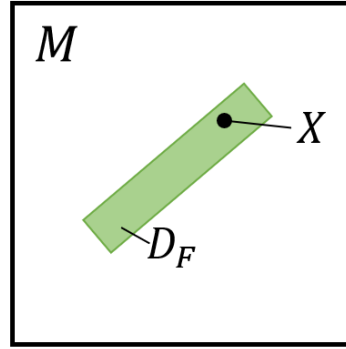


Figure 3- 3: Illustration of a fiber structure embedded in matrix material.

$$W_M(\underline{u}_M) = W_{int}(\underline{u}_M) + W_{kin}(\underline{u}_M) - W_{ext}(\underline{u}_M) \quad \text{Eq. 3-1}$$

$$W_F(\underline{u}_F) = W_{int}(\underline{u}_F) + W_{kin}(\underline{u}_F) - W_{ext}(\underline{u}_F) \quad \text{Eq. 3-2}$$

$$W = W_M(\underline{u}_M) + W_F(\underline{u}_F) \quad \text{Eq. 3-3}$$

This is the total energy for two separate meshes. To add the embedded constraint, we need the embedded (fiber) mesh to follow the displacements of the host mesh, that is, the displacement at any point in the embedded mesh needs to be equal to the displacement at the same location in the background matrix mesh.

$$\underline{u}_F(\underline{X}, t) - \underline{u}_M(\underline{X}, t) = 0, \quad \underline{X} \in D_F \quad \text{Eq. 3-4}$$

This constraint can be enforced by using a Lagrange multiplier to create an augmented work equation for the system.

$$W_{aug} = W_M(\underline{u}_M) + W_F(\underline{u}_F) + \int_{D_F} \lambda(\underline{u}_F(\underline{X}, t) - \underline{u}_M(\underline{X}, t)) dD_F \quad \text{Eq. 3-5}$$

The variation in W_{aug} (that is δW_{aug}) can be found by introducing variations relative to u_M, u_F , and λ . Setting δW_{aug} equal to zero and substituting in the finite element discretization leads to the finite element solution. However, to evaluate the integral in Equation 3-5 $\underline{u}_M(\underline{X}, t)$ must be evaluated in D_f , so there needs to be a map between the domain of the matrix, M , and the domain of the fibers, D_f . If we limit the application to isoparametric elements for the matrix, this map can be achieved by using the matrix element shape functions, as will be described in the next section.

3.2.1 Tying the Displacement Degrees of Freedom

If slip between the host and embedded elements is not allowed, then the displacement degrees of freedom of the embedded elements can be tied to the host element motion. To achieve this, there are three steps that need to be programmed. First, the nodes of the embedded elements must be identified from the user input and differentiated from the nodes of the host elements. Second, the location of the embedded nodes inside of the host elements needs to be found. Specifically, we need to know which host element contains each embedded node so finally, the displacements of the embedded nodes can be set equal to the displacement of the host elements at the location of the embedded node.

In the finite element method, when using Lagrangian elements, the displacements of an element are only calculated at the nodal coordinates. Displacements anywhere else in the element are interpolated from these values. Interpolation is generally done via the element shape functions, usually denoted in matrix form \mathbf{N} . Additionally, for isoparametric elements, these shape functions are used to map the element to its “natural” space. In the case of a linear hexahedral element being used here this is a perfect cube centered at the origin of its coordinate system. All calculations on

the element (numerical integration, internal force, deformation) are done in this natural space. The shape functions only take coordinates in the natural space as arguments ($N(\xi)$). For the embedded element method, we need to set the displacement of the embedded nodes equal to the displacement inside of the host element, which means using the shape functions to interpolate between host nodes and get the displacement of the host element at the location of the embedded node. This is represented by the Equation 3-6.

$$\underline{u}_i = N_H^a(\underline{\xi}_i) \underline{u}_H^a \quad \text{Eq. 3-6}$$

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 3- 4 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 in the global finite element model. The natural space of the element, including the nodes of the embedded element are shown on the left. $\underline{\xi}_i$ are the coordinates of the embedded node in the natural space of the host element. These natural coordinates are necessary for interpolating the displacement of the host element nodes to find what the displacement of the embedded nodes should be. The shape functions (along with the degrees of freedom) only work to map the natural space to the model space, the inverse function cannot be found explicitly. But since \underline{x}_i is known and we want $\underline{\xi}_i$, we need the inverse. We can get an approximate value of $\underline{\xi}_i$ using a Newton-Raphson root finding method as outlined below and the right side of Figure 3- 4.

\underline{x}_i and $\underline{\xi}_i$ are related by

$$\underline{x}_i = N_H^a(\underline{\xi}_i) \underline{x}_H^a \quad \text{Eq. 3-7}$$

Where $N(\xi)$, \underline{x}_i , \underline{x}_H^a are known. Subtracting \underline{x}_i from both sides

$$f(\underline{\xi}) = N_H^a(\underline{\xi}_i) \underline{x}_H^a - \underline{x}_i = \underline{0} \quad \text{Eq. 3-8}$$

Newton-Raphson can be used to find an approximate value of $\underline{\xi}_i$.

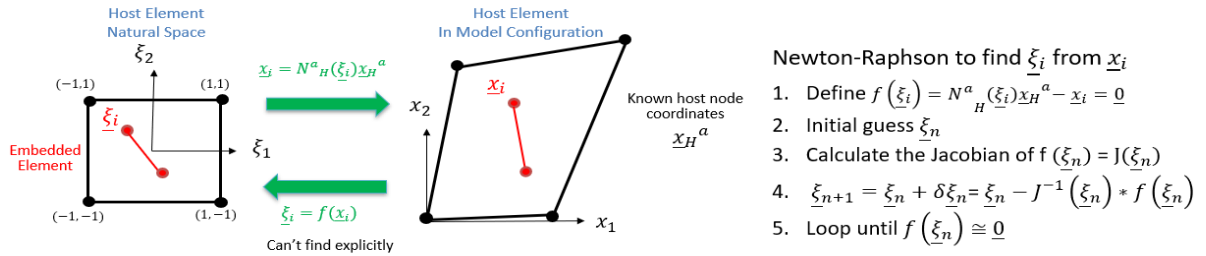


Figure 3- 4: Relation between host element, embedded node, and natural space, and the algorithm for Newton-Rapson method used to approximate an inverse relation.

With this approximate value of $\underline{\xi}_i$ the displacement of the host element at the location of the embedded node can be assigned to the embedded node.

3.2.2 Interaction of Forces

The embedded element method is essentially solving two interdependent finite element problems. It is more efficient than solving two full problems because only the displacements of the host elements need to be calculated. The embedded node displacements are simply interpolated from the host displacements. The embedded elements contribute to the stiffness of the host elements and the rest of the solution method focuses on how the combined forces of the embedded and host element stiffness influence the motion of the host nodes. 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- 5 shows the general algorithm for the implementation of explicit dynamics[64]. In the implementation of this algorithm, FFlagSHyP uses a lumped mass matrix to simplify the inversion of the matrix (\mathbf{M}) for the acceleration calculations.

Box 6.1 Flowchart for explicit time integration

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. *getforce*
3. Compute accelerations $\mathbf{a}^n = \mathbf{M}^{-1}(\mathbf{f}^n - \mathbf{C}^{\text{damp}} \mathbf{v}^{n-1/2})$
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 I on Γ_{v_i} : $v_{iI}^{n+1/2} = \bar{v}_i(\mathbf{x}_I, t^{n+1/2})$
7. Update nodal displacements: $\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t^{n+1/2} \mathbf{v}^{n+1/2}$
8. *getforce*
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.

Subroutine *getforce*

0. Initialization: $\mathbf{f}^n = \mathbf{0}$, $\Delta t_{\text{crit}} = \infty$
1. Compute global external nodal forces $\mathbf{f}_{\text{ext}}^n$
2. Loop over elements e
 - i. GATHER element nodal displacements and velocities
 - ii. $\mathbf{f}_e^{\text{int},n} = \mathbf{0}$
 - iii. Loop over quadrature points ξ_Q
 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. $\mathbf{f}_e^{\text{int},n} \leftarrow \mathbf{f}_e^{\text{int},n} + \mathbf{B}^T \sigma^n \bar{w}_Q J |_{\xi_Q}$
END quadrature point loop
 - iv. Compute external nodal forces on element, $\mathbf{f}_e^{\text{ext},n}$
 - v. $\mathbf{f}_e^n = \mathbf{f}_e^{\text{ext},n} - \mathbf{f}_e^{\text{int},n}$
 - vi. Compute Δt_{crit}^e , if $\Delta t_{\text{crit}}^e < \Delta t_{\text{crit}}$ then $\Delta t_{\text{crit}} = \Delta t_{\text{crit}}^e$
 - vii. SCATTER \mathbf{f}_e^n to global \mathbf{f}^n
5. END loop over elements
6. $\Delta t = \alpha \Delta t_{\text{crit}}$

Figure 3- 5: Explicit time integration flowchart from Nonlinear Finite Elements for Continua and Structures by Belytschko et al. [64].

Garimella et al. [40] presents a method on how to add the interaction of the embedded element forces with their respective host elements. The subroutine shown in Figure 3- 6 is meant to replace the get force subroutine in Figure 3- 5.

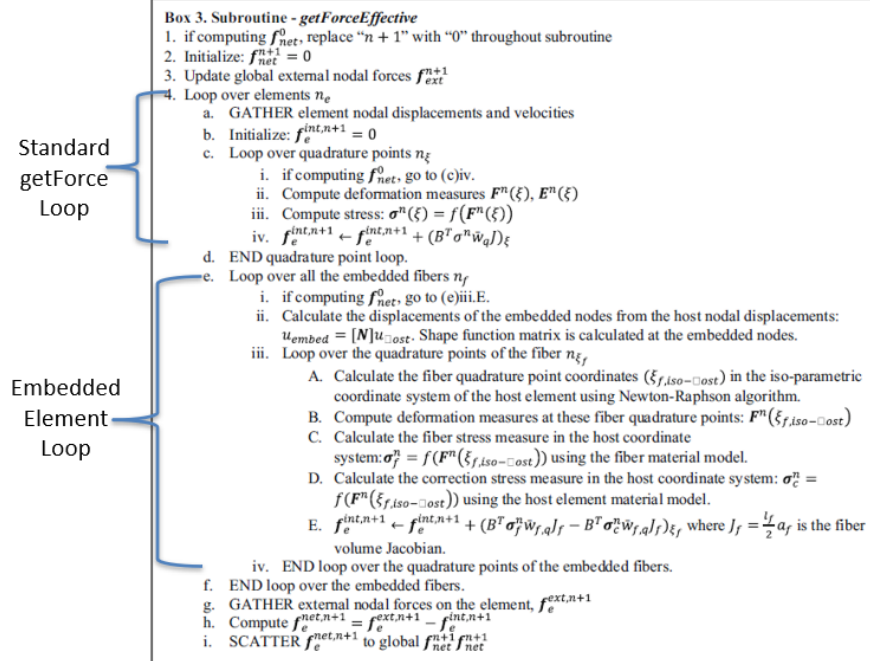


Figure 3- 6: Subroutine for internal force calculation from Garimella et al. [40].

The internal force vector is assembled in point 4.e.iii.E. This equation is shown in Figure 3- 7, with each term labeled with the type of force they represent. The volume redundancy correction will be discussed later. An interesting feature of this equation is that, while the second term is essentially a numeric integral over the embedded element, the shape function derivatives of the host elements are used. Since the embedded elements displacements are tied to the host element in the first step, the displacement of the host element and embedded element is represented by the host element shape functions, provided that the coordinates of the embedded element in the host natural domain are used.

$$f_e^{int,n+1} \leftarrow f_e^{int,n+1} + (B^T \sigma_f^n \bar{w}_{f,q} J_f - B^T \sigma_c^n \bar{w}_{f,q} J_f) \xi_i$$

Host Element Internal Force
Embedded Element Contribution
Volume Redundancy Correction

σ_f^n – embedded element stress
Both B^T are the derivatives of the host element shape functions.

σ_c^n – "correction stress" embedded element stress, but calculated using the host element material

Figure 3- 7: Calculation of the host element internal force vector with the influence of embedded elements and volume redundancy correction.

This method was implemented for the case of a smaller hexahedral element embedded in a larger hexahedral element and worked well. But since this is just adding embedded element forces to the host element internal forces, it seemed unnecessary to use Gauss Quadrature and integration over a one-dimensional truss element. Truss element volume can be integrated exactly so there is no need for Gauss Quadrature. Therefore, it would be easier 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. A new version of getForce_effective that uses this method is included in Appendix B.

3.3 Volume Redundancy Correction

The classic embedded element 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 3- 8 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[40].

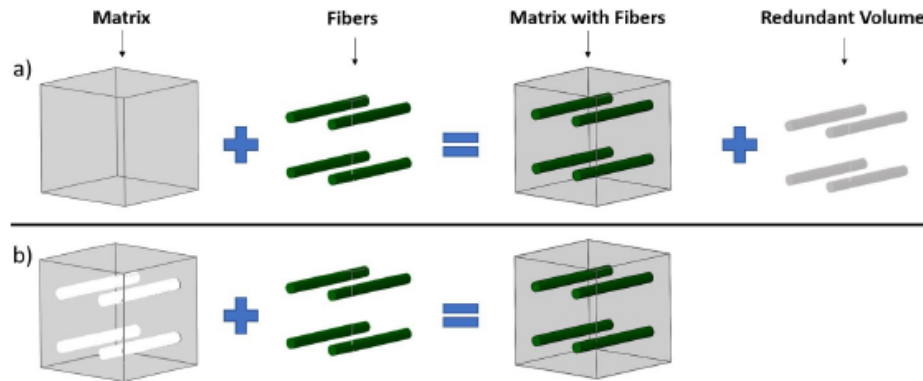


Figure 3- 8: a) How volume redundancy is created in the embedded element method. b) How the system should be correctly modeled.

3.3.1 Numerics of Additional Mass, Internal Force, Kinetic Energy

The energy issues created by double counting volume can be shown by how the model's energy terms are calculated. For embedded element models, the total energy terms are the sums of the energies of the host element and of the embedded element. Here we will begin to refer to the host elements as the matrix elements and the embedded elements as fiber elements. This is the terminology that would be used for modeling a fiber-reinforced composite. The internal energy, W_{int} , (Eq. 3-9) is found by integrating the strain energy over the volumes of the two meshes. The

kinetic energy, W_{kin} (Eq. 3-10) is an integral of momentum times velocity over the volume, and the external work, W_{ext} , (Eq. 3-11) is the summation of the body forces on the two meshes plus any applied traction forces, multiplied by their respective displacements.

$$W_{int} = \overbrace{\int_M u_{i,j} \sigma_{ij_M} dV_M}^{\text{Matrix Strain Energy}} + \overbrace{\int_F u_{i,j} \sigma_{ij_F} dV_F}^{\text{Fiber Strain Energy}} \quad \text{Eq. 3-9}$$

$$W_{kin} = \overbrace{\int_M \rho_M \frac{d^2 u_i}{dt^2} u_i dV_M}^{\text{Matrix Kinetic Energy}} + \overbrace{\int_F \rho_F \frac{d^2 u_i}{dt^2} u_i dV_F}^{\text{Fiber Kinetic Energy}} \quad \text{Eq. 3-10}$$

$$W_{ext} = \overbrace{\int_M \rho_M b_i u_i dV_M}^{\text{Matrix External Load}} + \overbrace{\int_F \rho_F b_i u_i dV_F}^{\text{Fiber External Load}} + \overbrace{\int_{S_2} t_i u_i dA}^{\text{Surface Traction Load}} \quad \text{Eq. 3-11}$$

In these equations, u_i is the displacement field of the system and σ_{ij} is the Cauchy stress 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) applied as a boundary condition on the surface S_2 (assuming b_i and t_i are constant and the initial displacements are zero). 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. However, in finite element programs 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.

$$W_{int} = \underbrace{\int_M u_{i,j} \sigma_{ij_M} dV_M}_{\text{Matrix Strain Energy}} - \underbrace{\int_F u_{i,j} \sigma_{ij_M} dV_F}_{\text{"Extra" Matrix Strain Energy}} + \underbrace{\int_F u_{i,j} \sigma_{ij_F} dV_F}_{\text{Fiber Strain Energy}} \quad \text{Eq. 3-12}$$

$$W_{kin} = \underbrace{\int_M \rho_M \frac{d^2 u_i}{dt^2} u_i dV_M}_{\text{Matrix Kinetic Energy}} - \underbrace{\int_F \rho_M \frac{d^2 u_i}{dt^2} u_i dV_F}_{\text{"Extra" Matrix Kinetic Energy}} + \underbrace{\int_F \rho_F \frac{d^2 u_i}{dt^2} u_i dV_F}_{\text{Fiber Kinetic Energy}} \quad \text{Eq. 3-13}$$

$$W_{ext} = \underbrace{\int_M \rho_M b_i u_i dV_M}_{\text{Matrix External Load}} - \underbrace{\int_F \rho_M b_i u_i dV_F}_{\text{"Extra" Matrix External Load}} + \underbrace{\int_F \rho_F b_i u_i dV_F}_{\text{Fiber External Load}} + \underbrace{\int_{S_2} t_i u_i dA}_{\text{Surface Traction Load}} \quad \text{Eq. 3-14}$$

It can be seen that 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 and no embedded elements.

3.3.2 Proof of Volume Redundancy

Volume redundancy is not corrected in most commercial codes. This is acknowledged[65], leaving it up to the analyst to account for the redundant mass if it would cause any significant effect. Since the embedded element method is mostly used in quasi-static models, the redundancy can be addressed by reducing the density and stiffness of the host or embedded material[4], [7], [31], [33], [34]. For applications in ballistic fiber reinforced composites, this is unacceptable. Since the fiber volume ratio is large, it will alter the wave speed of the matrix 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 unit cube with embedded truss elements ought to behave as if it was a single homogenous cube if the cube and the embedded truss elements have the same material properties (ie, 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. 25 truss elements result in a fiber volume ratio of 0.5. Examples of the models are shown in Figure 3- 9a.

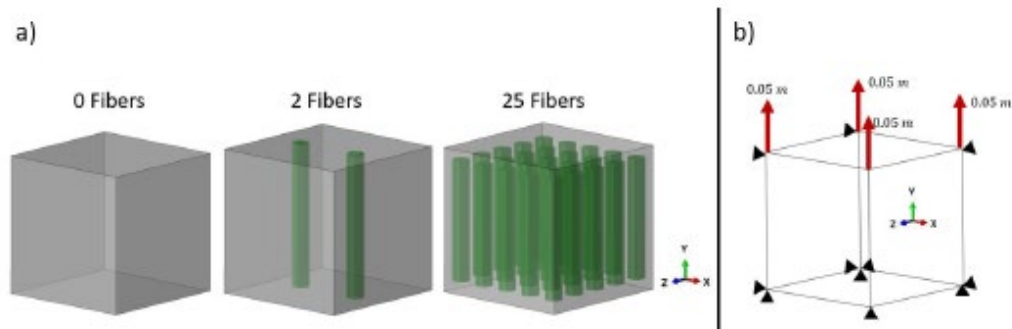


Figure 3- 9: a) Unit cube models containing 0, 2, and 25 embedded elements. b) Applied displacement boundary conditions for the embedded element models.

The simulations were run using the dynamic explicit solver in Abaqus with the boundary conditions shown in Figure 3- 9b. Abaqus was chosen as a representative commercial code because it is widely used industry and it is the most familiar commercial code to the authors. The boundary conditions an applied displacement 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. The results of these simulations are discussed in Section 3.5.

3.3.3 Method of Volume Redundancy Correction

Garimella et al. [40] suggested a way to correct the volume redundancy in the algorithm for dynamic explicit finite element analysis. Adding this correction to the code itself rather than modifying the material properties (via a user subroutine or otherwise) is attractive because then the

true fiber and matrix properties can directly be entered in the model without the in-between step of creating an effective matrix model (which requires knowledge of the overall composite response).

The illustrations in Figure 3- 10 demonstrate how the volume correction works. During mass calculation, the mass of the host that is occupied by the embedded (fiber) element is subtracted from the total host element mass. To correct for the energy redundancy, the strain energy of the host element is integrated over the embedded element volume and subtracted from the total host element energy. In practice, this is done by calculating the equivalent internal forces of these integrals and distributing them to the host element nodes as discussed in the section on Interaction of Forces.

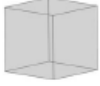





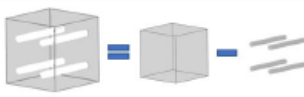
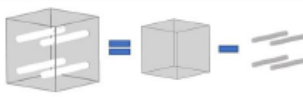


Calculate the mass of the total host volume: m_{Host}		Calculate host element nodal forces from the internal energy of the total host volume: f_{Host}	
Calculate embedded elements' nodal mass: m_{Fiber}		Calculate embedded elements' nodal forces from the internal energy of the embedded element volume: f_{Fiber}	
Calculate the mass of embedded elements' volume occupied by the host material: $m_{H/F}$		Calculate embedded elements' nodal forces from the internal energy of the host element material in the embedded volume: $f_{H/F}$	
Subtract $m_{H/F}$ from nodal force m_{Host} to get the host element mass for the corrected volume: m_{H_Eff}		Subtract $f_{H/F}$ from f_{Host} to get the host element nodal forces for the corrected host element volume: f_{H_Eff}	
Find the total system equivalent mass by adding m_{H_Eff} and m_{Fiber}		Find the total system equivalent forces on the host element nodes by adding f_{H_Eff} and f_{Fiber}	

Figure 3- 10: Algorithm suggested by Garimella et al. [40] to address mass and force redundancy.

3.4 Verification of FFlagSHyP Modified

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 and Wood's FFlagSHyP code that was written to go along with their textbook. When this work started only the static implicit

version of FFlagSHyP was available, so the structure and functions used by FFlagSHyP were built upon to create a dynamic explicit solver that could be modified to include the volume redundancy correction. The algorithm suggested by Garimella was implemented so that the user can choose to use the correction or run the standard dynamic explicit algorithm without the new volume correction code. The resulting FE code will be referred to as FFlagSHyP Modified (FM). When the volume correction feature is in use, the code will be referred to as FM Corrected.

To validate this code, 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 the same response. As shown previously, this is not the case for the Abaqus models. However, the results in Figure 3- 11 show that with the volume redundancy correction algorithm turned on, our code was able to reproduce the results for a single solid element when the truss and the host were the same materials.

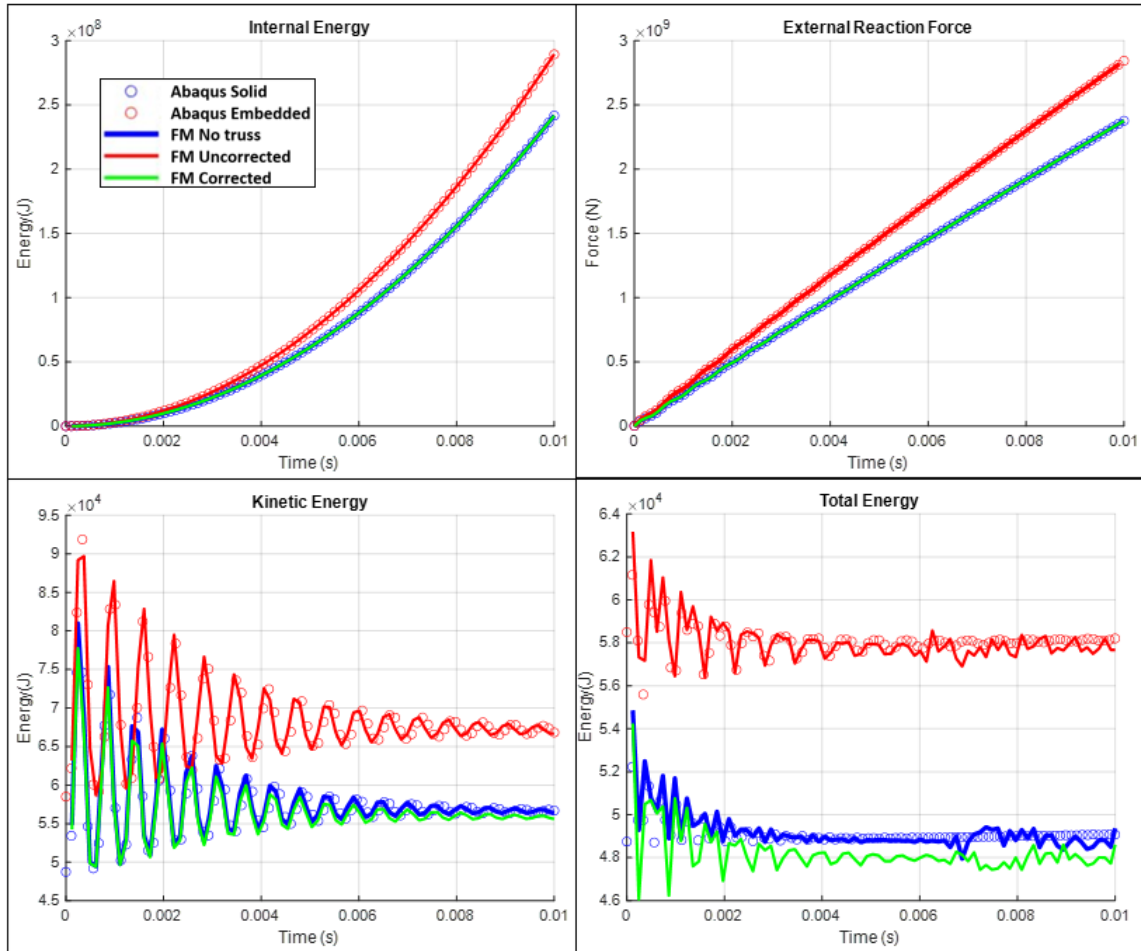


Figure 3- 11: Raw energy data comparison from FFlagSHyP Modified verification tests. A solid element is compared with an identical model with the addition of an embedded element. The embedded element was assigned the same material properties as the host element so that the system equivalent behavior 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 can remove this extra energy and returns the same results as a solid cube.

3.5 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 3- 12 shows a plot of the strain energy of each model as a function of displacement. For a given displacement, adding

more embedded elements increases the amount of stored internal energy. Figure 3- 13 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 artificial viscous dissipation that is added for solution stability also increases with increasing 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.

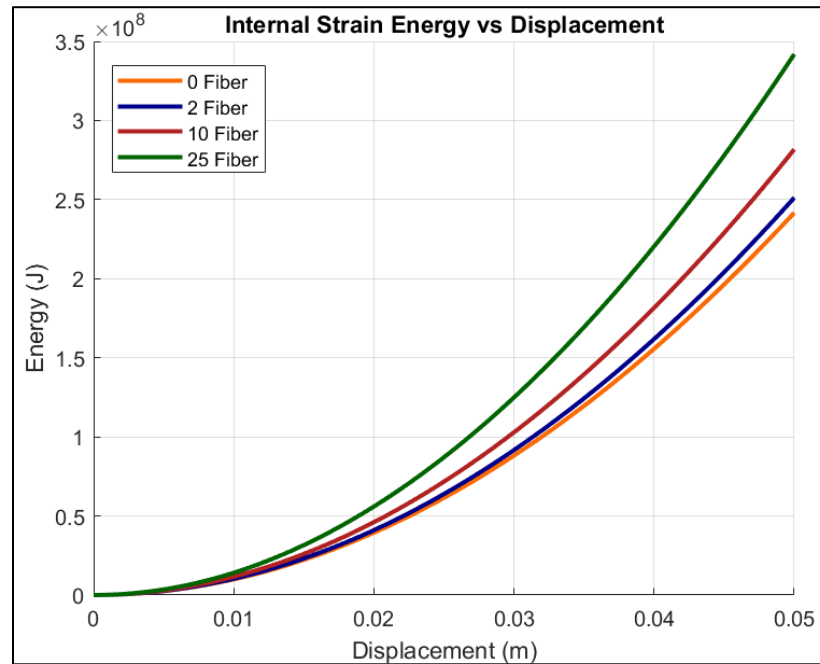


Figure 3- 12: Internal strain energy plotted with displacement of the 0 fiber, 2 fiber, 10 fiber, and 25 fiber models with an applied displacement boundary condition. Although all the models should be equivalent systems, adding embedded elements increases the amount of internal strain energy.

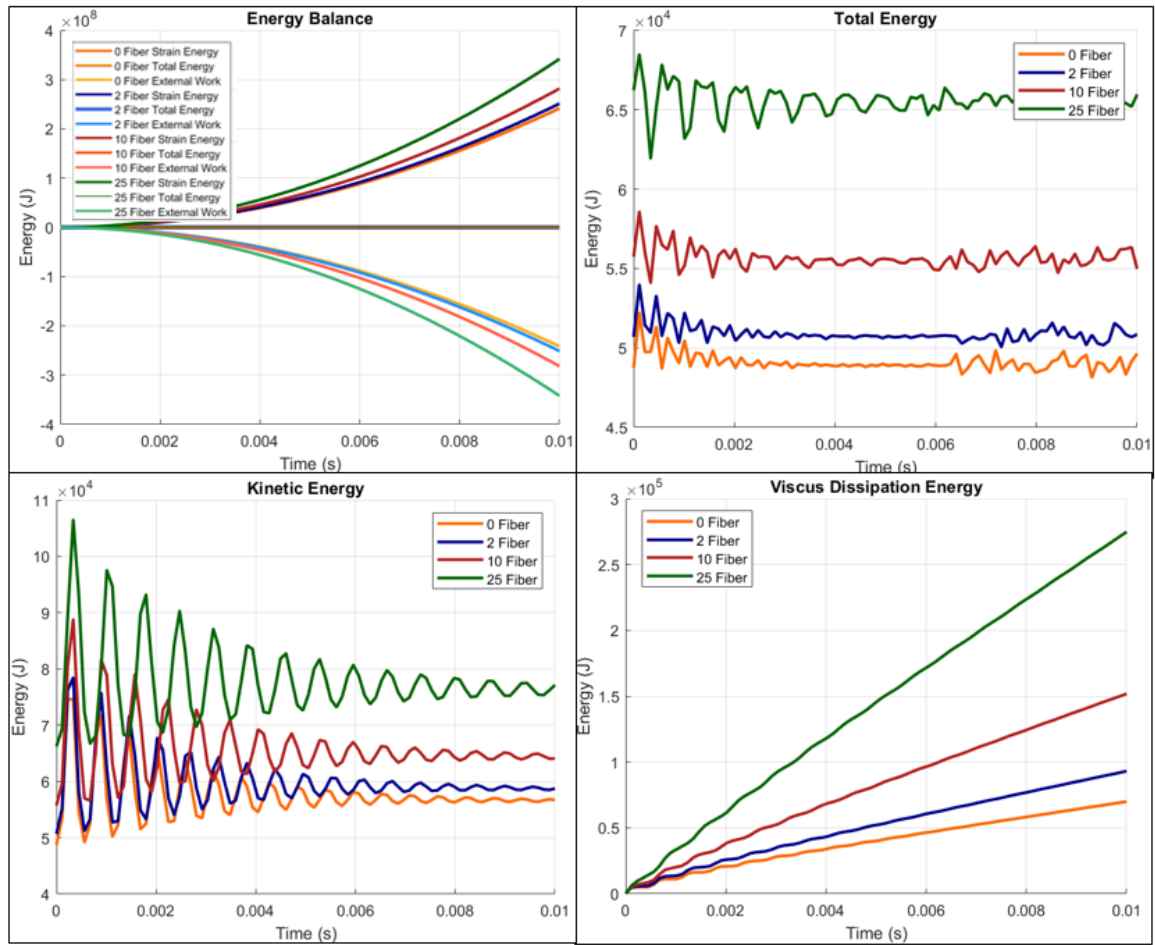


Figure 3-13: Energy plots from the 0 fiber, 2 fiber, 10 fiber, and 25 fiber analyses with the applied displacement boundary condition. The addition of embedded elements affects all types of energies.

3.5.1 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 in the kinetic energy terms. 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.

Table 3- 1: Fiber Volume Fraction Dependency Tests

	Number of Truss Elements	Volume Fraction	Total Mass (kg)	Applied Displacement (m)	Strain Rate (1/s)
Abaqus	0	0	7800	0.05	5
	2	0.04	8112	0.05	5
	10	0.2	9360	0.05	5
	25	0.5	11700	0.05	5
FLagSHyP	0	0	7800	0.05	5
	2	0.04	8112	0.05	5
	10	0.2	9360	0.05	5
	25	0.5	11700	0.05	5
FLagSHyP Corrected	2	0.04	8112	0.05	5
	10	0.2	9360	0.05	5
	25	0.5	11700	0.05	5

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 run in FM again, but this time with the volume redundancy correction turned on. Table 3- 1 shows all the test cases. In order to compare the effect of the embedded elements, the internal and kinetic energy of each model was compared with the solid element model. 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 Figure 3- 14. 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 where FM's volume redundancy correction was used. As shown in Figure 3- 14, the correction nearly completely eliminated the error and internal energy and drastically reduced the kinetic energy error. The remaining error is likely due to other small numerical errors associated with the central difference algorithm and the small size of the test problem being used. 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.

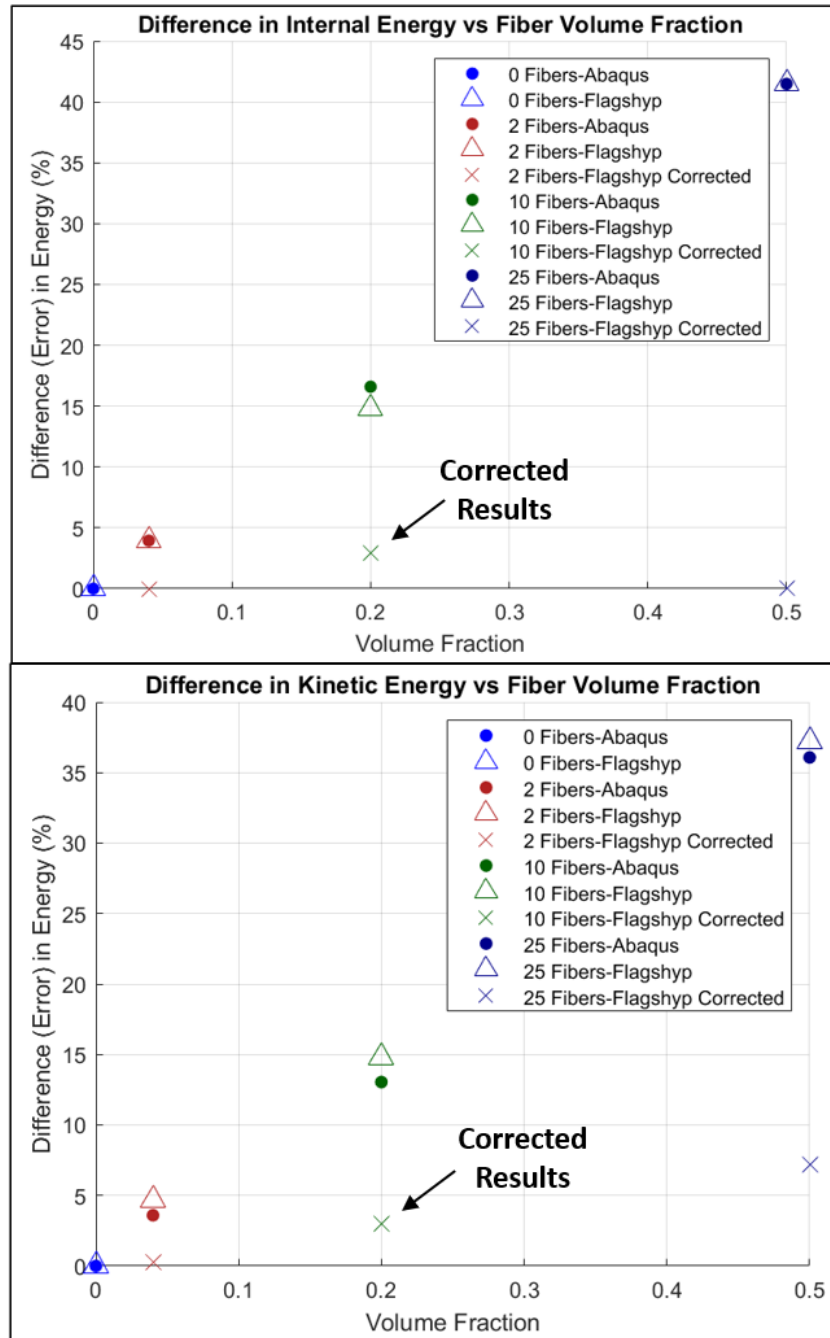


Figure 3- 14: Comparison of internal and kinetic energy from Abaqus and FM embedded element models with different fiber volume fractions with reference to the energy of solid element cases with no embedded elements. Increasing the fiber volume fraction increases the amount of volume redundancy and increases the error in the system. FM Corrected drastically reduces the errors in energy.

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 result in 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 in FM, but with volume redundancy correction. Table 3- 2 lists all of the test cases. Internal energy was expected to remain constant since the total displacement was unchanged. Kinetic energy was expected to increase. Similar to the volume fraction cases, each embedded element model was compared with its' system 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 Figure 3- 15. Again, FM 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. This could be because the change in velocity was not relatively large.

Table 3- 2: Strain Rate Dependency Tests

	Number of Truss Elements	Volume Fraction	Total Mass (kg)	Applied Displacement (m)	Applied Strain Rate (1/s)
Abaqus	0	0	7800	0.05	5
		0	7800	0.05	25
		0	7800	0.05	50
		0	7800	0.05	200
	25	0.5	11700	0.05	5
		0.5	11700	0.05	25
		0.5	11700	0.05	50
		0.5	11700	0.05	200
FLagSHyP	0	0	7800	0.05	5
		0	7800	0.05	25
		0	7800	0.05	50
		0	7800	0.05	200
	25	0.5	11700	0.05	5
		0.5	11700	0.05	25
		0.5	11700	0.05	50
		0.5	11700	0.05	200
FLagSHyP Corrected	25	0.5	11700	0.05	5
		0.5	11700	0.05	25
		0.5	11700	0.05	50
		0.5	11700	0.05	200

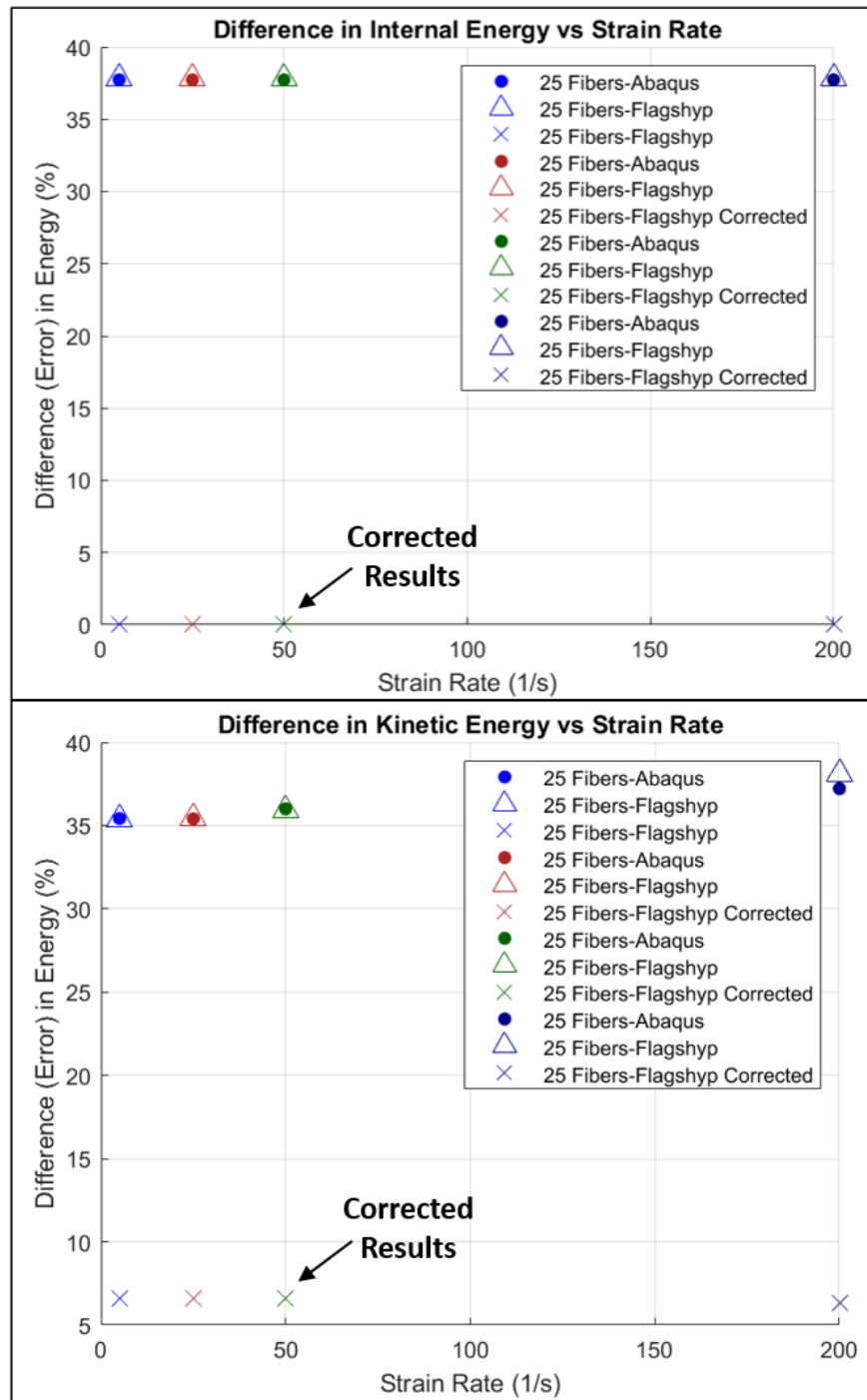


Figure 3- 15: Comparison of internal and kinetic energy from Abaqus and FM models with twenty-five embedded elements with reference to the energy of a solid element with no embedded elements. The additional strain energy from the volume redundancy is constant with changes in strain rate as expected and 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 high strain rates. FM Corrected can bring the error in energy down to less than 10% for all velocities.

3.6 Validation with Multiple Host Elements

The next stage in the work on FM was validating the code for problems with larger numbers of elements. A cube meshed with 8 hexahedral elements with 4 embedded truss elements was tested in both Abaqus and FM. Figure 3- 16 shows the layout of the elements in the model. Notably, the truss elements span multiple host hex elements. It is important for FM to be able to handle embedded elements crossing multiple hosts to allow for easy generation of larger models. Some finite element codes will remesh embedded elements so that embedded elements do not cross host element boundaries. To avoid the difficulties associated with remeshing, FM allows embedded trusses to be associated with two host elements. Displacements of the nodes are found in the same way as normal. The magnitudes of the forces from the embedded elements to the host elements are determined by the percentage of the percentage of the total embedded element located in that specific host element. The 8 host, 4 truss model was run with three loading conditions: tension, compression, and shear. The tension and compression results are correct when compared between the two cases. Characteristic results from these models are shown in Figure 3- 16.

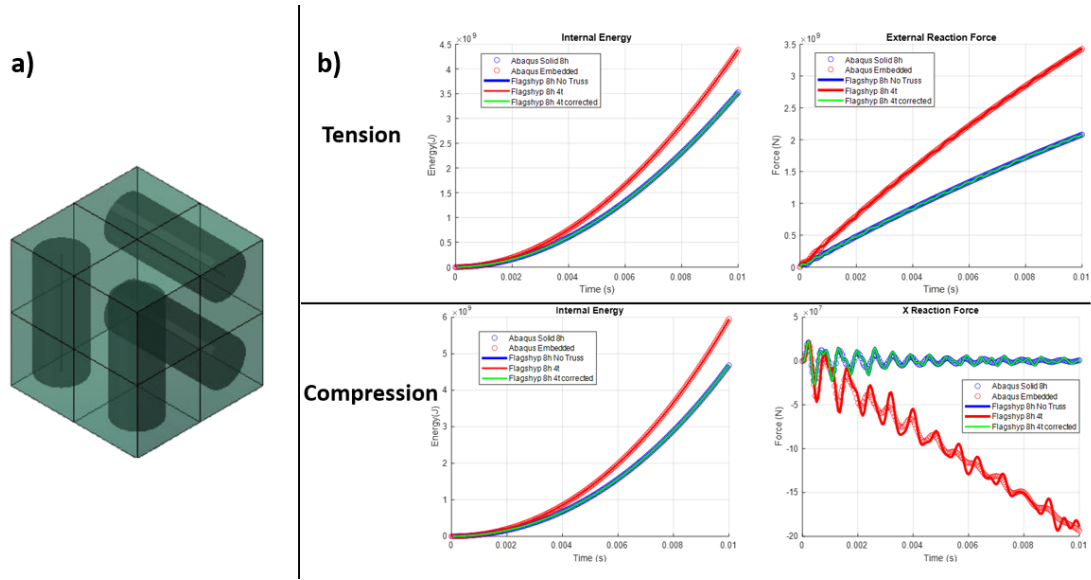


Figure 3- 16: a) 8 hexahedral elements with the 4 embedded truss elements b) Abaqus and FFlagSHyP results for the 8 host and 4 embedded element models for tension and compression tests.

3.7 Tension/Compression Comparison to Experiment

One of the draws of using the embedded element method for modeling UHMWPE composites is the ability to scale the mesh size up and down to create models at multiple scales while using the same modeling method and material definitions. By using denser embedded element distributions to represent UHMWPE fibers, various deformation mechanisms can be more clearly represented when comparing to experimental data. Additionally, by eliminating volume redundancy, the material properties used will be closer to the true properties of the fiber and matrix materials while still producing accurate predictions of material behavior. In this work, we focus on exploring how sensitive the overall stress response of a model is to the number of truss elements used to represent the fibers in the material. This will give insight into the ability of this method to function as a multiscale modeling technique.

3.7.1 Literature Data

As part of the preliminary testing for using the embedded element method to model UHMWPE composites, experimental data on two types of quasi-static material tests of Dyneema® HB26 were chosen. The papers of Russell et al. [66] and Attwood et al. [12] both used the same type of Dyneema® and are widely cited in the Dyneema® modeling community. Russell et al. [66] ran tensile tests of custom tension sample shown in Figure 3- 17 at strain rates of 10^{-2} - 10^{-4} 1/s. Attwood's paper [12] was focused on the indirect tension mechanism discovered due to the interactions of the composite's cross plies in compression, so while this paper included tension test data, we chose to compare to the through thickness compression test data. Finite element models, shown in Figure 3- 17, of each of these sample were generated using Abaqus. Only the gauge section of the tensile specimen was modeled for simplicity. A mesh convergence study was conducted in Abaqus to determine the smallest converged mesh, which are shown in Figure 3- 17.

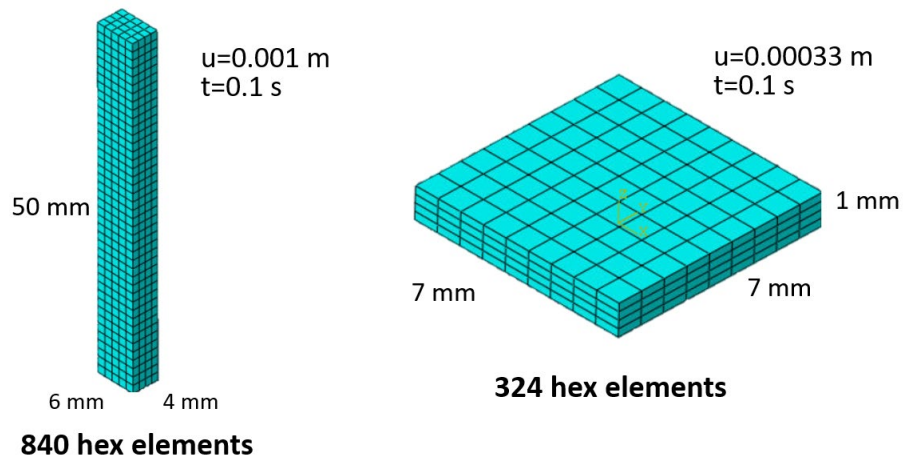


Figure 3- 17: Finite element models with boundary conditions generated for this study.

3.7.2 Embedding Elements

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. The displacement degrees of freedom of the embedded elements are 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 \mathbf{N} . For the embedded element method, this same interpolation can be used to find the displacement of the embedded nodes. This is represented by Equation 3-6.

$$\mathbf{u}_i = \mathbf{N}_H^a(\xi_i)\mathbf{u}_H^a \quad \text{Eq. 3-6}$$

where \mathbf{u}_i is the displacement of the embedded node, ξ_i is the coordinates of the embedded node in the isoparametric space of the host element, and \mathbf{u}_H^a are the known displacements of the host element nodes. Figure 3- 18 shows a two-dimensional equivalent of the scenario: the host element with nodal coordinates \mathbf{x}_H and embedded node (located at \mathbf{x}_i) are shown on the right as they are 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 3-6 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.

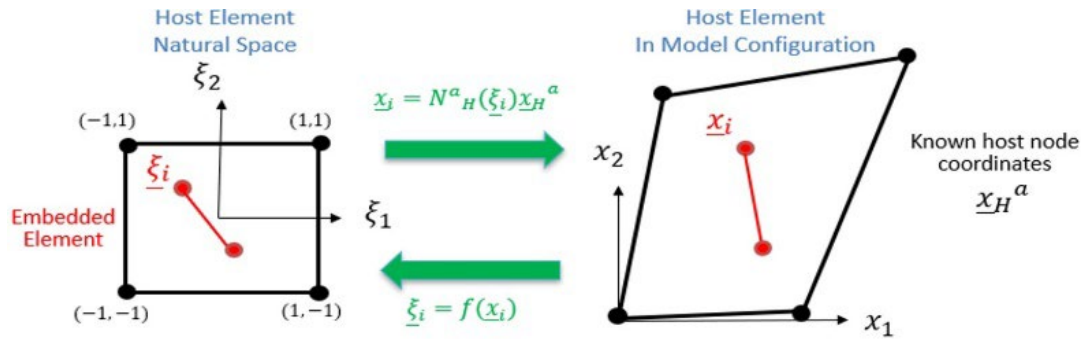


Figure 3- 18: Definition of variables in the mapping between host element natural space and the model global coordinates.

Embedded truss elements were added to these models using a custom MATLAB script. The size of the embedded truss elements were determined by assigning a number of fibers to be represented by a single truss element, similar to the process in Utomo et al. [4]. The cross-sectional area of a single truss element was then determined by multiplying the cross-sectional area of a single fiber (diameter of 17 microns) with the number of fibers per truss. The MATLAB script then stacks truss elements in $0^\circ/90^\circ$ layers based on the outline of the shape. Trusses are not placed more than $\frac{1}{2}$ a truss element diameter away from the edge of the part to avoid trusses that fall outside of the part. An example of several embedded element meshes are shown in Figure 3- 19. The non-ideal stacking sequence used leaves the fiber volume fraction lower than that of the true composite, and fiber volume fraction increases slightly as the number of fibers per truss increases (and the total number of truss elements decreases). In this study, we are investigating how well this method does with different truss mesh densities, correcting the volume fraction will be addressed in later work.

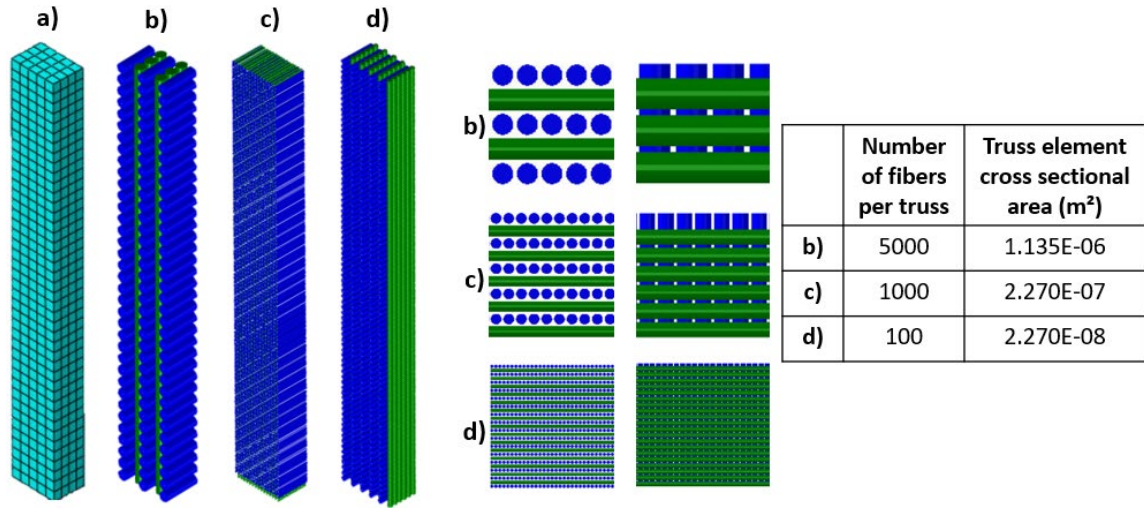


Figure 3- 19: Examples of truss element stacking in the tension test model. A) shows the matrix material mesh used for all embedded element models. The truss meshes were made by assigning b) 5,000, c) 1,000, and d) 100 fibers per truss. The cross-sectional areas reflect the equivalent cross-sectional area of the total number of fibers.

3.7.4 Material Properties

Ideally, the embedded element method would allow for the true properties of the matrix and fiber materials to be explicitly used in the meshes that represent those respective constituents. Many people have tested the stiffness of individual fibers in various different ways[4], [67]–[69], but since the matrix is small percentage of the total composite, there is no data on that material's response alone. To get material data for our finite element models, a literature review was conducted and the basic material properties from different experiments and finite element models were compiled to get an average of the acceptable material properties. Most finite element models are homogenized orthotropic models, but a general idea of the stiffness of the components can be backed out by assuming that the stiffness in the fiber directions is dominated by the fibers, and the stiffness in the third direction is solely a result of the matrix stiffness. Using these criteria, we compiled data from several sources[4], [11]–[13], [67], [70] and decided on a starting material for our model. From this starting material, the stiffness of the fibers and matrix material were adjusted

independently until a good fit to the experimental data was achieved. The results of this testing are shown in Figure 3- 20, and material 5 ($E_f = 180 \text{ GPa}$, $\nu_f = 0.49$, $E_m = 1.1 \text{ GPa}$, $\nu_m = 0.3$) was chosen as the best fit. Experiments have shown that Dyneema® fibers are orthotropic[71], [72], but here we are simplifying them to an isotropic material so that they can be modeled with the one-dimensional truss elements.

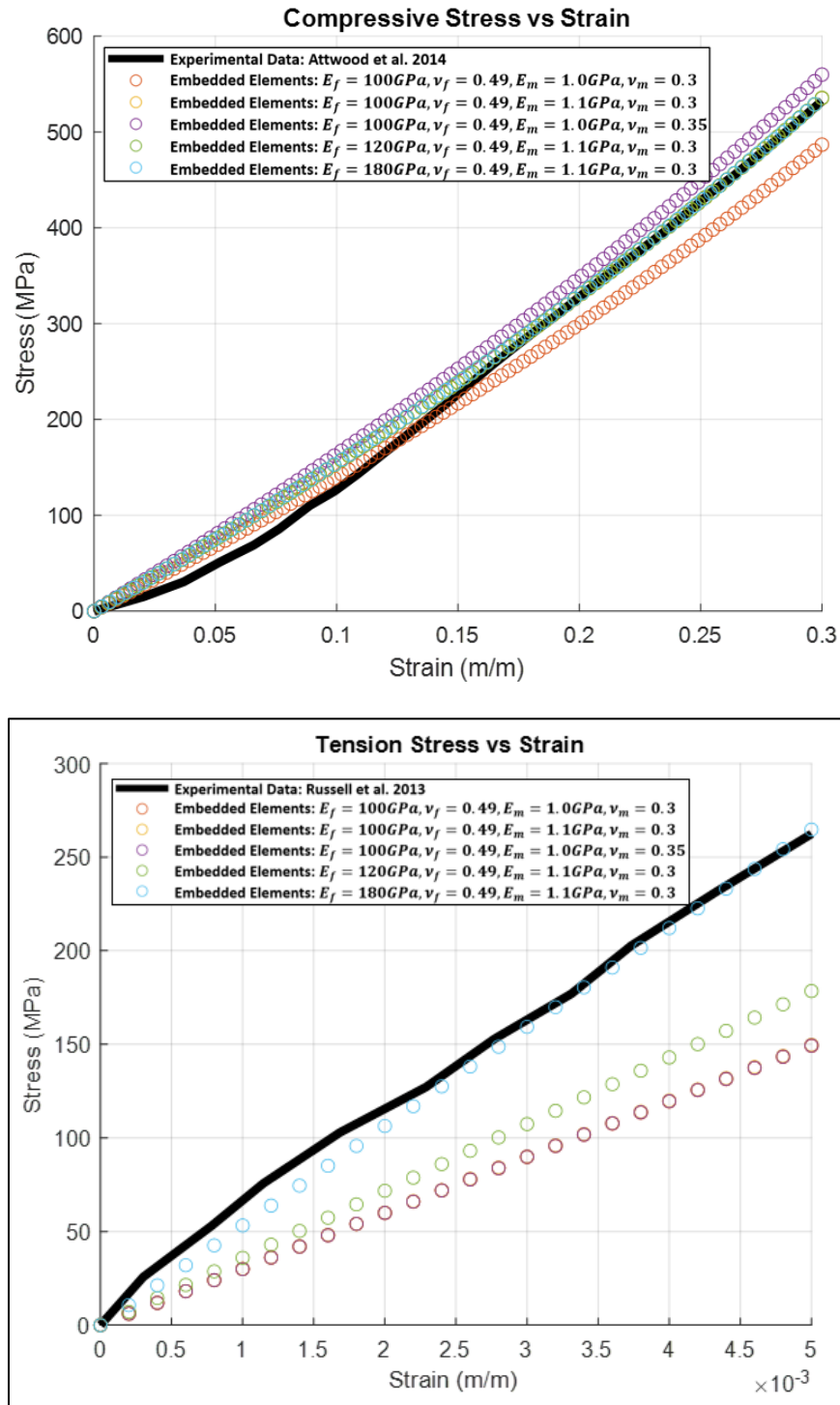


Figure 3- 20: Overall model response of the embedded finite element models for the tension and compression tests when a consistent truss element size and varying the truss and matrix material behavior.

After selecting a material, the number of fibers assigned per truss was adjusted for both the tension and compression models to determine the influence on the density of truss elements on the models' response. Initially, the number of fibers per truss was 12: this gave each truss layer a thickness of about 60 microns, which is the laminate thickness of Dyneema® HB26. From there, it was increased to 100, 500, 1000, and then 5000, and increasing the number of fibers per truss decreased the total number of truss elements allowing for faster computational time. If the same response is achieved for 5000 fibers per truss as 12, it would show the scalability of this method for easy multiscale modeling.

3.7.5 Results and Discussion

When adjusting the material properties, the tension model was only dependent on the fiber stiffness, while the compression response mainly depended on the matrix material. This is what others have observed, although we were hoping to get more of the indirect tension mechanism discussed by Attwood[12] in compression. We found that to get a good representation of the composite tensile response, the fiber stiffness needed to be somewhat higher than is reported in the literature. This is because the truss embedding algorithm, we used does not optimally stack the truss elements, so the fiber volume ratio of our models was closer to 60% rather than the 83% of the true composite. Currently there is space allowed between the edges of the part and the mathematical volume of the truss element to avoid truss elements that extend beyond the edge of the part. This space can be removed with the knowledge that some parts of the truss will be outside of the part, but still be used in the volume redundancy elevation algorithm. Since the current volume fraction is so low, this may be acceptable. Also, since in real Dyneema®, fiber bundles are

compressed to be not perfect cylinders and push the packing efficiency much higher, so this would be a way to artificially increase the packing efficiency.

Another method is to implement a $[0^\circ/90^\circ]$ stacking pattern of the fibers where each layer is offset to allow for higher packing efficiently. Figure 3- 21 shows three different ways of packing the truss elements, the original way used in this work, a version where the space between fibers is eliminated, and the $[0^\circ/90^\circ]$. The reduced spacing version gives a fiber volume fraction of 78%, which is much closer to the fiber volume fraction of Dyneema®. The $[0^\circ/90^\circ]$ gives an even higher packing efficiency, but with the disadvantage of the fiber layers being thicker and the 0 and 90 layers less evenly distributed. This fiber embedding algorithm will be updated to use one of the latter two packing mechanisms for future work.

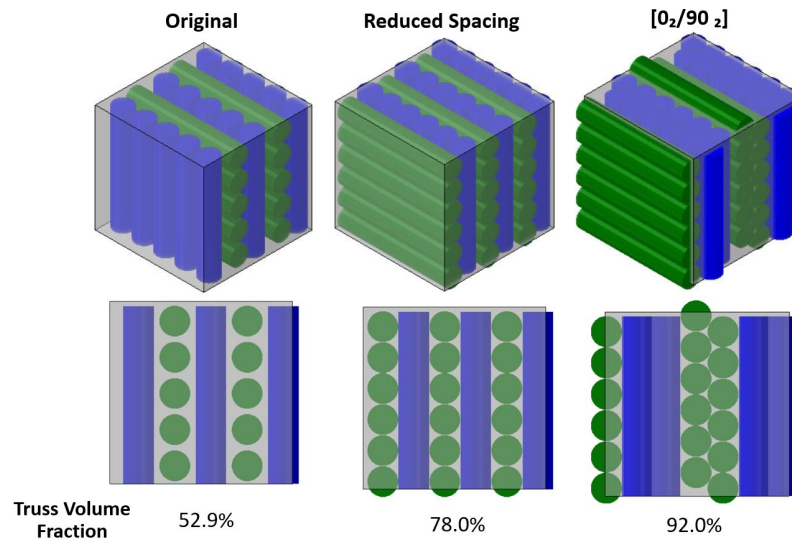


Figure 3- 21: Different packing arrangements for embedded truss elements in a cube of material. The original arrangement, used in this work, only gives a fiber volume fraction of 52.9%, which is much less than the typical 83% found in Dyneema®. By reducing the buffer space between the trusses, and by changing the stacking arrangement to $[0^\circ/90^\circ]$ increases the fiber volume fraction to 78.0% and 92.0%, respectively. One of these latter two packing arrangements will be used in future work.

An exciting result, shown in Figure 3- 22, is that when the material properties and fiber volume fraction are kept constant, changing the number of fibers represented by each truss element has almost no effect on the total model response. Since the volume occupied by the truss elements remains relatively constant, and the stiffness per volume is also constant, it makes sense that the response is unchanged. This is promising for larger models, such as full plate impact problems. The material properties for the embedded element method can be verified at a smaller experimental scale and then applied to a larger model with fewer truss element representing more fibers for a more efficient run time. The same material properties can also be used for micro and macro scale modeling.

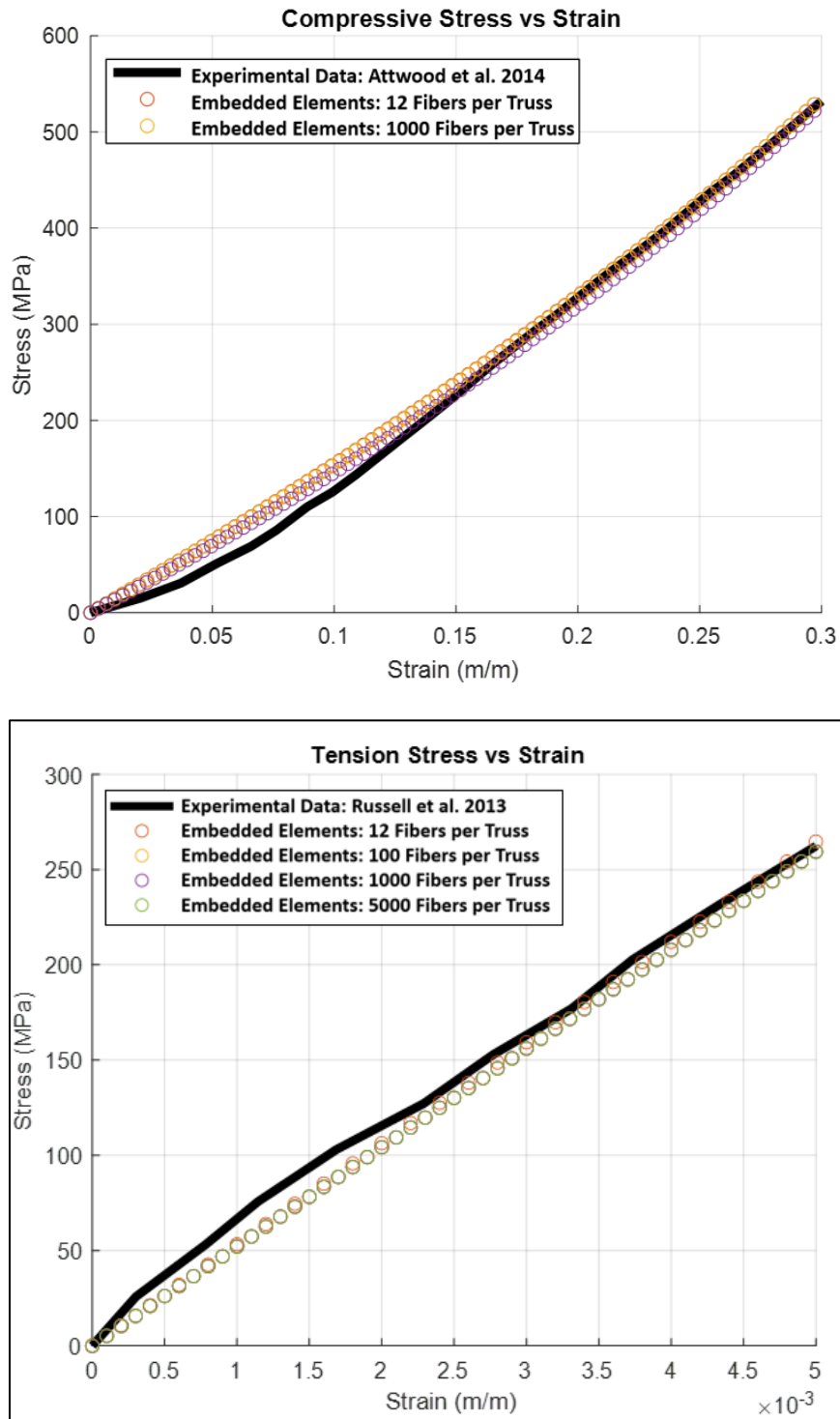


Figure 3- 22 : When the truss and matrix element material is kept constant, changing the cross-sectional area and number of truss elements does not significantly affect the model response if the truss volume fraction is kept relatively constant.

3.8 Speed Updates

Since FLaSHyP Modified is based on a MATLAB program written with the primary purpose of teaching finite element analysis theory and not for efficiently running large models, FM is not a fast-running program. It works well as proof-of-concept tool for the volume corrected embedded element method, but we would like to test the method on larger scale models. To make this feasible the code needed to be sped up. First, all global variables were removed and replaced with variables that are passed only to functions that need them. There were a few variables and structures that did not have pre-allocated sizes; this was corrected. With these simple updates, the code became nearly 10 times faster, but still too slow to run large jobs. Next, the run and time profiler built into MATLAB was used to identify the functions that took the most time to run. This led to the discovery that some functions were being called more often than they should. Removing this oversight approximately halved the run time.

With these updates, the Attwood compression test discussed in the previous section was run in FM with a total of 1820 embedded truss elements. The total time to be modeled was 0.1 s. After a run time of 36 hours only about 40% of the simulation was completed. The next easiest way to speed up the code was to implement MATLAB's parallel computing capabilities. By replacing a for loop with the command `parfor`, MATLAB will run the loop in parallel on available separate worker computers. There are some conditions on the types of variables that can be used in such a loop, so some edits needed to be made, but the `parfor` loop was added to the code during the element internal force calculation step where it must loop through all elements, calculate their internal force, and distribute internal forces of the truss elements to their connected hosts. This is the most repeated time intensive activity. With the parallelization, the compression simulation ran to completion in

6.9 hours. **Figure 3- 23** shows the results of FFlagSHyP before and after the speed updates compared with the results from Abaqus.

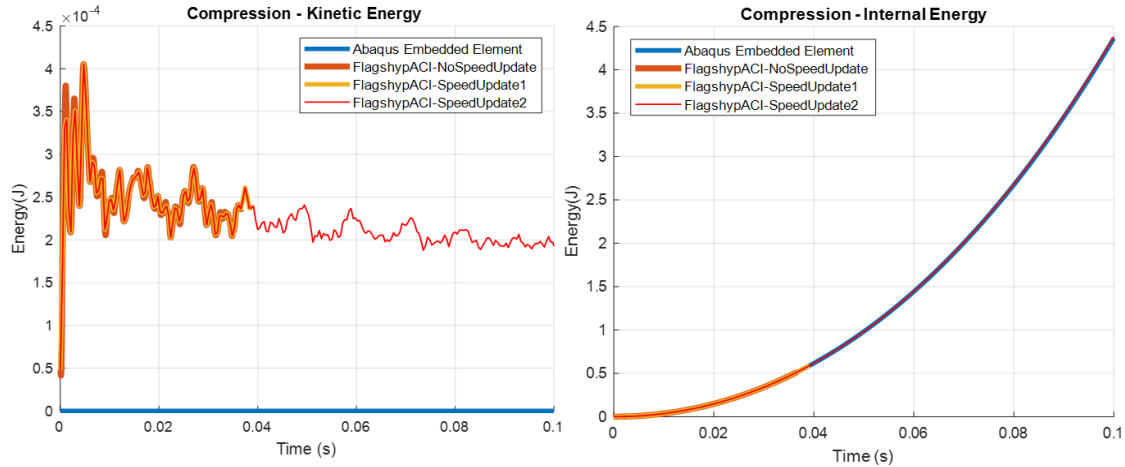


Figure 3- 23: Comparison of the progress of an embedded element compression model in FFlagSHyP where run time was capped at 36 hours. A small amount of mass scaling was used in the FFlagSHyP model.

This timeframe was still unacceptable, so we returned to single core processing to work on speeding up the base code. The MATLAB Profiler tool was used to identify the 5 functions that ran for the longest amount of time. These functions were:

Gradients – Used in stress calculations and involves the time-consuming operation of inverting two matrices.

InternalForce_explicit – Runs inside of a loop to calculate the internal force applied to the nodes of each element. Time consuming parts of this function include calling gradients and kinematics_gauss_points, as well as a complicated loop to identify embedded elements associated with a host.

Calc_element_size – Finds the largest and smallest side lengths of an element. This function is also run in loop to find the smallest characteristic element length for the stable time increment calculation.

Kinematics_gauss_points – All this function does is pull data from a structure called Kinematics. There is no real way to speed this process up, we can only look for places where it is being called too often.

TrussCorrectedInternalForce_explicit – This function calculates the internal force from embedded truss elements and distributes it to the nodes of the host element. This is done for each host element, so the forces in each embedded element end up getting calculated multiple times (since truss elements can span two host elements).

Once the functions were identified, the following edits were made to the code to help improve efficiency and speed:

Gradients - Calculated and saved initial gradients instead of recalculating every time step.

Additionally, the stress data for all elements is now saved in each step and the saved data is used for the output write step (previously the output writing recalculated stress for each element, which includes running gradients). This reduces the number of times gradients is called.

InternalForce_explicit - Created a cell based structure to hold lists of the embedded elements associated with each host instead of searching through the embedded element-host list every time InternalForce_explicit is called.

Calc_element_size – Replaced this function with calc_min_element_size which finds only the smallest element length (instead of largest and smallest, since only smallest is needed). The original version of this code also used a complicated looping system that ended up calculating some element side lengths twice. Simplifying this to a direct calculation of each of the 12 side lengths sped this function up by nearly 56%.

Kinematics_gauss_points – Similar to gradients, this function was run less often due to using saved stress data rather than recalculating stresses so often.

TrussCorrectedInternalForce_explicit_from_mem – This function was modified to use the newly saved stress data of the embedded elements to get the internal force instead of recalculating it for every single host element of the embedded element.

With these edits to FM, the total runtime of the entire code was increased by about 49%. With this improvement in the base code, parallel computing was reimplemented. The new speed when using different numbers of processors was compared with the old version of FM, before the speed edits were added. A rectangular tension model with 420 hex elements and 402 embedded truss elements was run with a simulation time of 0.1 sec. The stable time increment was $4.30\text{e-}6$ seconds meaning the total number of steps needed was 23,258. The comparisons between the two versions of the code is shown in Figure 3- 24, where the blue dots are the original speeds and the green stars are the new.

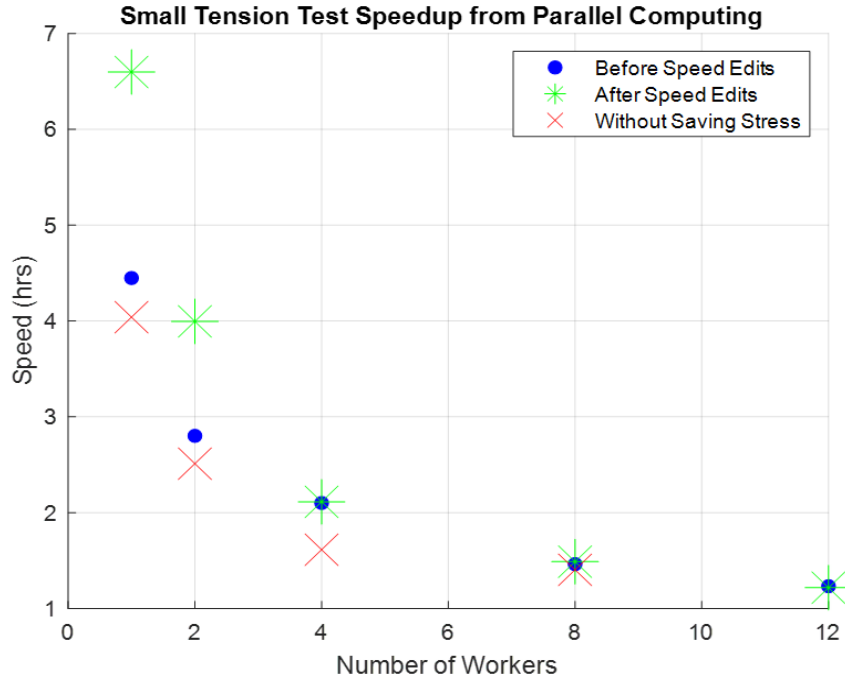


Figure 3- 24: Wallclock time needed to run an 800 element model with embedded elements using different numbers of parallel cpus for before speed edits to FLaGSHyP (blue circles), after edits (green stars), and removing the saved stress data to reduce overhead (red x's).

We found that the speed edited FM with parallel computing was no faster than the old version. This was because some of the new edits created more overhead cost when passing information to multiple processors. For example, instead of recalculating the stress during the output writing function, the stress is now saved in a structure during the main code loop and then passed to the output function. This means that a copy of the stress structure needed to be passed to each processor for them to update the saved stress. When the process of saving the stress at each time step was removed, parallelization did improve the program speed, as shown in Figure 3- 24 by the red x's.

Because the code is not optimized for parallelization, running in parallel did not increase the speed much. FM is still significantly faster than before the effort to increase the speed, but still very slow for models with more than a few hundred elements. Increasing the speed of the code more would at this point require rewriting and restructuring much of the code. For this reason, we will not continue improving the code's efficiency as it is. Rather, the principles of the embedded element method learned from programing FM will be used to incorporate the embedded element method into an existing finite element software developed by our lab that is specifically designed for efficiency in parallel computing. This will be a part of future work by subsequent students on this project.

3.9 Conclusion

It has been shown how incorrect volume and mass affect the energy calculations in the embedded element method, which in turn affects 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 continuum 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.

Additionally, we have shown that accurate models of Dyneema® can be made for quasi static tests using various numbers of embedded truss elements to represent fiber bundles. This means that the method could be used to model Dyneema® on different length scales: from a small specimen used in Split Hopkins Pressure Bar to a full-sized plate impact experiment, but just changing the number of fibers per truss without greatly impacting the overall material response. The embedded element method is more effective when the volume redundancy is corrected and some work has been done to make our embedded element code usable for modeling small material specimens, but the code is just not capable of the speed for it to be feasible. Future work on this method will be to move the elements of FM to a more optimized code. This is further discussed in Chapter 5.

Chapter 4

Finite Element Impact Models

4.1 Steel Plate Models

The main application of this project is to eventually model ballistic impacts on plate armor to improve on current armor designs. While the embedded element method was being established, there was some preliminary work on modeling impacts so that difficulties could be dealt with ahead of time. We had a few image results from an initial ballistic impact experiments that included steel and Dyneema® plates impacted by spherical fragment simulating projectiles (FSP). While actual stress and strain data is not available for these experiments, slow motion video capture provides some reference that can be visually compared to finite element models to show if the models are on the right track.

Before the sizing of the plates was determined, a model of a 10" diameter, ½" thick steel plate being struck with a 3/8" diameter steel sphere was created in Abaqus. The material in the experiment was identified as a mild steel, so the Johnson-Cook material defined in Table 4- 1 was used for the FE material properties. While this was not actually the same material as in the experiments, part of the preliminary modeling is to understand how element erosion and deletion work under these high-rate impact situations. The Johnson-Cook model had already been studied and should behave similarly enough to the real plate to visually compare the two, since stress and strain data for the experiments does not exist.

The plate impact problem was modeled as a quarter symmetry. In later tests, plates are planned to be clamped inside of a ring. This was modeled by constraining the surface of the outside ring of the plate to be constrained in all directions (rotations were left free). This surface is shown

in Figure 4- 1 **Error! Reference source not found.** and corresponds to the amount of plate that will be clamped inside of the testing ring.

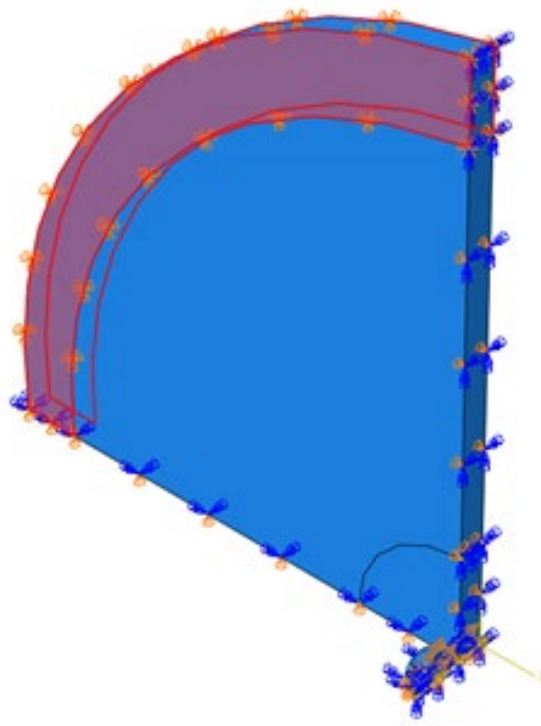


Figure 4- 1: Boundary conditions for the initial test plate problem, based plans for future ballistic impact tests with clamped edges.

The projectile was modeled first with only elastic properties defined. Later, the simulation was run again using a projectile with the same plastic and element deletion properties as the plate material. The results of these simulations are shown in **Figure 4- 2**. Both models show believable deformations and stresses, and the plasticity and element deletion definitions are working as expected.

Table 4- 1: Johnson-Cook flow stress parameters and damage evolution parameters used in the steel plate impact model.

A (MPa)	980
B (MPa)	996
n	0.45
c	0.0024
D1	0.09
D2	0
D3	0
D4	-0.060

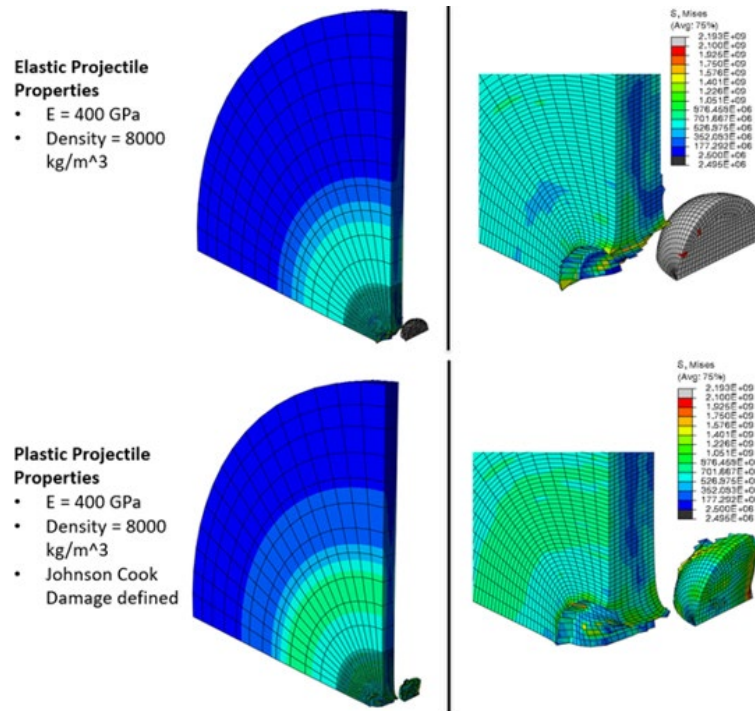


Figure 4- 2: Johnson-Cook material model used in a ballistic model with and without plasticity.

4.1.1 Mesh Convergence

A mesh convergence study was done on the ballistic impact FE model. Choosing criteria for convergence was difficult since we would like to know the stresses in the model

over time and that element deletion can cause numerical noise in the stress response near where the deletion occurs. In the end, the model stress history was pulled from three points near the impact point, shown in **Figure 4- 3** and the peak stresses over time were compared across different meshes. Additionally, the radius of the hole produced by the projectile, the penetration depth, and the percent change in mass of the projectile (due to element deletion in the projectile) were also estimated from the different meshes and compared.

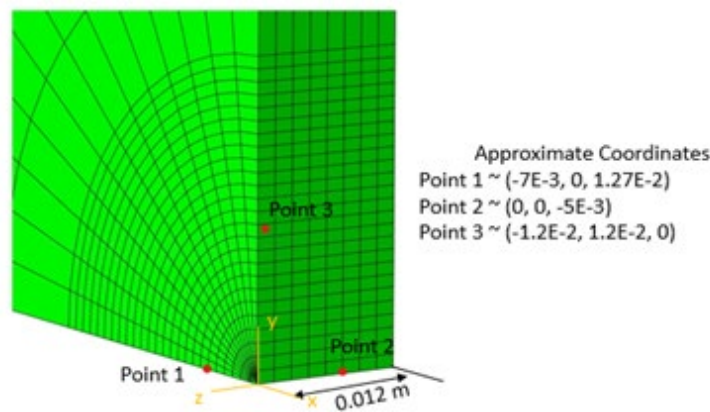


Figure 4- 3: Points used for extraction of stress data in the mesh convergency study.

A range of mesh sizes, beginning at 3,700 elements to nearly 180,000 elements were tested. A few of these meshes are shown in **Figure 4- 4**. The mesh sizes were named for the number of elements through the thickness of the plate (i.e., Mesh16 has 16 elements through the plate depth). The stress results over the time of impact for five different meshes are shown in **Figure 4- 5**.

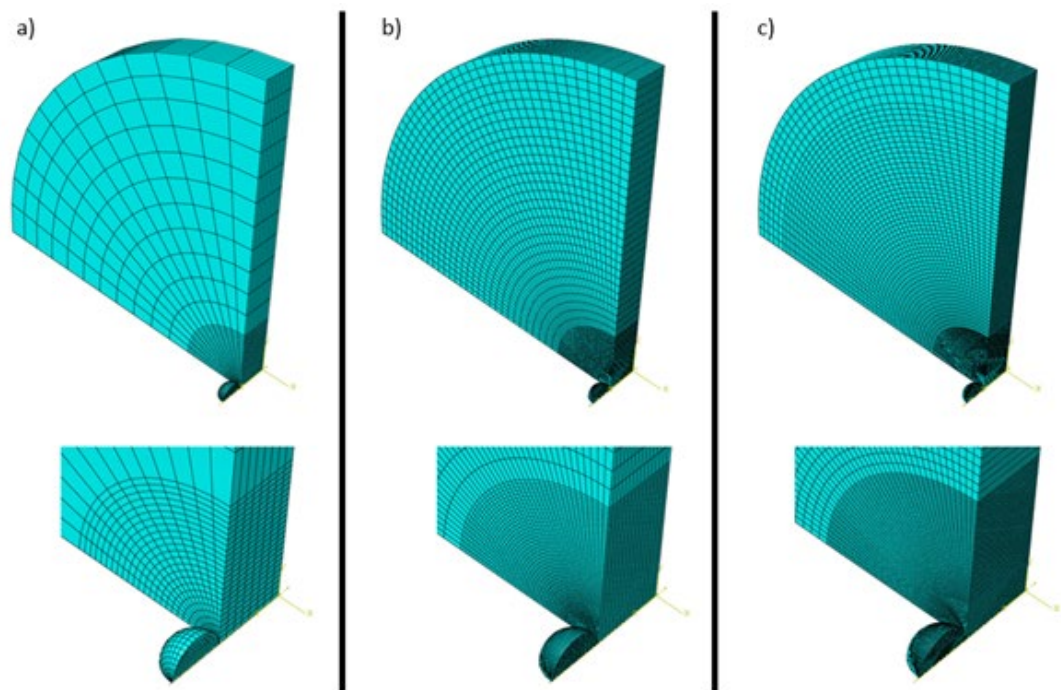


Figure 4- 4: Three of the meshes used in the mesh convergence study. a) Mesh8 (3,700 elements)
b) Mesh16 (?) c) Mesh26 (180,000 elements).

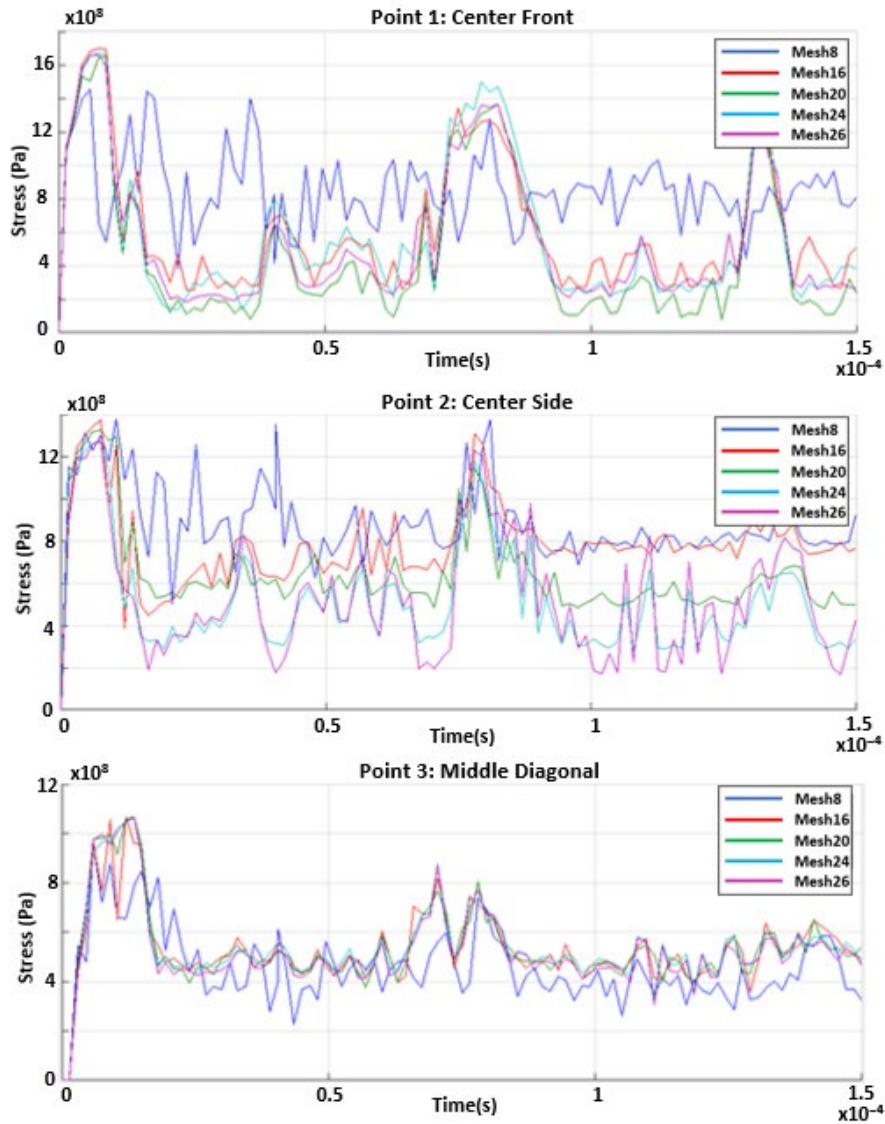


Figure 4- 5: Stresses at Points 1, 2, and 3 (locations shown in **Figure 4- 3**) for the five different mesh sizes. Mesh8 is the coarsest and Mesh 26 is the finest mesh with the most elements.

Mesh 8, the coarsest mesh, is clearly not converged. Further away from the impact (Point 3), the stresses are more consistent and converge more quickly. However, from the closer two points, it seems that the solution does not converge at the times of max stress until Mesh 24, and in between peak stresses there is less agreement. Figure 4- 6 shows the other three metrics of convergence, hole radius, penetration depth, and percent change in projectile mass for all the mesh sizes tested. The changes in these three metrics become minimal by Mesh 16. Based on these

requirements, Mesh20 should provide reasonable accuracy with a less than 3-hour runtime on two computer cores.

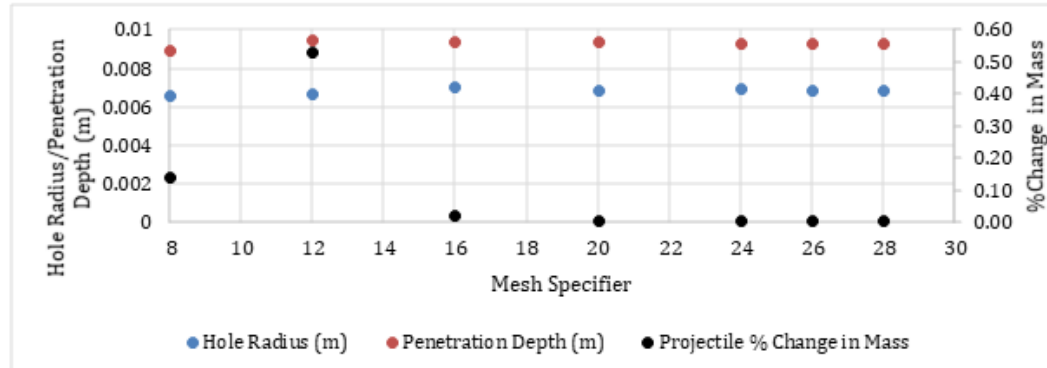


Figure 4- 6: Three other metrics of convergence that were used in the mesh refinement study. Changes in hole radius, penetration depth, and change in projectile mass become minimal in meshes larger than Mesh16.

4.1.2 Mild Steel Plates

After the circular plate model was established, a set of ballistic tests using rectangular steel plates and circular Dyneema® plates was run at Los Alamos National Lab. Little data was collected from these experiments since they mainly served as a test for a larger set of experiments being planned, but the data collected is shown in Table 4- 2. The steel plates used in the experiments were of a mild steel, with a clamp on the bottom surface to hold it in place during impact. These plate impact experiments were modeled in Abaqus as $\frac{1}{2}$ symmetry models shown in Figure 4- 7. The bottom of the plate was pinned in the z direction on both sides to represent the clamp used in the actual test. The $\frac{3}{8}$ " diameter projectile is modeled as a purely elastic tool steel. The material model for the mild steel plate was taken from Iqbal et al. [73] who fitted a Johnson-Cook plasticity and damage model to a similar mild steel using both dynamic tension tests and Kolsky bar testing.

Abaqus requires a fracture energy value to implement element erosion. The ASM Handbook for properties of irons, steels, and high-performance alloys lists the average Charpy impact energy for mild steels as 18-135 J. With a standard Charpy impact specimen cross section of 10x10 mm, that is a fracture energy of 180-1,350 MJ/m². This data was used to find an approximate value for the mild steel Abaqus model. A full list of the material behavior is included in Appendix C. Both the plate and projectile were meshed with linear hexahedral elements. The site of impact has a refined mesh with element size of 1 mm. The experiments included three different shots on three identical plates.

Table 4- 2: Preliminary ballistic impact testing data from Los Alamos National Lab taken November 2019

Shot	Powder load	Target	DIC	Velocity	Penetration
1	150 grains	AR500, 6-in square, 0.5-in thick	-	1145 m/s	no
2	150 grains	Mild steel, 6-in square, 0.25-in thick	yes	1071 m/s	yes
3	150 grains	Dyneema®, 8-in disk, 0.5-in thick	-	N/A	yes
4	50 grains	Dyneema®, 8-in disk, 0.5-in thick	-	291 m/s	no
5	150 grains	Dyneema®, 8-in disk, 0.5-in thick	yes	1065 m/s	yes
6	120 grains	Dyneema®, 8-in disk, 0.5-in thick	yes	N/A	no
7	200 grains	AR500, 6-in square, 0.5-in thick	yes	1376 m/s	yes
8	175 grains	AR500, 6-in square, 0.5-in thick	-	1252 m/s	no
9	185 grains	AR500, 6-in square, 0.5-in thick	-	1284 m/s	no
10	75 grains	Mild steel, 6-in square, 0.25-in thick	-	582 m/s	no
11	75 grains	Mild steel, 6-in square, 0.25-in thick	-	475 m/s	no

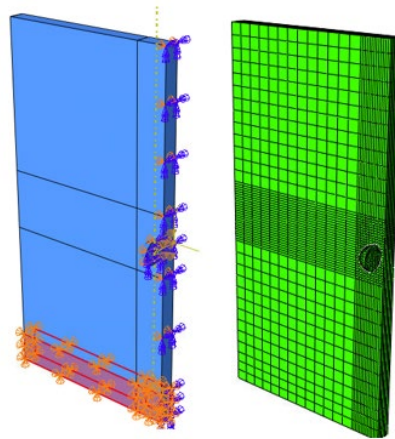


Figure 4- 7: Mild Steel plate Abaqus model, boundary conditions, and mesh.

Shot 2 was simulated by setting the initial velocity of the projectile to -1071 1/s. This speed should penetrate the plate, based on the experimental results. The first attempt at running the simulation was prematurely terminated because the deformation rate was too high. Initially, this was thought to be a fault of the Johnson-Cook model used. The highest strain rate used in this model fit was 1700 1/s; the highest strains in the simulation were on the order of 1×10^6 1/s. However, it turned out that there were only two elements causing the error. Changing the mesh of the model eliminated this problem and the projectile penetrated the plate as the tests showed. Figure 4- 8 shows the stress, plastic equivalent strain, and strain rate during penetration. Unfortunately, the back-face deformation could not be compared to the test since the test video is obstructed after the projectile penetration.

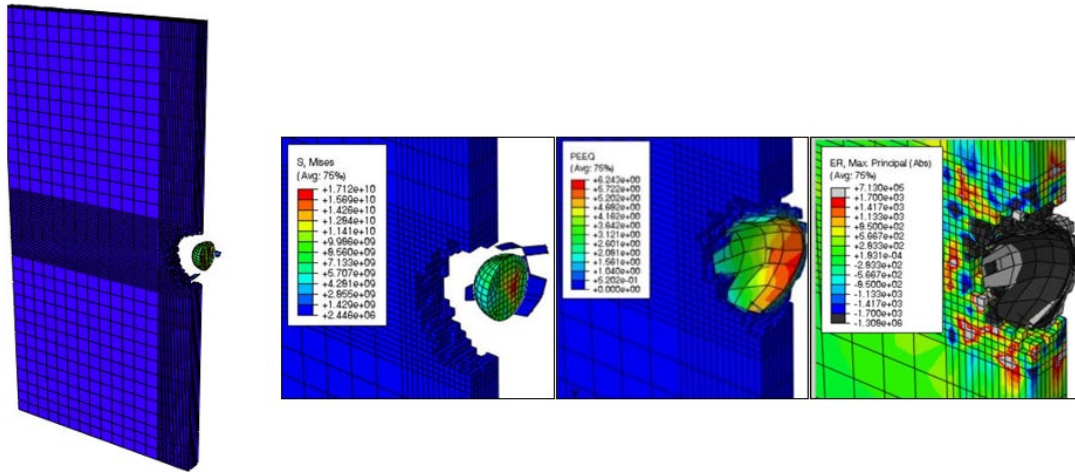


Figure 4- 8: Stress, plastic equivalents stress, and strain rate results from the completed mild steel ballistic model.

The plate size was reduced to a $\frac{1}{4}$ symmetry model, and a mesh refinement study was done to ensure that the mesh would not continue causing problems. The results are shown in Figure 4-9. It appears that the plate meshed with 120,000 elements has the same stress distribution as the largest model, so that mesh was chosen for the rest of the analyses (the projectile stress is different in these two meshes, due to the large number of elements and the difficulty of meshing spheres with hex elements. Mainly we are concerned with modeling the response of the plate which seems generally unaffected by this discrepancy).

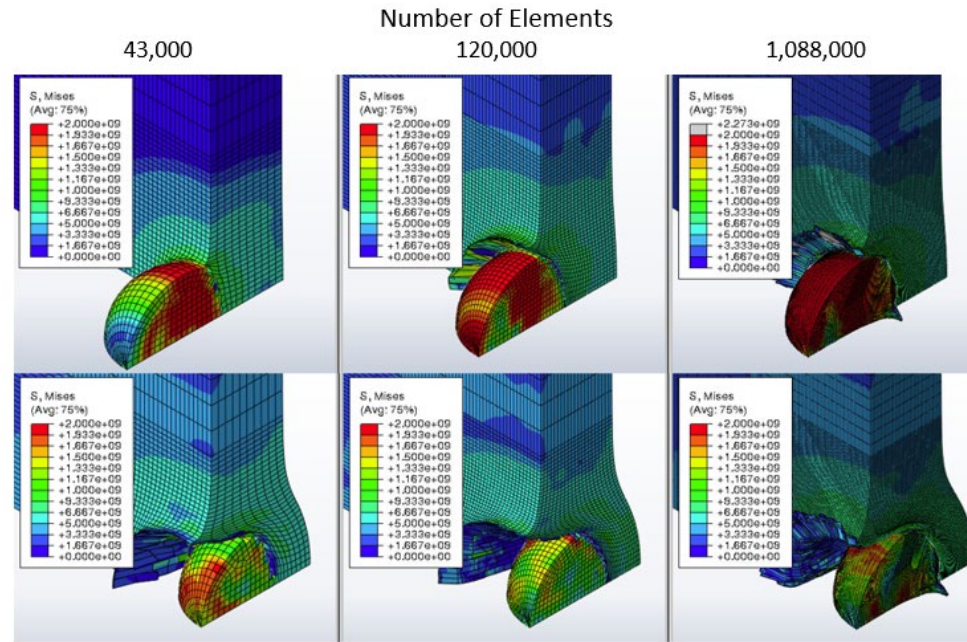


Figure 4- 9: Three different meshes of the steel plate. The top and bottom rows represent two snapshots in time.

Plasticity, damage, and element erosion were then introduced to both the plate and projectile parts. All the parameters for Johnson-Cook stress and failure are in Iqbal et al. [73] (and listed in Appendix C), which was the chosen source of strain rate dependent material properties. However, the fracture energy of the material is not included. This parameter is necessary for element erosion in Abaqus. The fracture energy of a similar metal was found from the result of the Izod impact test data from ASTM to be $180,000 \text{ J/m}^2$ and it was tested in Abaqus by modeling a dynamic dog bone tension test. By slightly adjusting the value of the fracture energy (since the exact material is unknown), the stress strain curves from the model were similar to the paper's experimental results. **Figure 4- 10** shows a graph of the experimental results and the results from our Abaqus model for a few different strain rates. A fracture energy of $180,000 \text{ J/m}^2$ was chosen as the best approximation, although the model does not appear to adequately cover the entire strain range presented in the paper. Potentially this is the case because the Johnson-Cook model is not used to cover such a wide range of strain rates (at least not with constant model parameters).

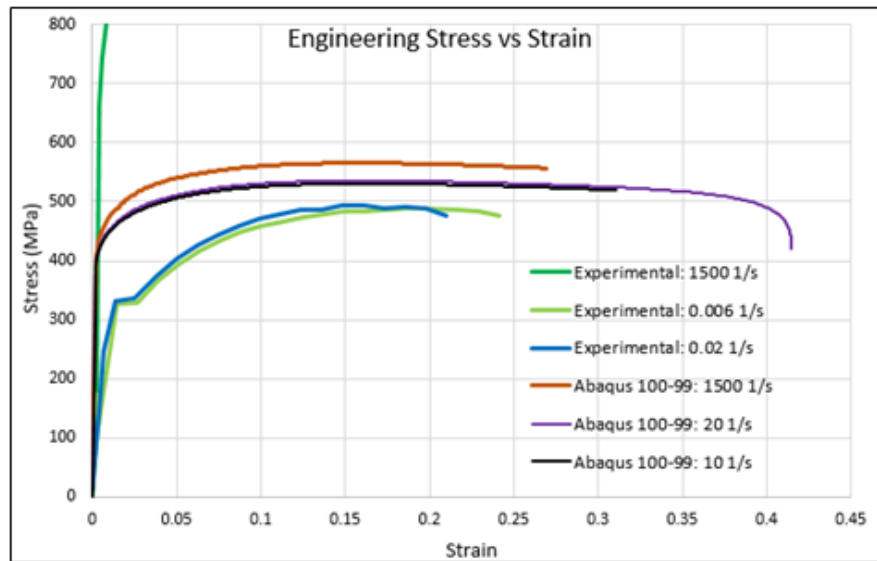


Figure 4- 10: Stress strain results of experimental tensile tests from Iqbal[73] and tensile tests that were simulated in Abaqus at different strain rates.

4.1.3 Element Erosion

Despite this Johnson-Cook model not being exact, it was used to test ballistic impact model. Abaqus does not distribute the mass of eroded elements to adjacent nodes, so element erosion destroys some mass. An investigation of how element erosion affects the model mass was conducted. The amount of mass lost is dependent on the quality of the mesh. In a fine mesh, each element contains less mass and so their loss is not as impactful as it would be in a coarser mesh. This was shown by comparing the initial and final masses of both the steel plate and the projectile for three different mesh sizes. **Figure 4- 11** shows a visual representation of the mass loss due to element erosion and **Table 4- 3** lists the total masses and the total mass lost in each mesh. The loss of mass did decrease with increasing mesh size. It was also much more significant for the projectile, as a large amount of the projectile's surface is involved in the impact while only a fraction of the plate experienced any element erosion. The entire mass of the plate was considered when it really should be only some affected zone.

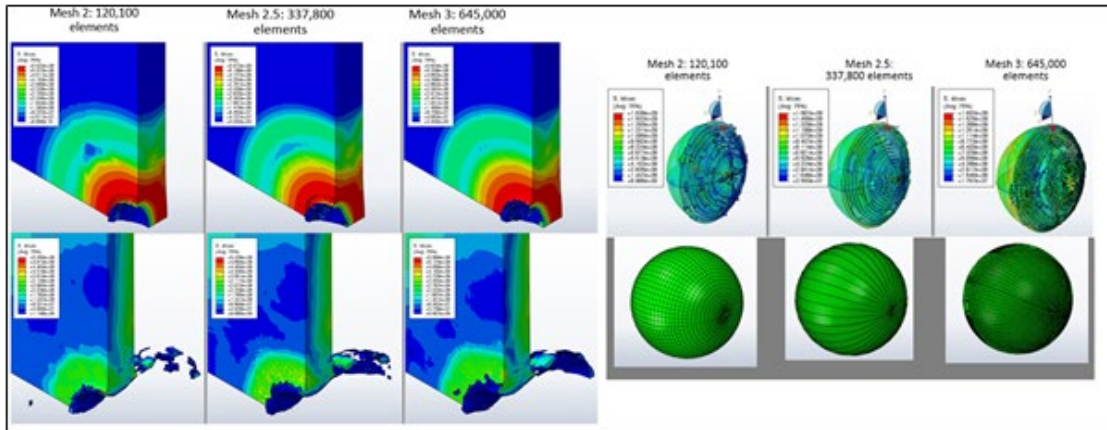


Figure 4- 11: Ballistic impact model of the steel plate and projectile. The amount of element erosion is compared between three different mesh sizes.

Table 4- 3: Mass loss of the steel plate and the projectile for each mesh

Projectile				Plate			
Mesh #	Initial Mass	Final Mass	Percent Difference	Mesh #	Initial Mass	Final Mass	Percent Difference
2	8.98760	5.67077	36.90451	2	0.28944	0.28814	0.44880
2.5	8.95328	6.30355	29.59508	2.5	0.28944	0.28828	0.39940
3	9.00412	7.60222	15.56954	3	0.28943	0.28844	0.34367

4.2 Embedded Element Modeling

4.2.1 Initial Tests

While the embedded element method in Abaqus should be improved before being used to model plates with high fiber volume ratios, a plate of Dyneema® was modeled in Abaqus using the embedded element method to visually compare results to the preliminary ballistic experiments discussed earlier. This also provided an opportunity to test the MATLAB embedding code in a true model. The MATLAB script was used to add embedded elements to a 10” diameter, ½” thick Dyneema® plate shown in Figure 4- 12. The added truss elements had diameters of $5.38 \times 10^{-4} m$ representing 10,000 fibers per truss. Only elastic properties were assigned to the fibers (embedded

truss elements) and the matrix (continuum elements) based on material properties used in a thesis paper from Delft University of Technology[4], also included in Table 4- 4. The bottom of the plate was constrained to represent a clamped boundary condition and a modeled tool steel 3/8” diameter steel ball impacted the plate at 1376 m/s. Since only elastic properties were modeled with no damage mechanisms specified, the sphere did not penetrate the plate, but the initial displacement response looks correct when visually compared with images from the ballistic test in Figure 4-13.

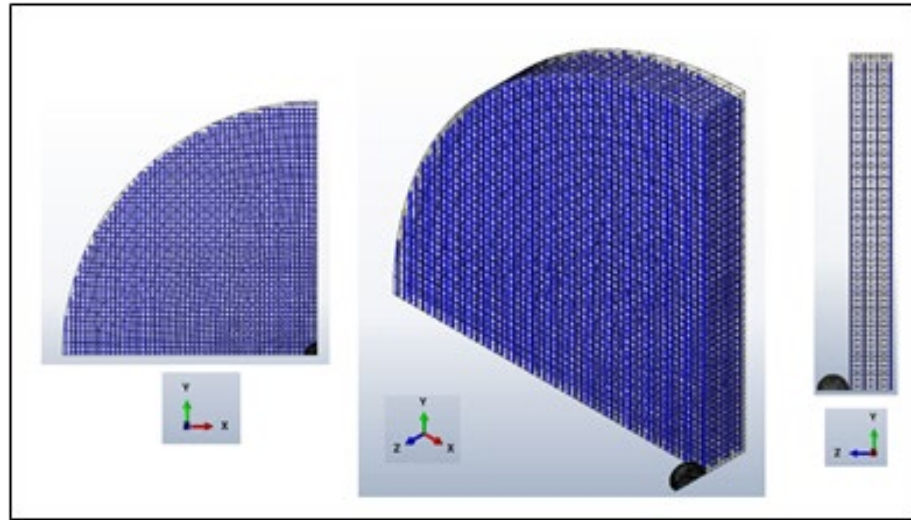


Figure 4- 12: Dyneema® embedded element mesh from three different views.

Table 4- 4: Material properties used for the embedded element model of Dyneema®[4]

	Fibers	Matrix
E	200 GPa	5.08 MPa
ν	0.3	0.4
ρ	981 kg/m ³	980 kg/m ³
Fibers per Truss (FPT)	10000	
Truss Element Diameter	2.269E-5 m ²	
# Elements	2840	8000

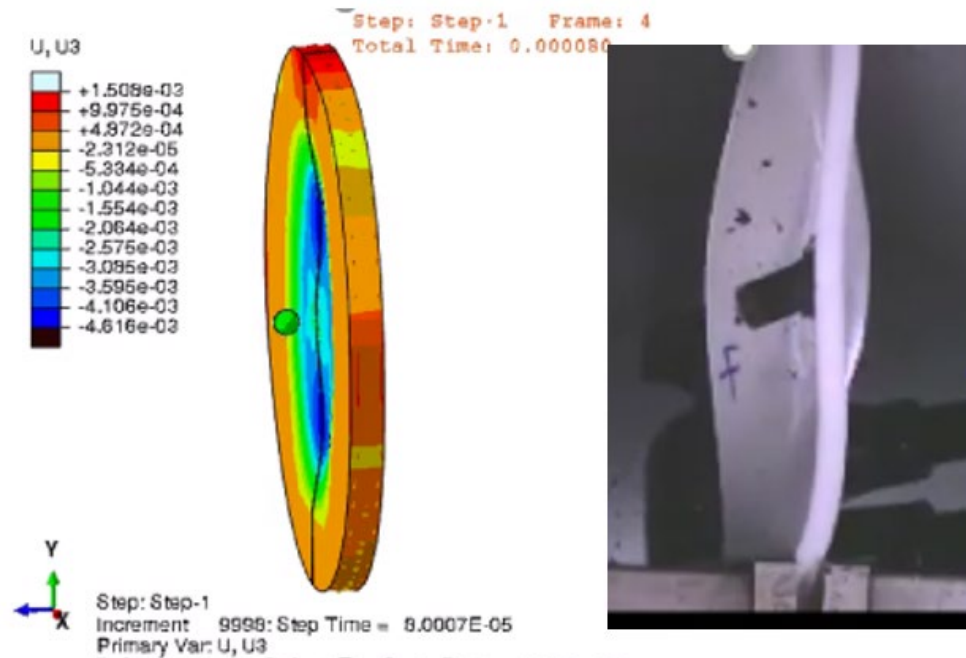


Figure 4- 13: Embedded element elastic model of Dyneema® shows similar deformation patterns as the ballistic tests.

One of the goals of using the embedded element method for modeling Dyneema is to naturally capture some of the deformation mechanisms that are intrinsic to having fibers in the $[0^\circ/90^\circ]$ orientation. The back face deformation cone, shown in Figure 4- 14, is where the fibers that bear the main load of the impact stretch out to the edges of the plate and create a sort of tent like deformation pattern. This initial model, as shown in Figure 4- 15a, produced some of the tenting effect, but not as pronounced as we would like. The plate was modified to be $\frac{1}{4}$ " thick rather than $\frac{1}{2}$ " to increase the amount of elastic back face deformation. Additionally, the matrix material, which was initially modeled as an orthotropic material based on Utomo et al. [4], was changed to be isotropic, since the embedded elements should be creating all the orthotropic response. These modifications helped to increase the visibility of the tenting effect and create a much more symmetrical deformation pattern as often seen in the literature. The results of the modifications are shown in Figure 4- 15**Error! Reference source not found.**b and c. At this point, we had no

numerical data to compare the simulation results with, but based on visual comparisons of images such as **Figure 4- 13**, **Figure 4- 14**, and **Figure 4- 15**, we concluded that the chosen material model was behaving in a reasonable manor.



Figure 4- 14: The back face deformation shape that we refer to as tenting.

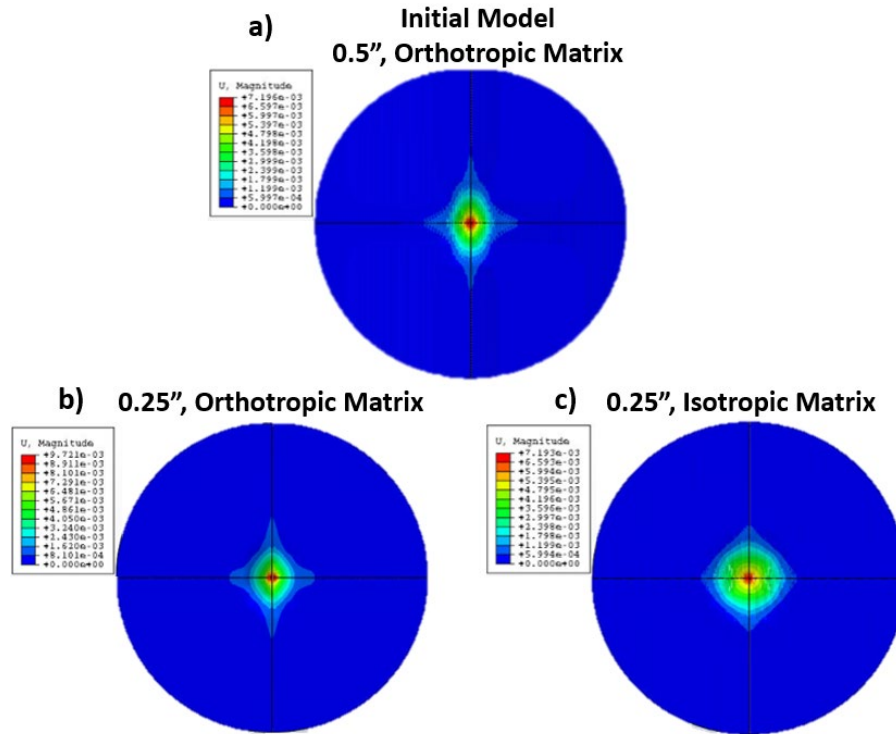


Figure 4- 15: Maximum back face deformation of plate impact test. a) the initial iteration of a 1/2" thick plate with an orthotropic matrix material. b) 1/4" thick plate with an orthotropic matrix material) 1/4" thick plate with an isotropic matrix material. This model gives the most realistic looking symmetrical tenting effect.

4.2.2 Karthikeyan Plate Impact

After getting a decent result from the initial models, we decided to attempt a comparison to experimental data available in the literature. Karthikeyan et al. [74] reports a test using shadow moiré images to measure back face deformation. It was not difficult to modify the embedded finite element model to match the specifications for this experiment and then numerically compare the amount of back face deformation. The parameters for Karthikeyan's experiment are shown in **Figure 4- 16**, along with the finite element mesh used. Two different embedded truss meshes were used to investigate the effect on truss mesh size on the back face deformation, one used 100 fibers per truss and the other 500 fibers per truss. The same material properties as the earlier plate impact

tests were used. **Figure 4- 17** shows the model back face deformation for two points in time. There is little difference in the finite element deformation when different numbers of fibers per truss are used, and both are close to what is observed in the experiment. The deformation in the shadow moiré images is determined by counting the fringes in the light interference pattern and multiplying by 1.58 mm for each fringe to get the displacement, so there is a fair amount of error in this measurement (as each fringe contains a deformation range of 1.58 mm so we do not know for sure the maximum deformation). Also given that the material properties used in the finite element model were not calibrated or chosen exactly, this is a very good result for a proof-of-concept model.

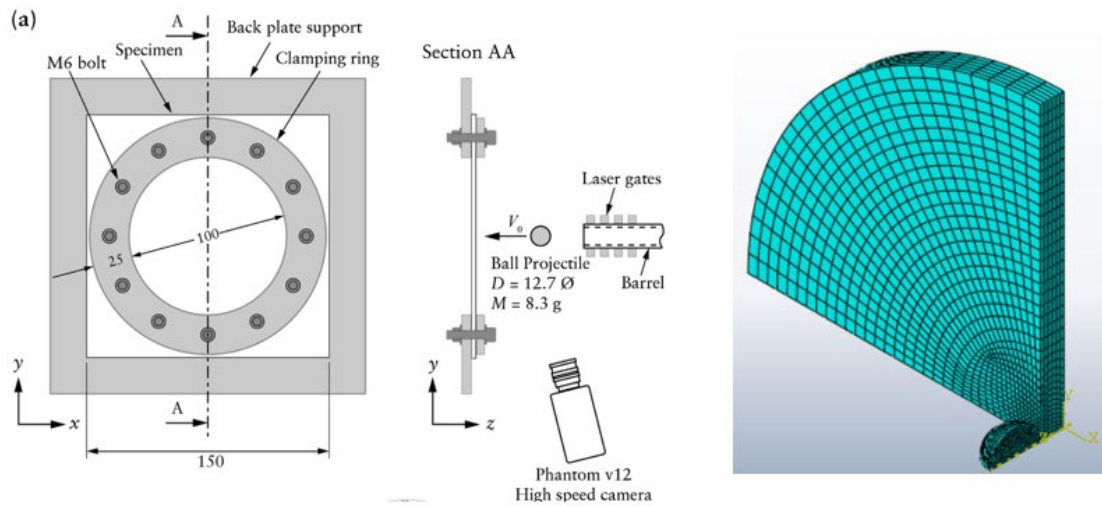


Figure 4- 16: Set up used in Karthikeyan's impact experiment^[74] and the finite element mesh used to create the embedded element model.

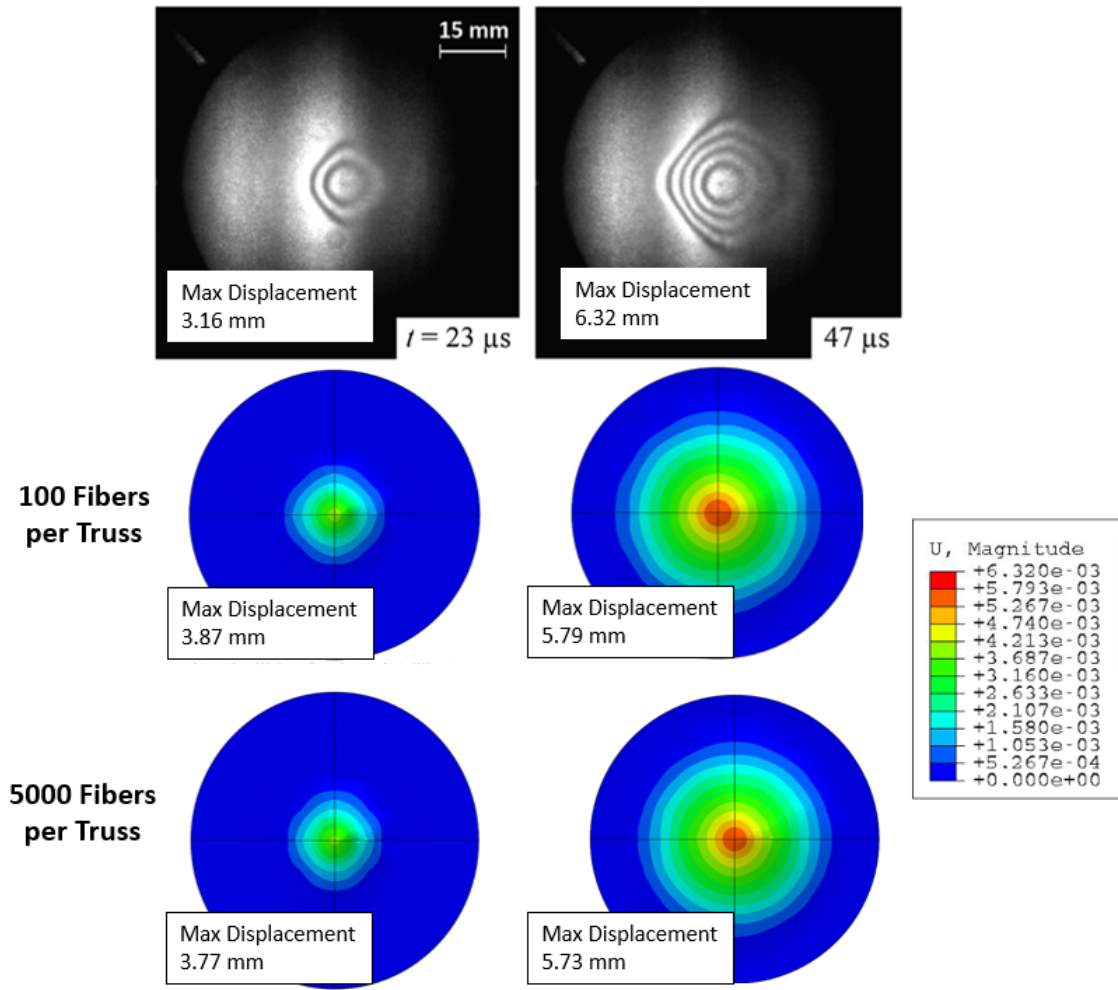


Figure 4- 17: Results of the embedded element models of Karthikeyan's impact experiment[74]. Both the 100 and the 500 fibers per truss cases produce good results.

4.3 Using the Embedded Element Method to Simulate Full Impact

After the success of the back face deformation modeling of the plate from the Karthikeyan experiment, we attempted a model that included damage and failure using our own experimental data. Here, we present a proof-of-concept model of a plate of Dyneema® under impact conditions using the embedded element method to represent the cross-ply fibers grouped into truss elements. We want to show that the embedded truss elements provide an easy way to implement the

orthotropic material properties and capture specific failure mechanisms observed in experimental data.

4.3.1 Methods

Plate impact experiments

A series of experimental tests were conducted to characterize the response of HB26 composite panels, a subset of which is used here as validation data for the model. The test being used for comparison had a single HB26 circular panel measuring one inch thick by eight inches in diameter. The plate was held in the fixture shown in **Figure 4- 18**. For this test, a single plate was

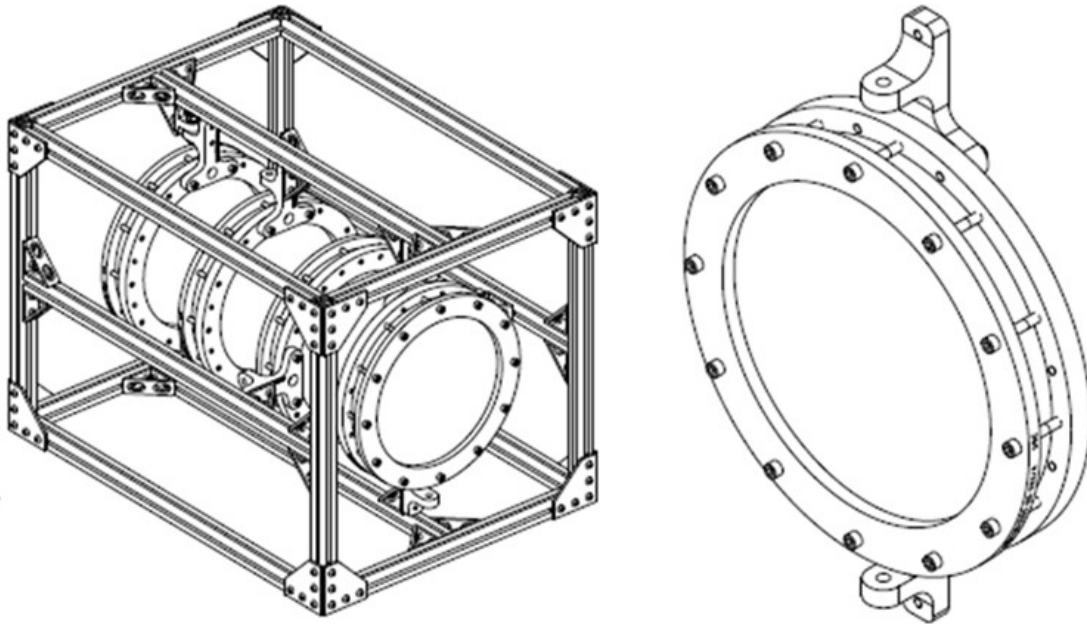


Figure 4- 18: Experimental test fixture (left) and a close up of the clamping rings which fit into the fixture (right).

clamped in rings with the bolts torqued down to 16.27 Nm (12 ft-lbs), which was selected to replicate a roughly fixed boundary condition in the clamped region, which not permanently deforming the plate due to the clamping pressure. Once clamped inside the support rings, the

exposed diameter of the plate is 171.45 ± 0.762 mm (6.750 ± 0.03 in). For this test, a 9.525 mm ($3/8$ in) diameter hardened tool steel ball was fired from a .50 caliber barrel with the use of a plastic sabot, which was captured/deflected using a series of stripper plates to minimize the potential for sabot interaction with the target panel. High speed cameras and a chronograph were used to independently measure the velocity of the projectile, which was 1028 m/s. With this impact velocity and impact occurring at the center of the panel, the projectile was successfully captured in the target.

CT scans and results

After the testing was completed, the plate was scanned in a micro-CT scanner with an image resolution of 55 micrometers. While this is not enough to capture the roughly 17-19 μ m diameter fibers, it is enough resolution to see plies, cracks, and other meso-scale features while maintaining a large field of view of approximately four inches by four inches and through the entire deformed thickness. A view centered on the main penetration channel is shown in **Figure 4- 19**.

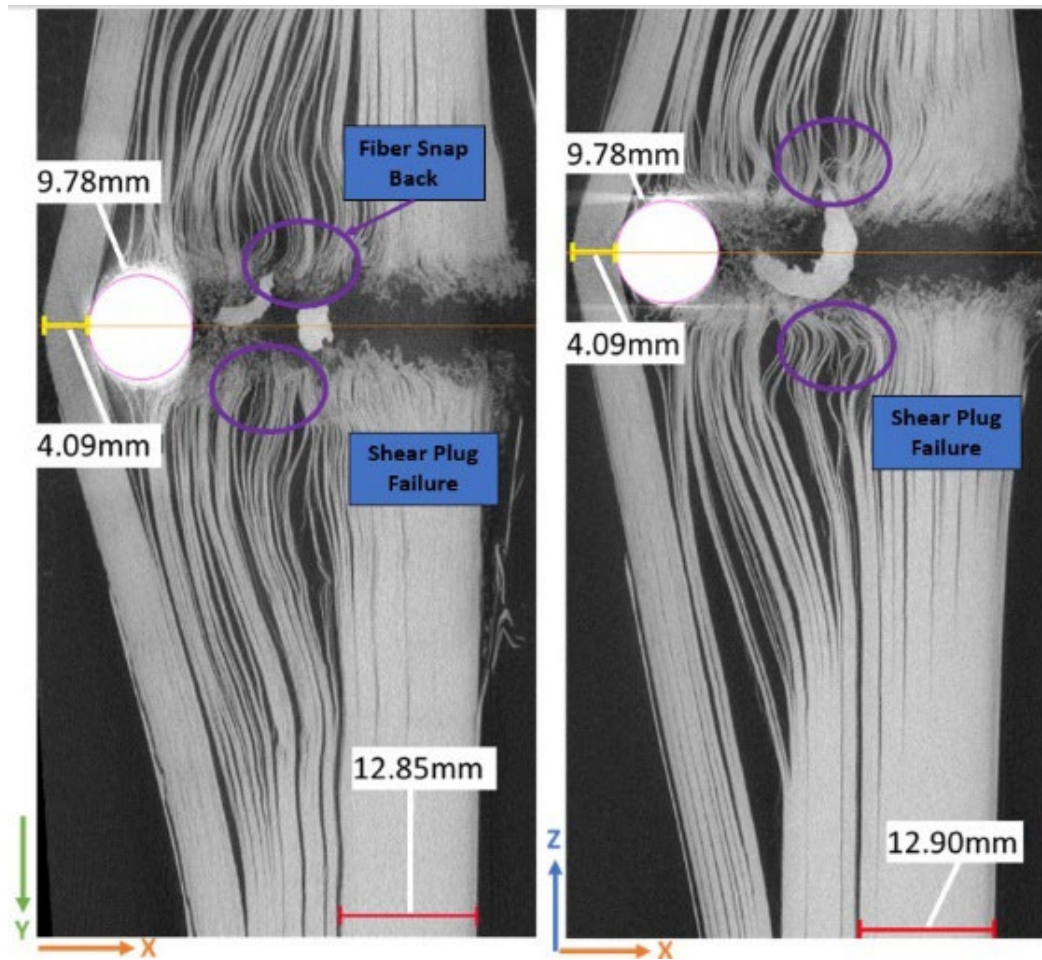


Figure 4- 19: X Y view (left) and X Z view (right) of the 8-inch diameter by 1 inch thick HB26 plate when impacted by a 3/8 inch diameter steel sphere traveling at 1028 m/s at impact.

Of note is the thickness of consolidated or non-delaminated composite measuring 12.85-12.90 mm representing the thickness of the shear plug or fiber cutting zone, as well as the 4.09 mm thickness of unpunctured plate remaining ahead of the projectile. Additionally, we see significant fiber/ply curling at the transition between the shear plug dominated zone and into the delamination dominated zone. This curling is due to non-trivial deformation of fibers that have debonded from the matrix when breaking, as opposed to the little to no deformation in the shear-plug failure while the projectile velocity is still high at the beginning of the penetration event. This axial deformation or tension in the fibers causes them to retract into the plate when the fibers are broken and the

energy released, creating what we call a fiber snap-back zone highlighted in **Figure 4- 19** at the end of the shear plug failure area of the target.

Finite Element Modeling

To model delamination as well as fiber and matrix failure, the embedded element method is combined with a layered mesh with cohesive contact between the layers. One quarter of the 1 inch thick, 8 inch diameter plate was modeled with 20 1.27mm thick layers as shown in **Figure 4- 20a** and **Error! Reference source not found.b**. Each layer is meshed with 8880 hexahedral elements for a total of 177,600 matrix mesh elements. The mesh is seeded at 0.5mm in the 25mm radius near the impact zone and is allowed to be a courser seeding of 5mm closer to the clamped edges. Each layer has 3 through thickness elements. The cohesive contact properties in **Table 4- 5** are taken from Hazzard et al. [11] Boundary conditions on the model in **Figure 4- 20b** are symmetry conditions on the two cut sides and the clamp boundary condition from the experiment is modeled by fixing all of the displacements of the front and back surfaces of the plate that would be under the clamp. The 3/8 inch spherical steel projectile is modeled as a rigid body since no deformation was observed in the impact experiment. The full mesh of the host elements is shown in **Figure 4- 21**.

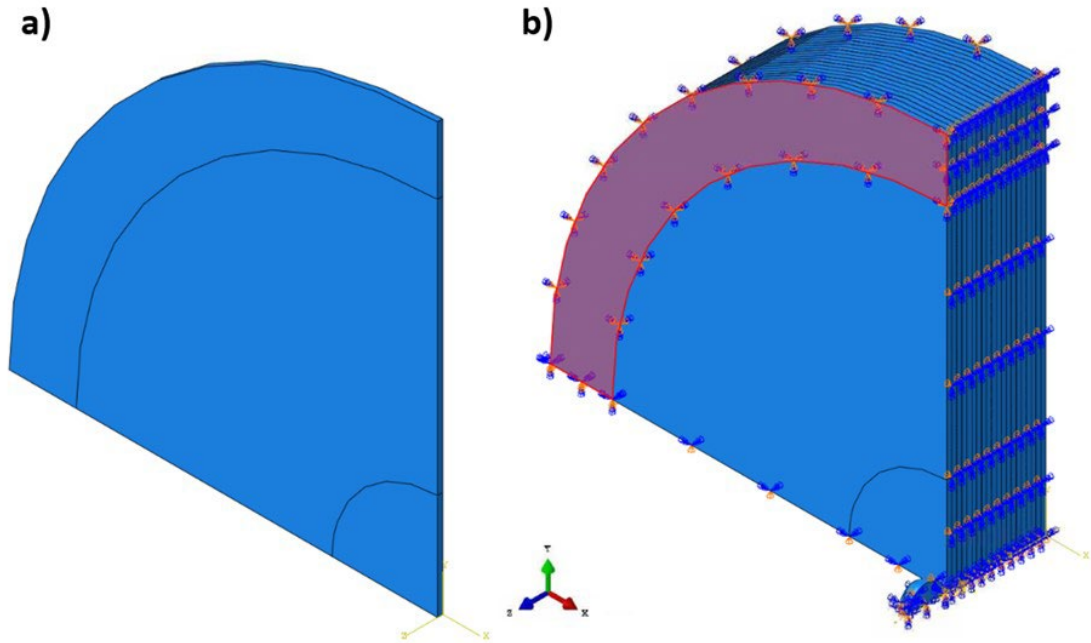


Figure 4- 20: a) A single layer of matrix material b) quarter symmetry finite element model with 20 matrix layers connected by cohesive contact constraints and the clamp boundary conditions (red) and symmetry boundary conditions imposed on the model.

Table 4- 5: Cohesive contact properties for the matrix layers.

	Mode I Strength	Mode I Stiffness	Mode I Fracture Energy	Mode II Strength	Mode II Stiffness	Mode II Fracture Energy
Hazzard 2018[11]	1.2 MPa	60 N/mm ³	0.544 N/mm	1.8 MPa	36 N/mm ³	1.088 N/mm

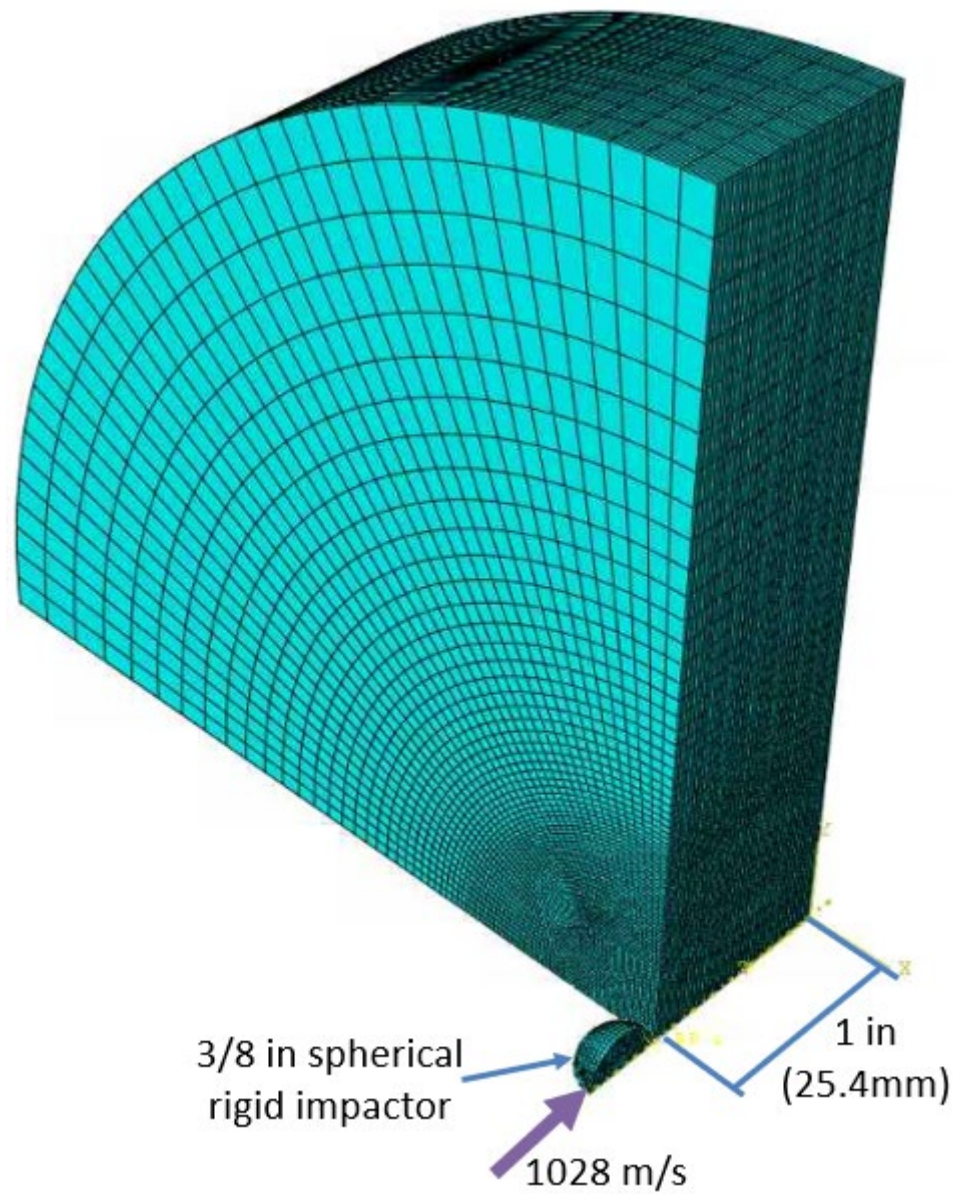


Figure 4- 21: Host element matrix mesh of 177,600 hexahedral elements.

The embedded fibers were added using a custom MATLAB script. The MATLAB script takes an Abaqus input file and reads in the node connectivity of the part to host the embedded elements (**Figure 4- 22a**). The endpoints of the truss elements are determined by creating a point grid around the part and then determining which points lie within the part (**Figure 4- 22Error! Reference source not found.b**). The points on the edges of these bounds are specified as the

endpoints (**Figure 4- 22Error! Reference source not found.c**) and the lines between the endpoints define the truss elements (**Figure 4- 22Error! Reference source not found.d**). Using this method allows the script to add truss elements to complex geometries that may include concave surfaces or internal holes. Finally, the MATLAB script writes a new input file, complete with Abaqus's embedded element constraint that can be run by Abaqus without any further edits by the user. The number of truss elements added is determined by how many can be packed together in the part based on their cross-sectional area. The area is determined by the user by specifying the number of fibers they want each truss to represent. The area is then calculated by Equation 2-1, repeated here, where n is the number of fibers assigned to a truss and D_{fiber} is the fiber diameter. We assume a fiber diameter of 17 microns (which is common for Dyneema® fibers). A larger number of fibers per truss yields less but larger truss elements for a courser mesh, while a smaller number creates a finer mesh. For this model, we tested two different fiber meshes. The first used 1380 fibers per truss (FpT) which created truss elements with a 0.632mm diameter and 405,000 total truss elements in the model. The second used 5581 FpT which created truss elements with a 1.3mm diameter and 100,740 total truss elements. Both create a fine mesh of truss elements, as shown in **Figure 4- 23Error! Reference source not found..** These two FpT numbers were chosen because the truss diameters correspond to having two layers of 0°/90° trusses (for the 1380 FpT) or one layer of truss elements (for 5580 FpT) per 1.27mm thick matrix layers.

$$A_{truss} = n \frac{\pi}{4} D_{fiber}^2 \quad \text{Eq. 2-1}$$

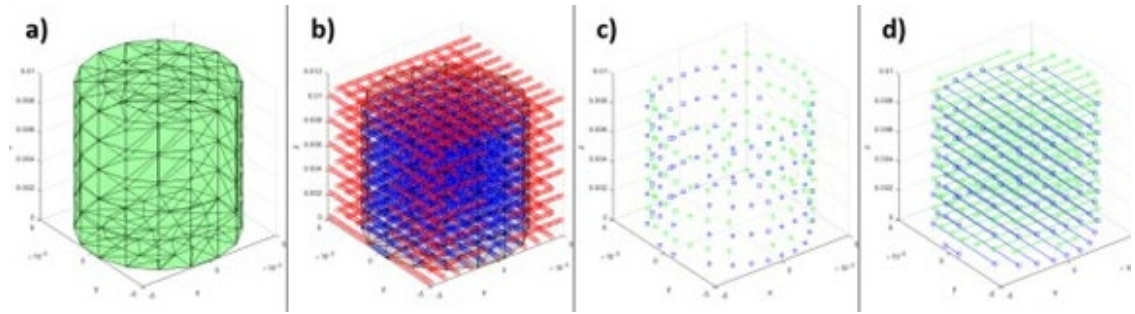


Figure 4- 22: Progression of creating embedded truss elements in a cylinder-shaped part.

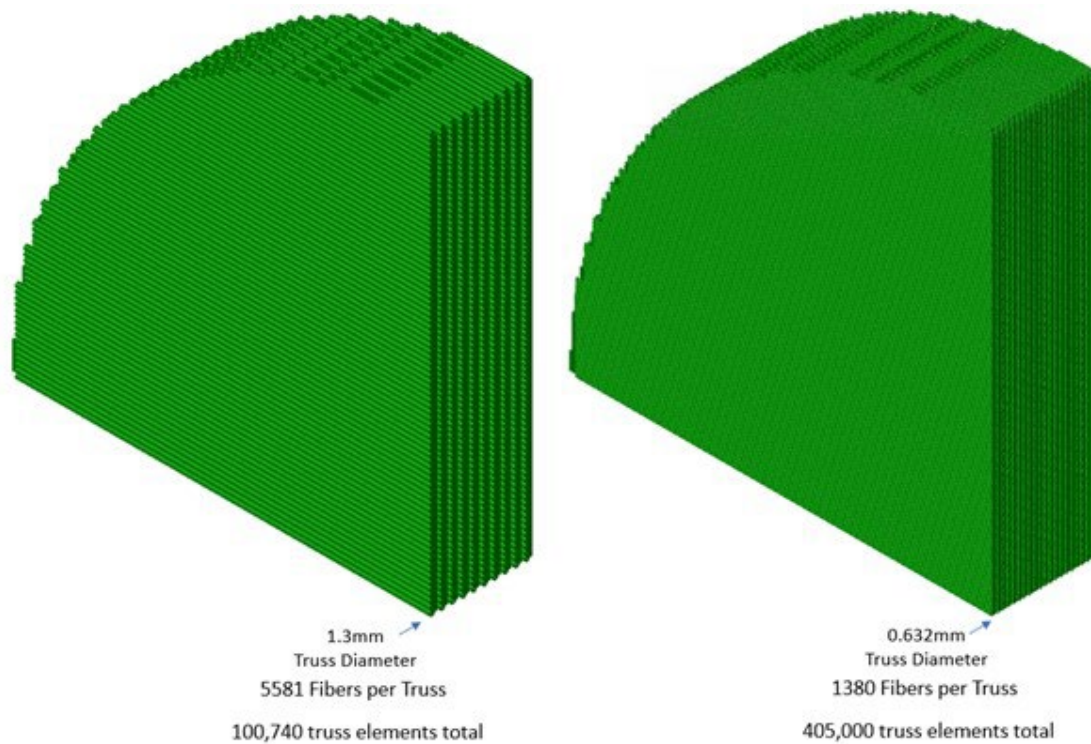


Figure 4- 23: Truss element meshes for the 5581 FpT and 1380 FpT meshes.

Material properties for the matrix and fiber materials were chosen from literature sources and are listed in **Table 4- 6**. Both components were modeled as elastic brittle, which is how the overall material as well as individually tested fibers are assumed to behave in high rate loading conditions.[4], [11], [44], [67] Material properties for just the matrix material are scarce as the overall response of the Dyneema® is assumed to be dominated by the fiber response[75], [76].

There are several studies on the high strain rate response of fibers that suggest they are rate dependent[67] so a rate dependency was added to the damage behavior of the fibers.

Table 4- 6: Material properties for isotropic matrix material and fibers.

Matrix Material Properties		Ref	Truss Material Properties		Ref
E	700 MPa	[50]	E	135 GPa	[50]
ν	0.45	[50]	ν	0.45	
ρ	980 kg/m ³	[50]	ρ	981 kg/m ³	[50]
σ_{yield}	70 MPa	[50]	$\sigma_{yield}, \dot{\epsilon}$	3.5 GPa, 0.01 1/s 4.3 GPa, 1.0 1/s 4.5 GPa, 1000 1/s	[67]
Fracture strain, $\dot{\epsilon}$	0.1, 0.01 1/s	[50]	Fracture strain, $\dot{\epsilon}$	0.0391, 0.01 1/s 0.0251, 1.0 1/s 0.01167, 1000 1/s	[67]
Fracture Energy	1460 J/m ²	[70]	Fracture Energy	1 J/m ²	

Modeling Challenges

We have not accounted for volume redundancy in this model, which is inherent to the embedded element method[77]. The embedded element method overlays two meshes and since it's only mathematical the volumes of both meshes exist in the same space. Commercial finite element

solvers do not remove any of the host element volume to account for the space occupied by embedded elements and instead leave it to the user to decide what to do about the redundancy.[78] Generally the redundancy is dealt with by reducing the stiffness and density of one or both of the mesh materials[33], [34], [36], [39], [41]. This has worked well for models with only small amounts of inclusions and quasi static deformation but could be problematic for modeling a high fiber volume fraction material like Dyneema® in a dynamic environment where kinetic energy is non-trivial and the material sound speed matters. Since the goal of this model is to provide a proof-of-concept model to show if the embedded element method can capture some of the specific deformation mechanisms of Dyneema®, we are not addressing the volume redundancy at this time. We have other work on a way to remove the redundancy from the finite element calculations on an algorithmic level, but that work is not advanced enough to model an impact event[77].

4.3.2 Results

Both embedded element models were run in Abaqus Explicit with parallel processing on four CPU cores. The 1380 FpT model took about 30 minutes to complete while the 5581 FpT model finished in under 15 minutes. Additionally, we tried running the 1380 FpT model using beam elements instead of the truss elements. Shear plug failure is the main failure mechanism for the first part of the impact, and we were interested in differences between using truss and beam elements since beam elements have shear stiffness and can fail in a shear mode where trusses only support tension and compression. We found that although failure initiated sooner in the beam elements due to the addition of the shear component, the beam model took significantly longer to run than the truss model. For the same number of elements, the beam model took 12 hours to run. The results between the two models in **Figure 4- 24Error! Reference source not found.** look similar, so for the rest of the analysis we chose to use the truss model.

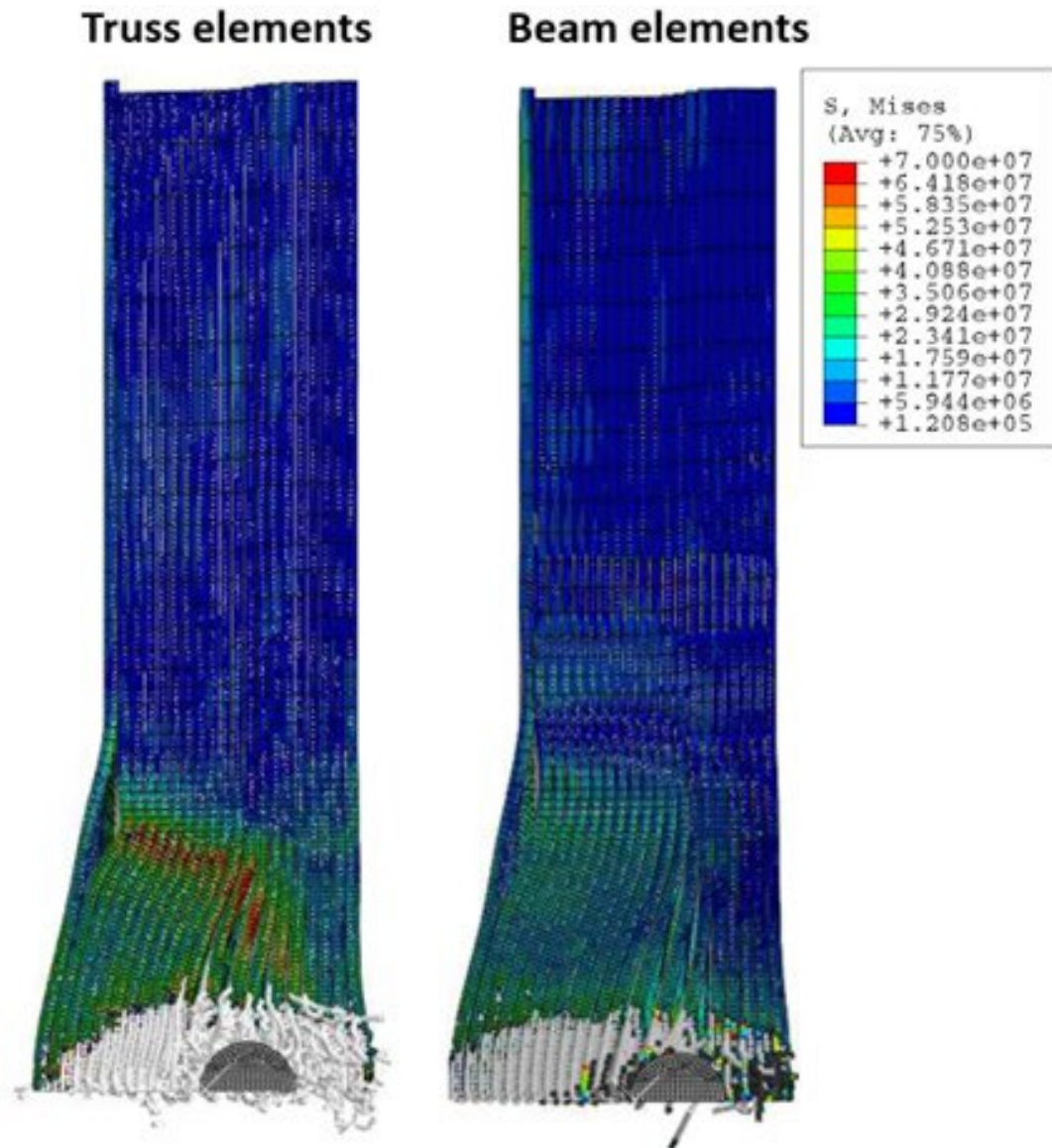


Figure 4- 24: Running the embedded element model with truss elements and with beam elements produced visually similar results, with the beam elements showing more early failure and less matrix delamination than the trusses. The beam element model takes significantly longer to run.

With this embedded element model, we were looking to see if we could capture the deformation cone, shear plug failure, fiber snap back, and indication of the indirect tension mechanism. Comparison between the 1380 FpT model and the CT imaging of the plate cross section in **Figure 4- 25** shows that, although the model projectile did not penetrate the plate as far as in the experiment, we were able to replicate some of the failure features. The initial failure zone

appears to be a shear plug failure where the material plies hold together and the projectile creates a gap just wide enough for itself. The shear plug zone is also a similar size to experimental data, 10mm depth versus the experimental depth of 12.85mm. After the shear plug failure, ply delamination becomes more dominant. With the delamination, fibers are stretched further before they fail, causing them to elastically snap back, which can be seen in both the CT imaging and in the finite element model. A direct comparison could not be made between the back face deformation between the model and experiment because the penetration depth was not great enough in the model. To accommodate for this, the Von Mises stress profiles for the front and back of the plate, as demonstrated in **Figure 4- 26 Error! Reference source not found.** are used to draw a comparison between the experiment and model. The characteristic deformation cone on the back face of the plate during impact is due to the stretching of the main 0 and 90° fibers across the plate. The embedded finite element model shows this mechanism in the Von Mises stress profile on both the front and back of the plate.

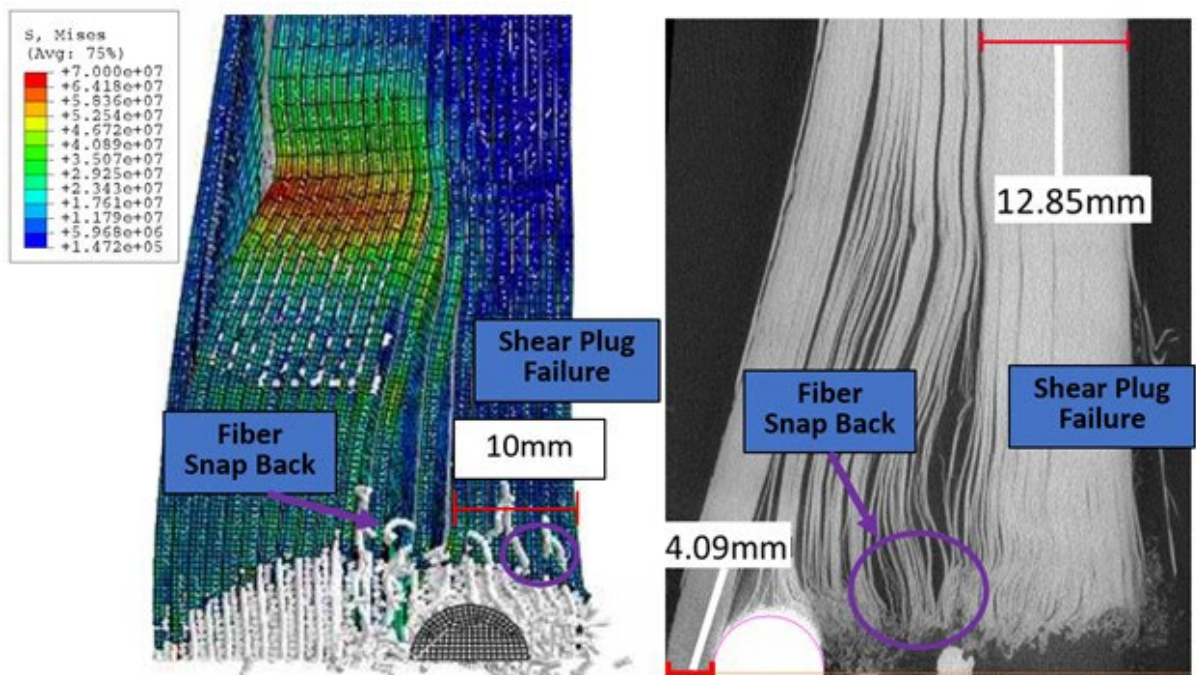


Figure 4- 25: When compared to CT images of the impact experiment, the embedded element model shows a similar shear plug failure zone and the fiber snap back although the projectile penetration distance is shorter.

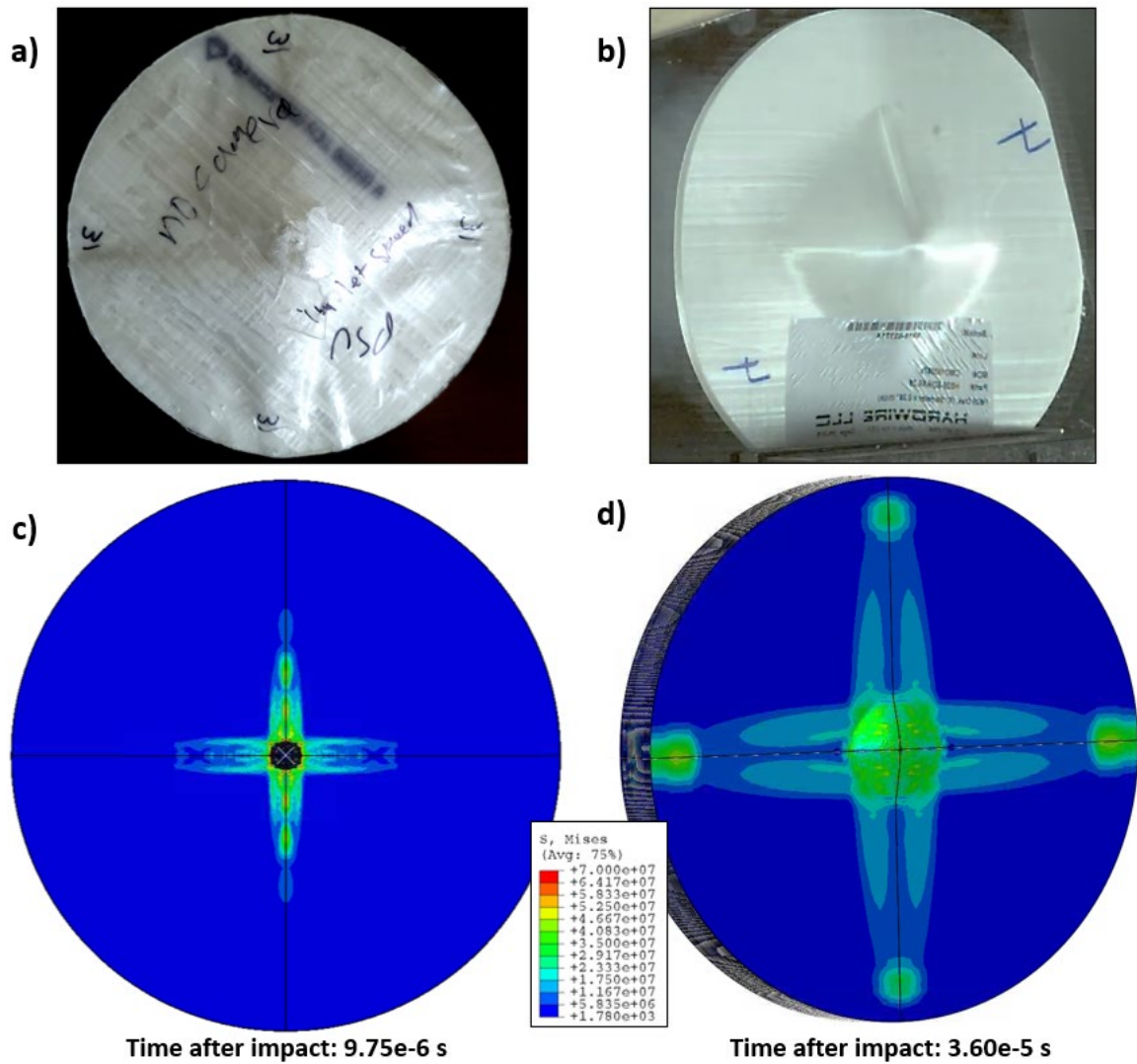


Figure 4- 26: a&b) Examples of the deformation cone created by the stretching of the principal fibers from other impact experiments. c) Front and (d) back face stress patterns on the fully mirrored plate reveal the characteristic deformation cone associated with Dyneema® and other 0°/90° composites.

The indirect tension mechanism is described as the mechanism where the transverse compressive loading of the projectile generates tension in the plies due to anisotropic expansion of the alternating 0° and 90° plies[12], [13]. **Figure 4- 27** from Liu et al. shows an illustration of the mechanism[13]. Tension in the 0° layer is transferred across the 90° layer by shear in the matrix

interface between the two layers. This causes compression in the 90° layer and then the same matrix shear transfers this load to a tension stress in the next 0° layer.

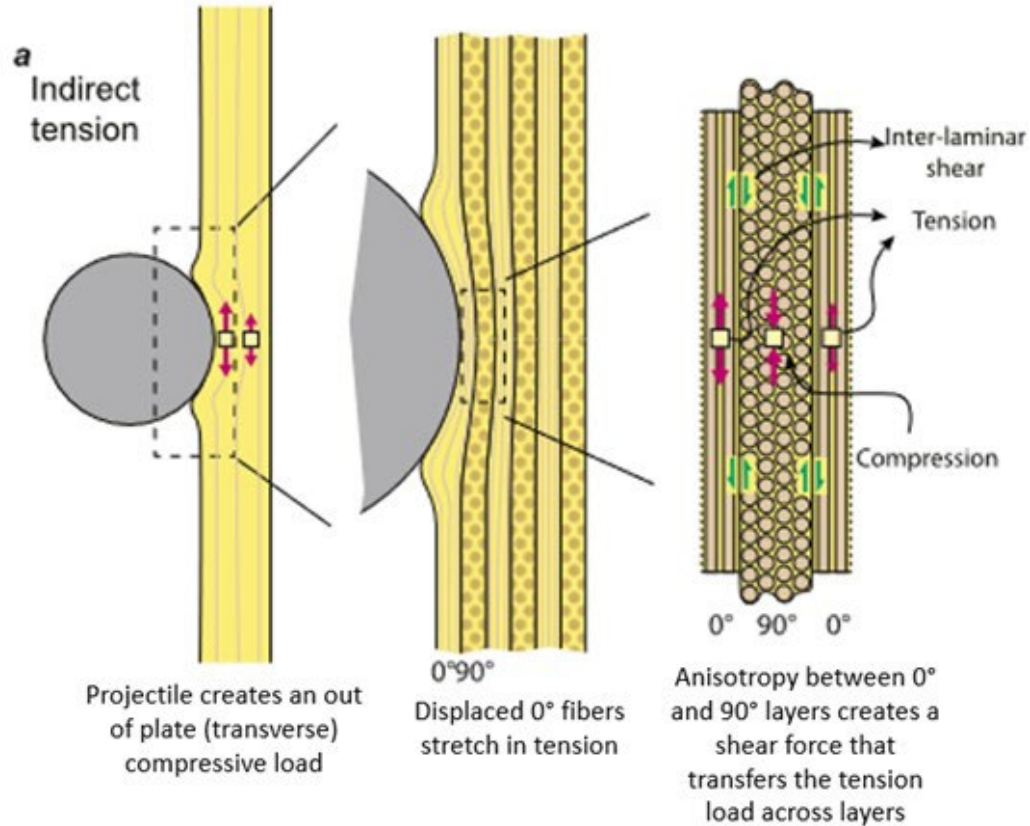


Figure 4- 27: Sketch of the indirect tension mechanism from Liu et al. [13] with added description of the mechanisms involved.

We investigated the degree to which our embedded element model reflected this indirect tension by looking at the tensile/compression stress in the fiber layers and the shear stress in the matrix in the first microseconds of the impact, displayed in **Figure 4- 28**. Both the 1380 FpT and 5581 FpT models showed the first 0° layer in tension, and there is a shear coupling between the first two layers of matrix material. There is significantly more shear in the 5581 FpT case, since this is the case where each matrix layer contains only a 0° or 90° layer of truss elements, while the 1380 FpT model has one of each for each matrix layer so there is less anisotropy between layers. The compression of the second fiber layer (the 90° layer) does appear somewhat, but it's unclear if

this is simply from the compressive forces of the projectile or caused by the indirect tension caused by the interlaminar shear. It can be argued that this shear driven compression will not be captured by this model since it is dependent on the change in cross sectional area/shape of the fibers as they compress together in a group and get closer together and this model only includes one row of truss elements for each 0° and 90° layer instead of multiple as in the **Error! Reference source not found.** diagram. Additionally, Abaqus does not allow for change in cross sectional area of truss elements and assumes they are incompressible, which will decrease the ability of the truss element layers to reproduce the compression in the 90° layer[79].

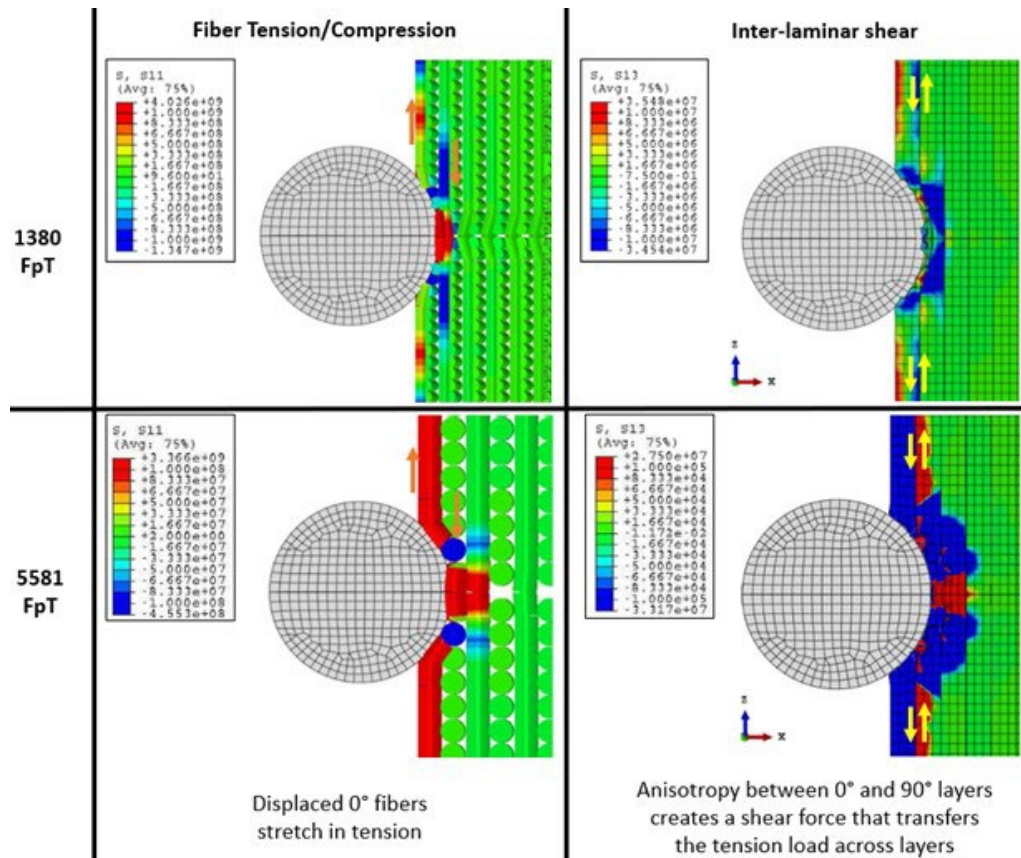


Figure 4- 28: Tensile/compressive stresses in the truss elements and shear stress between the matrix layers shows characteristics of the indirect tension mechanism which facilitates the tensile failure of the truss elements early in the impact process.

4.3.3 Discussion

Failure mechanisms

We set out to see if modeling Dyneema® using embedded truss elements would capture some specific deformation and failure mechanisms in an impact experiment. We used cohesive contact constraints between layers of continuum element matrix material which was able to capture the delamination of Dyneema® almost exactly how it is observed in the CT images of experiments. The characteristic deformation cone that is formed from the stretching of the principle $0^\circ/90^\circ$ fibers during impact can also be captured by the truss elements embedded in the matrix layers. Since this is mainly seen on the back face deformation, which the experiment used here did not experience much of, we looked at the stress profiles in the plate and were able to identify where the truss elements representing the principal fibers took most of the tensile stress. Related to this, when those principal fibers stretch and then fail, they have some elastic recovery and snap back in a distinct pattern near the beginning of the delamination regime in the penetrator's pathway. Modeling the fibers explicitly as truss elements was able to capture this feature very well. Finally, we investigated if the proposed indirect tension mechanism would appear in the model. Our model shows some features of the indirect tension mechanism, notably the shear coupling between the matrix layers. Since the truss elements only support longitudinal tensile and compression loading, they can only fail via those loadings. The fact that the embedded element model showed an initial shear plug failure zone like the experiment means that the fibers were loaded in tension by something like the indirect tension mechanism. More work needs to be done with the arrangement of the truss elements and potentially tests using finer truss and matrix meshing to see what the size effects of the mesh are on this mechanism.

Volume Redundancy

An important feature of the embedded element method that was not addressed was the volume redundancy that occurs when two meshes are superimposed on one another. This adds additional mass and stiffness to the model which impacts how the whole model behaves. Generally, redundancy is dealt with by reducing the stiffness and density of one or both mesh materials[33], [34], [36], [39], [41]. This has worked well for models with only small amounts of inclusions and quasi static deformation but could be problematic for modeling a high fiber volume fraction material like Dyneema® in a dynamic environment where kinetic energy is non-trivial, and the material sound speed matters. We did not want to compromise those mechanics in this model so we did not modify stiffness or density. However, we think the extra mass and stiffness is one of the main reasons the model predicted far less penetration distance than the experiment did; there is more mass to push out of the way and the overall material is stiffer so harder to break. Future work will include comparing ways to address the redundancy and hopefully show that the redundancy needs to be dealt with on a solver level rather than just adjusting the material properties.

Future work

In addition to addressing the volume redundancy problem, there are a few other things that could be done to improve this model. The matrix material stiffness and failure properties could be calibrated better. There is very little research on the behavior of the matrix material since failure is assumed to be dominated by the fibers[7], [80]. The properties used in this model have been inferred from other papers that model the entire composite by simply assuming that the matrix stiffness and failure strength is the same as the total composite in the through thickness direction. This is likely not the case and it can be seen in our embedded element model that the matrix material fails far

ahead of the actual location of the projectile. Further efforts in using this type of modeling will need more information about the matrix stiffness and failure properties. Another component of the model that can be improved is the plate boundary conditions. The clamp boundary condition was applied as a fixed displacement to just the outside surfaces of the plate. This does not accurately capture the clamp boundary that is used in the experiment, since the added pressure could affect stress wave propagation. Additionally, the clamp would allow the edges of the plate to slide if the force was large enough to overcome the friction.

There are some studies on fiber material properties that show that the fibers have a different transverse stiffness and strengths than their longitudinal direction[69], [72], [81]. This could impact how the initial shear failure of the impact works since the reduced through thickness strength could mean it's easier to fail the fibers in that direction. Truss elements could not capture this type of anisotropy since they only support longitudinal loads. Using a beam type element would allow for a different through thickness stiffness, but as shown earlier in this work, the beam type elements are far less computationally efficient than truss elements, but if such a failure mechanism is shown to be important to the impact failure they could be implemented.

4.4 Curved Plate

A benefit of the embedded element method is that the truss elements can be embedded in any orientation, so this method could be used to model curved plates of UHMWPE for body conforming armor or helmets. There is some evidence that fiber curvature changes the behavior of the material[18]–[20], [82]. Embedded elements could also be used in modeling the formation process, which is of interest for tracking areas of increased strain or material consolidation[1]. As a proof of concept, we created a curved plate impact model using the embedded element method to

show that the method is acceptable for curved plates and to investigate some of the differences between the curved and flat plate behavior.

A curved geometry was created in Abaqus CAE by extruding the arc shown in Figure 4- 29 by 250 mm. This created a plate with one dimensional curvature in the x direction that is compatible with the MATLAB Script for Embedded Fiber Creation discussed in Chapter 2. This code was used to add embedded truss elements with 1380 fibers per truss element resulting in a truss element diameter of 0.635 mm. The curved plate itself was divided into 5 0.635 mm layers with cohesive contact defined between the layers with the properties defined in Table 4- 5. The mesh of the truss elements, host elements of the curved plate, and a close-up comparison of the two meshes are shown in Figure 4- 30. The same 9.525 mm (3/8 in) diameter hardened tool steel ball mesh from Section 4.3 was used as the projectile in this model with an initial velocity of 600 m/s. The plate and truss element material properties also used the properties listed in Table 4- 6.

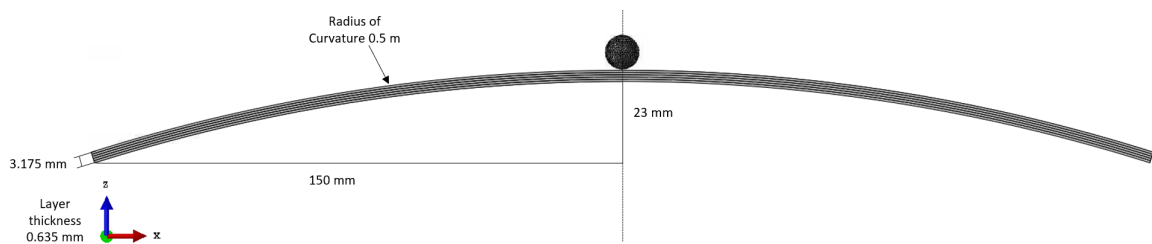


Figure 4- 29: Geometric definitions of the extruded cross section of the curved plate.

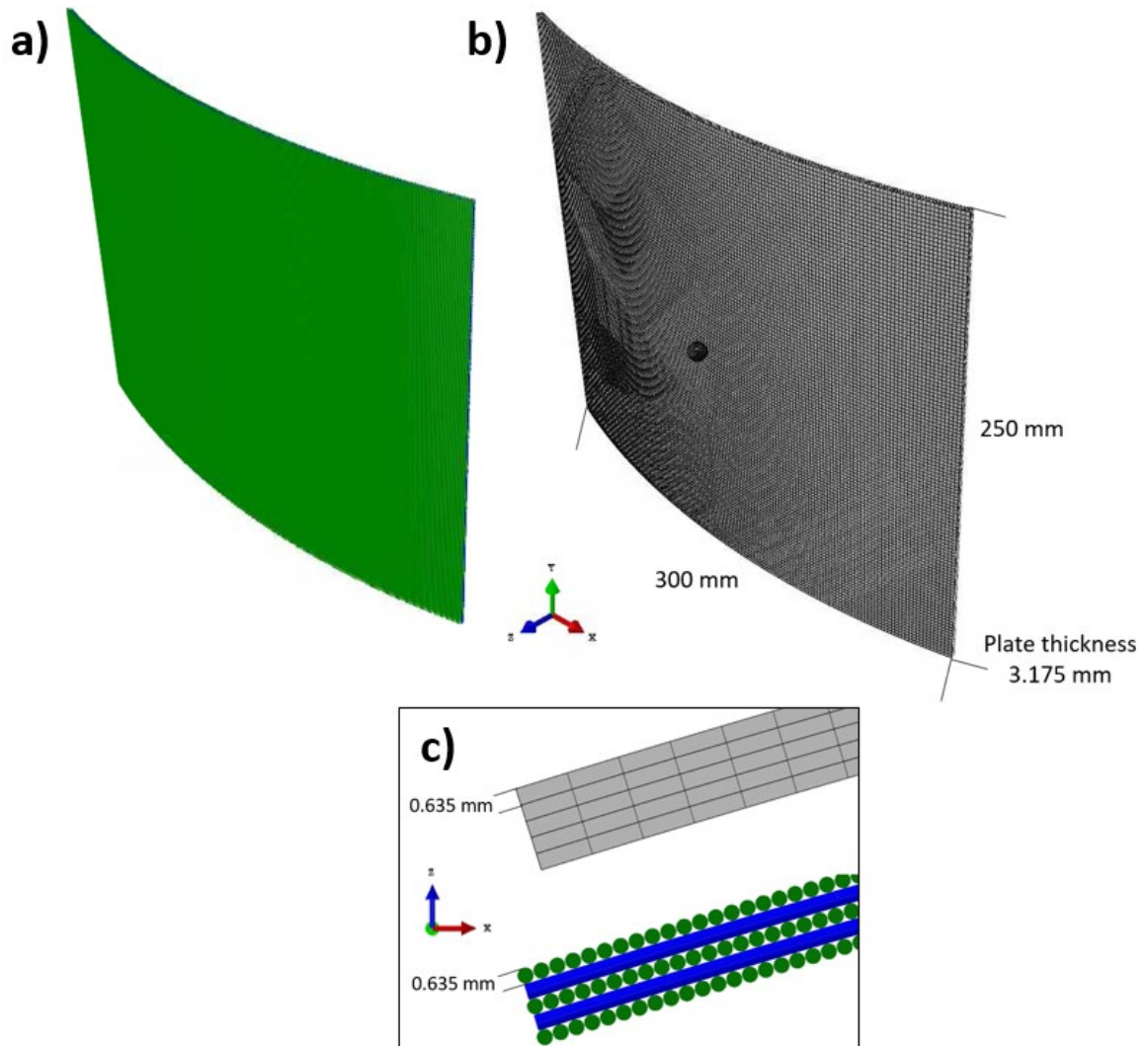


Figure 4- 30: a) Fiber mesh embedded in the curved plate model b) hexahedral mesh of the curved plate matrix material c) close up comparison of the the mesh density of the host mesh and the embedded truss mesh.

Von Mises stress results for the curved plate impact model are shown in Figure 4- 31. At 600 m/s, the thin plate was easily penetrated with a residual velocity of 350 m/s. The very distinctive deformation cone from the $0^\circ/90^\circ$ fibers is clearly visible. In the view of the back face stress in Figure 4- 32, it's clear that the fibers in the direction of curvature (fibers oriented along the x direction) take far less of the load than the fibers in the y direction. Measuring the speed of the stress wave in both directions, the y direction wave speed was significantly faster at 7500 m/s while the x direction wave speed was only 6000 m/s. This is because the stress wave is also

changing direction and not purely moving in the x direction, so it takes longer to cover an equivalent distance to the y direction stress. This shows that there are differences in the behavior of flat and curved plates that need to be taken into account when designing and modeling them.

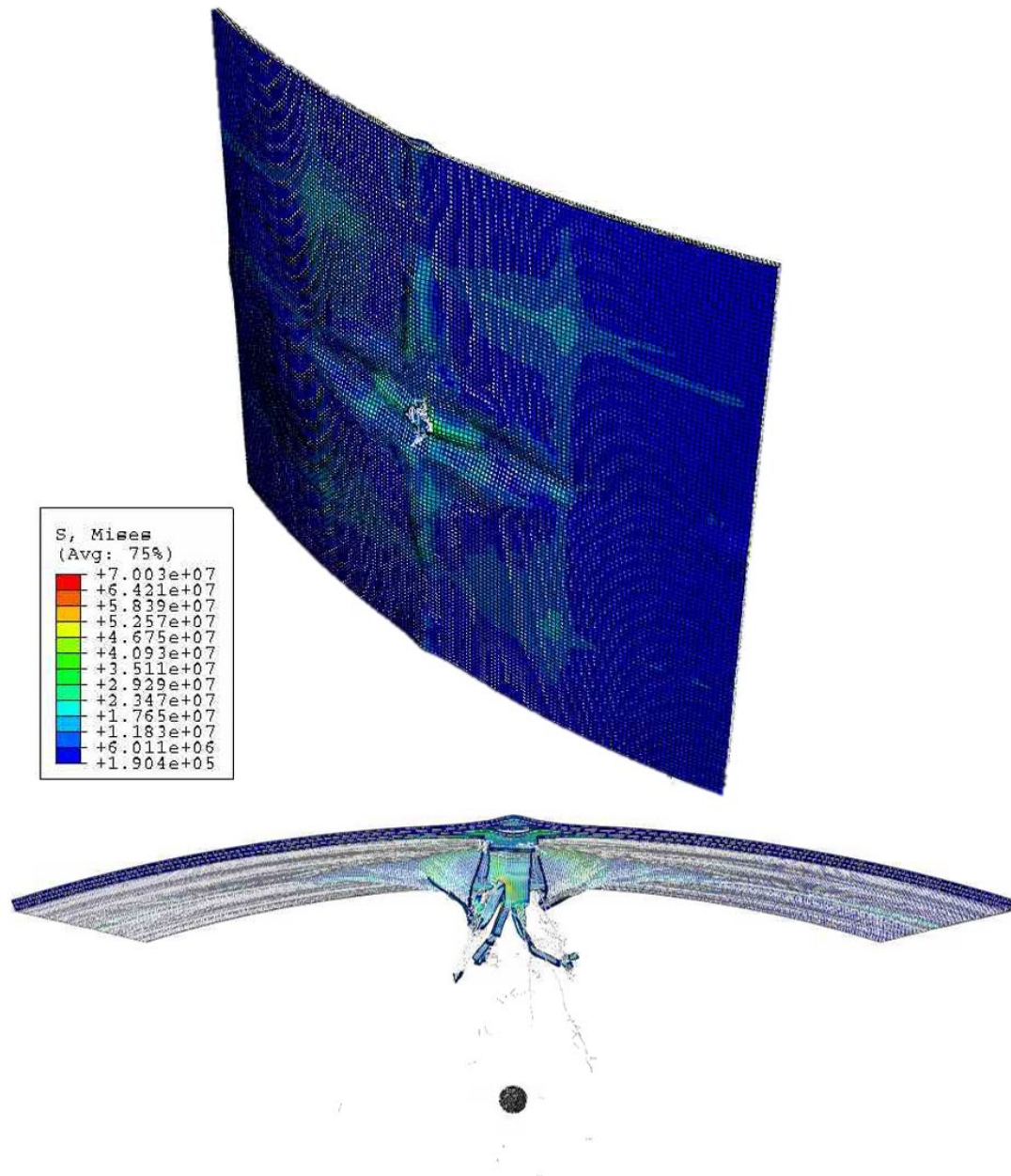


Figure 4- 31: Stress magnitudes of the impact on the curved plate from two different angles. The characteristic deformation from the 0°/90° can clearly be seen in the top image.

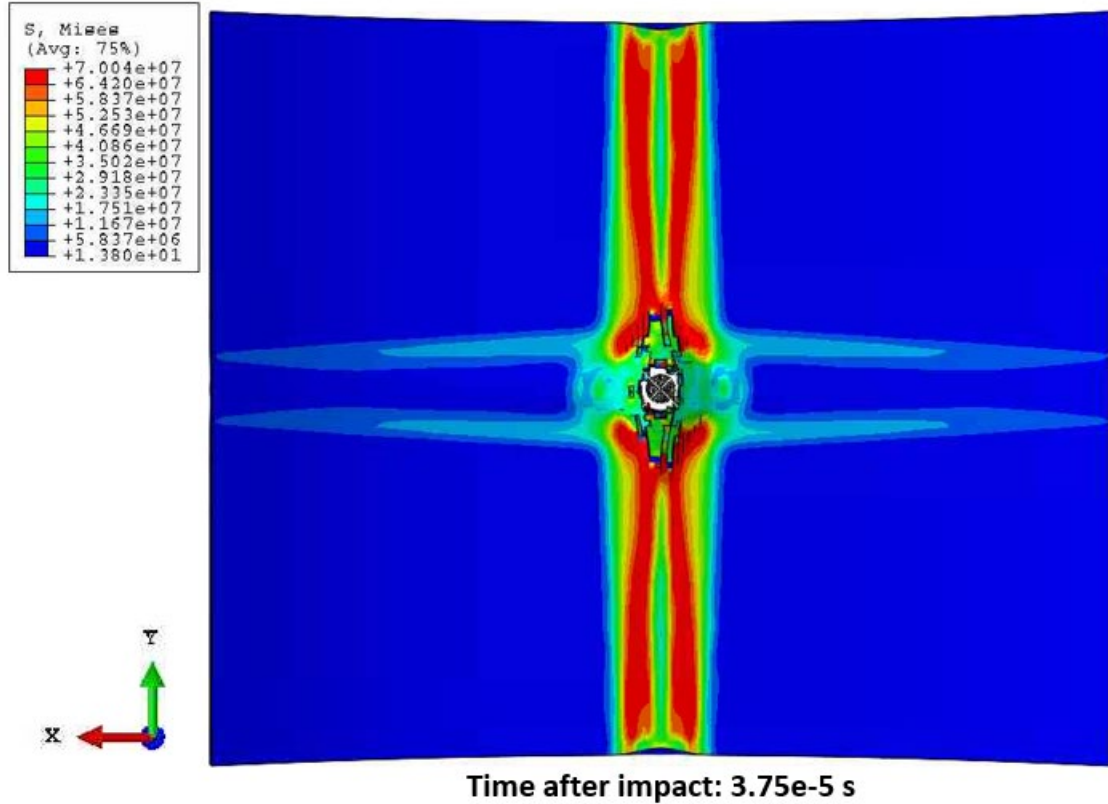


Figure 4- 32: Stress magnitudes part way through penetration viewed from the underside of the curved plate. From the stress magnitudes, it is observed that the curved fibers (aligned in the x direction) take very little of the stress load, while the flat fibers (aligned in the y direction) support most of the load.

4.5 Conclusions

Here, we have presented a proof-of-concept model of a plate of Dyneema® under impact conditions using the embedded element method to represent the cross-ply fibers grouped into truss elements. Fibers bundled together to create truss elements of 0.632mm (1380 FpT) and 1.3mm (5581 FpT) were embedded into layers of matrix mesh bonded together by cohesive contacts. The results showed that the embedded element model was able to capture several salient features including the initial shear plug failure zone and subsequent delamination and fiber snap back. Investigation of the fiber tensile stresses and the matrix shear stress under the projectile early in the

impact shows features of the indirect tension mechanism, but it is unclear if the stresses are driven by the anisotropic expansion of the layers or other mechanisms. We have shown that, while beam elements have the potential to perform better, the truss element model captures these mechanisms with an efficient run time. Future work on this model includes addressing the volume redundancy inherent in the embedded element method as well as performing more validation tests with experimental data and different sized plates and acquiring more information about the matrix strength and failure mechanisms.

Additionally, we have shown the capability of using the embedded element method to model the impact of a curved plate. The curved plate model shows that the curved fibers take far less of the impact stress than the non-curved fibers, and the stress waves travel more slowly in the curved direction, which will impact the interaction with the plate boundary conditions. The findings on the curved plate will need to be confirmed experimentally but show that there are new mechanisms in play during the impact of the curved plate as opposed to the flat plate that will need to be investigated to improve the design of curved plate armors.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

As a recap, the fundamental questions to be addressed in this effort are:

1. Can a truss element mesh be easily created for curved or complex geometry?
2. Can the embedded element method be modified to correct the volume redundancy and eliminate the need for material property smearing?
3. Does using embedded truss elements to model fiber bundles provide accurate predictions of material behavior on small scales (tension/compression tests) and larger scales (full plate impact)?

The work done to address each of these research questions will be discussed below:

5.1.1 MATLAB Script for Embedded Fiber Creation

Chapter 2 described a MATLAB script for adding embedded truss elements to a host mesh created in Abaqus. This script includes versatility for changing the size of the embedded elements and how close the fiber mesh should be to the shape by changing the size of the initial point grid used to draw the lines of the truss elements. The script creates an input file that can be immediately run in Abaqus to simplify the process of model creation. We have a modified version of this script that can map a specific type of curved plate to a flat shape to easily add the embedded truss elements and then map the whole shape along with the new embedded elements back to the original curve. This greatly simplifies the process of adding embedded elements to complex and curved

geometries. The curved plate script will need some more work to make it robust enough to handle more shapes, but in general, curved plates of fiber reinforced composite are created as a flat plate first then bent to be a curve, so the method presented here is following a similar method as the manufacturing process so most plates will fit the requirements to be mapped to a flat plate first.

5.1.2 Embedded Element Finite Element Code

Chapter 3 demonstrated that the finite element code we have developed here known as FFlagSHyP Modified (FM) is able to correct the volume redundancy inherent in the embedded element method and eliminates the need for using material property smearing as a method for addressing the redundancy. Comparing the internal and kinetic energy of a simple tension model with increasing numbers of embedded elements showed that increasing the number of embedded elements artificially increased both energy types, proving that for a model with a large percentage of the volume modeled as inclusions (such as in a model of a fiber reinforced composite) the volume redundancy must be corrected. The FM code uses a correction algorithm that can be included in other finite element codes so that the embedded element method can be used to model composite materials using true material densities and stiffness measurements which will allow for a truer modeling of the physics involved in impact or other processes and improve our understanding of material behavior and our ability to predict behavior and design for it.

5.1.3 Embedded Element Finite Element Code and Finite Element Impact Models

In Chapter 3, we showed that an accurate model of Dyneema® can be made for quasi-static tests using various numbers of embedded truss elements to represent fiber bundles. By increasing or decreasing the size of the embedded elements while keeping their total volume relatively

constant, we showed that these changes in mesh sizing did not impact the response of the model. This shows that the mesh is scalable for situations where we are looking at the full material response, such as tension and compression tests. This method could be used to model a small specimen, for example a specimen used in a Split Hopkins Pressure Bar, to calibrate the material properties of the fibers and the matrix materials used in the finite element model. The model could then be scaled up for a full plate impact test by increasing the size of the embedded truss elements, decreasing the fidelity of parts of the model in exchange for getting a model of a whole system response but still using the same material properties that were used in the small calibration model.

Chapter 4 proves examples of larger scale plate impact models of Dyneema® successfully modeled using the embedded element method. By combining the embedded element method with a matrix mesh divided into layers and held together with cohesive contacts, we were able to capture many of the salient features of the impact and failure mechanics of a Dyneema® plate when compared to experimental images. Unfortunately, the plate impact models could not be run with the volume redundancy correction. Even so, we showed that using the embedded element method allowed the stresses in the truss elements (fibers) and matrix material to be examined independently and can help show the transfer of load between the components as it occurs in the indirect tension mechanism, which is important for gaining an understanding of how the material behaves and what components drive failure. We also showed a proof-of-concept curved plate geometry modeled using embedded elements. The curvature of the fibers seems to influence the stress wave propagation through the material and will be important to investigate for the design of curved plate armor.

5.2 Future Work

5.2.1 Volume Redundancy Guidelines Paper Concept

This section documents the progress for the paper on recommendations for how the volume redundancy in the embedded element method should be addressed for different situations. While the volume redundancy correction does not exist in Abaqus, we would like to demonstrate some of the possibilities using the embedded element method for this type of modeling has. Most papers that use the embedded element either state (without showing) that the volume redundancy is negligible or reduce the stiffness and density of one of the two materials in the model. In this paper, we would like to show when or if these are acceptable solutions and when the redundancy should be removed in the solution method itself. We plan to do this by modeling simple tension/compression test of cubes of three different materials: Dyneema®, rebar reinforced concrete, and brain white matter. Each material will be modeled using embedded elements with three implementation methods: Abaqus (volume redundancy uncorrected), FLaGSHyP Modified (with volume redundancy correction), and Abaqus with the “matrix” stiffness and density reduced to correct for volume redundancy. We hope to show that the two types of correction are similar for the concrete material, since it has a low fiber volume fraction, but significantly different for the high-volume fraction materials of Dyneema® and brain matter.

To that end, we selected material properties for each of the materials (Dyneema®, rebar reinforced concrete, and brain white matter). Material properties for Dyneema® in **Table 5- 1** are the calibrated properties from the work done for the IMECE conference in November, concrete properties in **Table 5- 2** are from a paper by Garg et al. 2009[29] on FE modeling of reinforced concrete box culverts, and the brain material properties in **Table 5- 3** are from a paper by Garmella et al. 2017[83]. Note that while Dyneema® and the concrete can be modeled as linear elastic

materials, the brain material model consists of a Mooney Rivlin matrix material and an Ogden fiber. These two material types are not available in FLaGSHyP Modified yet and will need to be added to accurately model this material.

Table 5- 1: Material properties used in our previous embedded element models for Dyneema®

Dyneema®

	Density		Elastic Modulus		ν
Matrix	1100	kg/m ³	1.1	GPa	0.3
Fibers	980	kg/m ³	180	GPa	0.49
Fiber Volume Fraction	83%				

Table 5- 2: Material properties for steel reinforced concrete taken from Garg et al.[29]

Concrete, Garg et al. 2009[29]

	Density		Elastic Modulus		ν
Matrix	2400	kg/m ³	27,579	MPa	0.17
Fibers	7850	kg/m ³	200,000	MPa	0.3
Fiber Volume Fraction	~1.05%				

Table 5- 3: Material properties for brain white matter taken from Garmella et al. [83]. Note that the matrix material is modeled as a Mooney Rivlin material and the fibers as Ogden.

Brain, Garmella et al. 2017[83]

	Density		C10		C01		D1	
Matrix	1040	kg/m ³	-100	Pa	1.2	kPa	9.09E-10	1/Pa
			α		μ		D1	
Fibers	1040	kg/m ³	5		2.5	kPa	9.09E-10	1/Pa
Fiber Volume Fraction	79%							

5.2.2 Move FLaSHyP Modified to a Faster Language

As discussed in Chapter 3, FLaSHyP Modified (FM) is a MATLAB based finite element code developed from a code used for teaching the algorithm basis for large strain finite element analysis and is therefore not optimized for efficiency or computing on multiple processors. Currently it takes several hours to run a model with only a few hundred host and embedded elements. There are two options to make this code more usable. One would be to rewrite the code in a faster coding language like C. The other is to use the concepts and processes in FM to create a custom process that can be used with an established code like Abaqus. Abaqus allows the creation of user subroutines, material types, and even element types. Coding the volume corrected embedded element into Abaqus might be difficult because it involves correcting both the mass and the stiffnesses of all the host elements and some information (such as which embedded elements are located in which host elements) is unavailable to user subroutines. It may require combining several types of user subroutines to correct for volume redundancy. This complexity is why creating a MATLAB code that we have full control of was pursued for this project rather than adding the capability to Abaqus. However, the subroutine option is possible, it would allow the corrected embedded element method to be used for modeling composites far sooner than rewriting the code from scratch and attempting to make that new code competitive with the speeds of established commercial codes.

5.2.3 More Experimental Validation

Some experimental validation was presented for using the embedded element method for modeling Dyneema® in Chapters 3 and 4. However, confidence in this method would be improved

by more extensive experimental validation. Chapter 3 presented a first look at tension test data but the data was used to show that the overall response of the model did not depend on the mesh density of the embedded elements. The material properties used in those finite element models were calibrated from the experimental data. For the embedded element method to be useful in the way that we hope it will, independent measurements for fiber and matrix properties should be taken from tension tests on just fibers and just matrix material. Those properties should then be used in the embedded element model and compared to tests on the full composite to see if the combination of the two components using the embedded element method does truly represent what we see in composite behavior. This was not done in this work partially because, while experimental data corresponding to single fibers is available in the literature, data on the matrix material is incredibly scarce. Some sources claim that is impractical to test the matrix material because it is such a small percentage of the total composite and the thermosetting process in the creation of the composite alters the matrix properties anyway[13], [50], [80].

In addition, the effect of high strain rates on models using the embedded element should be further researched. We have shown in Chapter 3 that the additional kinetic energy from the volume redundancy increases with increasing strain rates (i.e., increasing velocity) but have not determined how important this is for impact simulations. We have also not investigated wave propagation through the model, though we believe this is also negatively impacted by the volume redundancy.

Finally, the curved plate model has no comparison to experimental data and is only a proof-of-concept. Obviously doing plate impact experiments on Dyneema® plates and then comparing those to an embedded finite element model would be a good step to see how much the curvature of the fibers effects the failure when compared to the failure and deformation of a flat plate. From a finite element standpoint, more modeling could be done with just different sizes of plates and types of curvature to see how the model responds to these changes and try to determine

what effects could be physical and which may just be numerical or due to assumptions imposed on the model itself.

References

- [1] Y. Li, H. Fan, and X.-L. Gao, “Ballistic helmets: Recent advances in materials, protection mechanisms, performance, and head injury mitigation,” *Compos. Part B Eng.*, vol. 238, p. 109890, Jun. 2022, doi: 10.1016/j.compositesb.2022.109890.
- [2] A. Bhatnagar, *Lightweight Ballistic Composites: Military and Law-Enforcement Applications*. Woodhead Publishing, 2016.
- [3] R. Varun Raj, B. G. Liu, R. H. J. Peerlings, and V. S. Deshpande, “Influence of microstructure on the failure of ultra-high molecular weight polyethylene composite beams impacted by blunt projectiles,” *Mech. Mater.*, vol. 164, p. 104106, Jan. 2022, doi: 10.1016/j.mechmat.2021.104106.
- [4] B. H. Utomo, “High-speed Impact Modelling and Testing of Dyneema Composite,” Ph.D. dissertation, Delft University of Technology, Delft, Netherlands, 2011.
- [5] M. O’Masta, “Mechanisms of Dynamic Deformation and Failure in Ultra-High Molecular Weight Polyethylene Fiber-Polymer Matrix Composites.” University of Virginia, May 2014.
- [6] S. D. Rajan, “Numerical analysis of ballistic composite materials,” in *Lightweight Ballistic Composites*, Elsevier, 2016, pp. 327–348. doi: 10.1016/B978-0-08-100406-7.00011-8.
- [7] B. G. Liu, K. Kandan, H. N. G. Wadley, and V. S. Deshpande, “High strain rate compressive response of ultra-high molecular weight polyethylene fibre composites,” *Int. J. Plast.*, vol. 122, pp. 115–134, Nov. 2019, doi: 10.1016/j.ijplas.2019.04.005.
- [8] T. Johnson, L. Freeman, J. Hester, and J. Bell, “Ballistic Resistance Testing Techniques,” *Inst. Electr. Electron. Eng. IEEE Access*, vol. 2, pp. 1442–1455, Dec. 2014, [Online]. Available: <http://dx.doi.org/10.1109/ACCESS.2014.2377633>
- [9] T. G. Zhang, S. S. Satapathy, L. R. Vargas-Gonzalez, and S. M. Walsh, “Ballistic impact response of Ultra-High-Molecular-Weight Polyethylene (UHMWPE),” *Compos. Struct.*, vol. 133, pp. 191–201, Dec. 2015, doi: 10.1016/j.compstruct.2015.06.081.
- [10] L. R. Vargas-Gonzalez, S. M. Walsh, and J. C. Gurganus, “Examining the Relationship Between Ballistic and Structural Properties of Lightweight Thermoplastic Unidirectional Composite Laminates”.
- [11] M. K. Hazzard, R. S. Trask, U. Heisserer, M. Van Der Kamp, and S. R. Hallett, “Finite element modelling of Dyneema® composites: From quasi-static rates to ballistic impact,” *Compos. Part Appl. Sci. Manuf.*, vol. 115, pp. 31–45, Dec. 2018, doi: 10.1016/j.compositesa.2018.09.005.
- [12] J. P. Attwood *et al.*, “The out-of-plane compressive response of Dyneema® composites,” *J. Mech. Phys. Solids*, vol. 70, pp. 200–226, Oct. 2014, doi: 10.1016/j.jmps.2014.05.017.
- [13] B. G. Liu, H. N. G. Wadley, and V. S. Deshpande, “Failure mechanism maps for ultra-high molecular weight polyethylene fibre composite beams impacted by blunt projectiles,” *Int. J. Solids Struct.*, vol. 178–179, pp. 180–198, Dec. 2019, doi: 10.1016/j.ijsolstr.2019.07.001.
- [14] S. West, “Designing a Human-Centric Rigid Body Armor for Female Police Officers: The Implications of Fit on Performance and Gender Inclusivity,” University of Arkansas, 2019.
- [15] Dan Yang, “Design, Performance, and Fit of Fabrics for Female Body Armour.” University of Manchester, 2011.
- [16] Y. Liang, X. Chen, and C. Soutis, “Review on Manufacture of Military Composite Helmet,” *Appl. Compos. Mater.*, vol. 29, no. 1, pp. 305–323, Feb. 2022, doi: 10.1007/s10443-021-09944-5.

- [17] F. Alkhatib, E. Mahdi, and A. Dean, "Design and Evaluation of Hybrid Composite Plates for Ballistic Protection: Experimental and Numerical Investigations," *Polymers*, vol. 13, no. 9, p. 1450, Apr. 2021, doi: 10.3390/polym13091450.
- [18] D. S. Stargel, "Experimental and Numerical Investigation into the Effects of Panel Curvature on the High Velocity Ballistic Impact Response of Aluminum and Composite Panels," Ph.D. dissertation, University of Maryland, College Park, United States -- Maryland, 2005.
- [19] M. Pasquali and P. Gaudenzi, "Effects of curvature on high-velocity impact resistance of thin woven fabric composite targets," *Compos. Struct.*, vol. 160, pp. 349–365, Jan. 2017, doi: 10.1016/j.compstruct.2016.10.069.
- [20] T. G. Zhang, L. R. Vargas-Gonzalez, J. C. Gurganus, and S. S. Satapathy, "Effects of Curvature and Architecture on Ballistic Performance of UHMWPE Helmets," in *Volume 3: Biomedical and Biotechnology Engineering*, Salt Lake City, Utah, USA: American Society of Mechanical Engineers, Nov. 2019, p. V003T04A055. doi: 10.1115/IMECE2019-11566.
- [21] P. Tan, "Ballistic protection performance of curved armor systems with or without debondings/delaminations," *Mater. Des.*, vol. 64, pp. 25–34, Dec. 2014, doi: 10.1016/j.matdes.2014.07.028.
- [22] T. Belytschko, J. Fish, and B. E. Engelmann, "A finite element with embedded localization zones," *Comput. Methods Appl. Mech. Eng.*, vol. 70, no. 1, pp. 59–89, Sep. 1988, doi: 10.1016/0045-7825(88)90180-6.
- [23] J. Fish and T. Belytschko, "Elements With Embedded Localization Zones for Large Deformation Problems," *Comput. Struct.*, vol. 30, pp. 247–256, Nov. 1988.
- [24] J. Fish and S. Markolefas, "The s-version of the finite element method for multilayer laminates," *Int. J. Numer. Methods Eng.*, vol. 33, no. 5, pp. 1081–1105, Apr. 1992, doi: 10.1002/nme.1620330512.
- [25] J. Fish, "The s-version of the finite element method," *Comput. Struct.*, vol. 43, no. 3, pp. 539–547, May 1992, doi: 10.1016/0045-7949(92)90287-A.
- [26] B. N. Cox, W. C. Carter, and N. A. Fleck, "A binary model of textile composites—I. Formulation," *Acta Metall. Mater.*, vol. 42, no. 10, pp. 3463–3479, Oct. 1994, doi: 10.1016/0956-7151(94)90479-0.
- [27] Q. Yang and B. Cox, "Spatially Averaged Local Strains in Textile Composites Via the Binary Model Formulation," *J. Eng. Mater. Technol.*, vol. 125, no. 4, pp. 418–425, Oct. 2003, doi: 10.1115/1.1605117.
- [28] L. Vanalli, R. R. Paccola, and H. B. Coda, "A simple way to introduce fibers into FEM models," *Commun. Numer. Methods Eng.*, vol. 24, no. 7, pp. 585–603, Jan. 2007, doi: 10.1002/cnm.983.
- [29] A. K. Garg and A. Abolmaali, "Finite-Element Modeling and Analysis of Reinforced Concrete Box Culverts," *J. Transp. Eng.*, vol. 135, no. 3, pp. 121–128, Mar. 2009, doi: 10.1061/(ASCE)0733-947X(2009)135:3(121).
- [30] J. Kang and J. E. Bolander, "Event-based lattice modeling of strain-hardening cementitious composites," *Int. J. Fract.*, vol. 206, no. 2, pp. 245–261, Aug. 2017, doi: 10.1007/s10704-017-0214-2.
- [31] U. Häussler-Combe, A. Shehni, and A. Chihadeh, "Finite element modeling of fiber reinforced cement composites using strong discontinuity approach with explicit representation of fibers," *Int. J. Solids Struct.*, vol. 200–201, pp. 213–230, Sep. 2020, doi: 10.1016/j.ijsolstr.2020.04.036.
- [32] M. Zako, Y. Uetsuj, and T. Kurashiki, "Finite element analysis of damaged woven fabric composite materials," *Compos. Sci. Technol.*, vol. 63, no. 3–4, pp. 507–16, 2003.

- [33] W.-G. Jiang, S. R. Hallett, and M. R. Wisnom, "Development of Domain Superposition Technique for the Modelling of Woven Fabric Composites," in *Mechanical Response of Composites*, in Computational Methods in Applied Sciences, vol. 10. Dordrecht: Springer Netherlands, 2008, pp. 281–291. doi: 10.1007/978-1-4020-8584-0_14.
- [34] D. Ohyama, T. Kurashiki, Y. Watanabe, Y. Fujita, and M. Zako, "Estimation of Mechanical Behavior of Braided Composites Based on Mesh Superposition Method," presented at the International Conference on Composite Materials, Aug. 2011.
- [35] 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, Dec. 2009, doi: 10.1016/j.compositesa.2009.04.034.
- [36] N. T. Chowdhury, M. W. Joosten, and G. M. K. Pearce, "An embedded meshing technique (SET) for analysing local strain distributions in textile composites," *Compos. Struct.*, vol. 210, pp. 294–309, Feb. 2019, doi: 10.1016/j.compstruct.2018.11.026.
- [37] L. Gigliotti and S. T. Pinho, "Multiple length/time-scale simulation of localized damage in composite structures using a Mesh Superposition Technique," *Compos. Struct.*, vol. 121, pp. 395–405, Mar. 2015, doi: 10.1016/j.compstruct.2014.11.005.
- [38] Y. Pan, D. Sullivan, D. Shreiber, and A. Pelegri, "Finite element modeling of CNS white matter kinematics: use of a 3D RVE to determine material properties." 2013.
- [39] S. A. Yousefsani, A. Shamloo, and F. Farahmand, "Micromechanics of brain white matter tissue: A fiber-reinforced hyperelastic model using embedded element technique," *J. Mech. Behav. Biomed. Mater.*, vol. 80, pp. 194–202, Apr. 2018, doi: 10.1016/j.jmbbm.2018.02.002.
- [40] H. T. Garimella, R. R. Menghani, J. I. Gerber, S. Sridhar, and R. H. Kraft, "Embedded Finite Elements for Modeling Axonal Injury," *Ann. Biomed. Eng.*, vol. 47, no. 9, pp. 1889–1907, Sep. 2019, doi: 10.1007/s10439-018-02166-0.
- [41] S. A. 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, May 2015, doi: 10.1016/j.compstruc.2015.02.014.
- [42] J. H. Gao, X. X. Yang, and L. H. Huang, "Application of Embedded Element in the Short Fiber Reinforced Composite," *Key Eng. Mater.*, vol. 774, pp. 241–246, Aug. 2018, doi: 10.4028/www.scientific.net/KEM.774.241.
- [43] M. J. N. Jacobs, "Ballistic protection mechanisms in personal armour," *J. Mater. Sci.*, vol. 36, pp. 3137–3142, 2001.
- [44] Y. Zhu, X. Zhang, B. Xue, H. Liu, Y. Wen, and C. Xu, "High-Strain-Rate Compressive Behavior of UHMWPE Fiber Laminate," *Appl. Sci.*, vol. 10, no. 4, p. 1505, Feb. 2020, doi: 10.3390/app10041505.
- [45] D. Zhang, Y. Sun, L. Chen, S. Zhang, and N. Pan, "Influence of fabric structure and thickness on the ballistic impact behavior of Ultrahigh molecular weight polyethylene composite laminate," *Mater. Des. 1980-2015*, vol. 54, pp. 315–322, Feb. 2014, doi: 10.1016/j.matdes.2013.08.074.
- [46] G. Mo, Q. Ma, Y. Jin, W. Yan, Z. Li, and Z. Wu, "Delamination process in cross-ply UHMWPE laminates under ballistic penetration," *Def. Technol.*, p. S2214914720303317, May 2020, doi: 10.1016/j.dt.2020.05.001.
- [47] A. Sharma and S. Daggumati, "Local Strain Analysis in a Uni-Directional Fiber-Reinforced Composite: DIC Versus FEA," in *Innovation in Materials Science and Engineering*, J. Chattopadhyay, R. Singh, and O. Prakash, Eds., Singapore: Springer Singapore, 2019, pp. 161–169. doi: 10.1007/978-981-13-2944-9_16.

- [48] L. H. Nguyen, T. R. Lässig, S. Ryan, W. Riedel, A. P. Mouritz, and A. C. Orifici, "A methodology for hydrocode analysis of ultra-high molecular weight polyethylene composite under ballistic impact," *Compos. Part Appl. Sci. Manuf.*, vol. 84, pp. 224–235, May 2016, doi: 10.1016/j.compositesa.2016.01.014.
- [49] Y. Wen, C. Xu, S. Wang, and R. C. Batra, "Analysis of behind the armor ballistic trauma," *J. Mech. Behav. Biomed. Mater.*, vol. 45, pp. 11–21, May 2015, doi: 10.1016/j.jmbbm.2015.01.010.
- [50] S. Chocron *et al.*, "Modeling unidirectional composites by bundling fibers into strips with experimental determination of shear and compression properties at high pressures," *Compos. Sci. Technol.*, vol. 101, pp. 32–40, Sep. 2014, doi: 10.1016/j.compscitech.2014.06.016.
- [51] T. A. Bogetti, M. Walter, J. Staniszewski, and J. Cline, "Interlaminar shear characterization of ultra-high molecular weight polyethylene (UHMWPE) composite laminates," *Compos. Part Appl. Sci. Manuf.*, vol. 98, pp. 105–115, Jul. 2017, doi: 10.1016/j.compositesa.2017.03.018.
- [52] L. Iannucci and D. Pope, "High velocity impact and armour design," *Express Polym. Lett.*, vol. 5, no. 3, pp. 262–272, 2011, doi: 10.3144/expresspolymlett.2011.26.
- [53] M. Grujicic *et al.*, "A ballistic material model for cross-plyed unidirectional ultra-high molecular-weight polyethylene fiber-reinforced armor-grade composites," *Mater. Sci. Eng. A*, vol. 498, no. 1–2, pp. 231–241, Dec. 2008, doi: 10.1016/j.msea.2008.07.056.
- [54] I. Lapczyk and J. A. Hurtado, "Progressive damage modeling in fiber-reinforced materials," *Compos. Part Appl. Sci. Manuf.*, vol. 38, no. 11, pp. 2333–2341, Nov. 2007, doi: 10.1016/j.compositesa.2007.01.017.
- [55] P. C. Chou, J. Carleone, and C. M. Hsu, "Elastic Constants of Layered Media," *J. Compos. Mater.*, vol. 6, no. 1, pp. 80–93, Jan. 1972, doi: 10.1177/002199837200600107.
- [56] M. W. Joosten, M. Dingle, A. Mouritz, A. A. Khatibi, S. Agius, and C. H. Wang, "A hybrid embedded cohesive element method for predicting matrix cracking in composites," *Compos. Struct.*, vol. 136, pp. 554–565, Feb. 2016, doi: 10.1016/j.compstruct.2015.10.030.
- [57] F. K. F. Radtke, "Computational modelling of fibre-reinforced cementitious composites: an analysis of discrete and mesh-independent techniques," Ph.D. dissertation, Delft University of Technology, Delft, Netherlands, 2012.
- [58] L. W. Meyer and M. Mayer, "New Strain Rate Dependent Material Model for Fiber Reinforced Composites," in *Time Dependent Constitutive Behavior and Fracture/Failure Processes, Volume 3*, T. Proulx, Ed., in Conference Proceedings of the Society for Experimental Mechanics Series. New York, NY: Springer New York, 2011, pp. 149–158. doi: 10.1007/978-1-4419-9794-4_24.
- [59] L. Puech *et al.*, "Investigating the impact behaviour of short hemp fibres reinforced polypropylene biocomposites through high speed imaging and finite element modelling," *Compos. Part Appl. Sci. Manuf.*, vol. 109, pp. 428–439, Jun. 2018, doi: 10.1016/j.compositesa.2018.03.013.
- [60] L. Iannucci, D. J. Pope, and M. Dalzell, "A Constitutive Model for Dyneema UD Composites," 2009, p. 31.
- [61] J. M. Staniszewski, S. E. Boyd, and T. A. Bogetti, "A multi-scale modeling approach for UHMWPE composite laminates with application to low-velocity impact loading," *Int. J. Impact Eng.*, vol. 159, p. 104031, Jan. 2022, doi: 10.1016/j.ijimpeng.2021.104031.
- [62] J. Bonet, A. J. Gil, and R. D. Wood, *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*. University Printing House, Cambridge CB2 8BS, United Kingdom: Cambridge University Press, 2016.

- [63] Dassault Systèmes, “SIMULIA User Assistance 2020: Abaqus/Analysis Procedures/Dynamic stress/displacement analysis/Explicit dynamic analysis/Bulk viscosity.” [Online]. Available: https://help.3ds.com/2020/english/dssimulia_established/SIMACAEKEYRefMap/simakey-r-bulkviscosity.htm?contextscope=all#simakey-r-bulkviscosity
- [64] T. Belytschko, W. K. Liu, B. Moran, and K. I. Elkhodary, *Nonlinear Finite Elements for Continua and Structures*, Second Edition. John Wiley & Sons, Ltd, 2014.
- [65] “Abaqus Analysis User’s Guide Section 35.4.”
- [66] B. P. Russell, Karthikeyan, V. S. Deshpande, and N. A. Fleck, “The High Strain Rate Response of Ultra High Molecular-Weight Polyethylene: From fibre to Laminate,” *Int. J. Impact Eng.*, pp. 1–9, 2013.
- [67] B. Sanborn, A. M. DiLeonardi, and T. Weerasooriya, “Tensile Properties of Dyneema SK76 Single Fibers at Multiple Loading Rates Using a Direct Gripping Method,” *J. Dyn. Behav. Mater.*, vol. 1, no. 1, pp. 4–14, Mar. 2015, doi: 10.1007/s40870-014-0001-3.
- [68] T. S. Gspann *et al.*, “A comparative study of the tensile failure of carbon nanotube, Dyneema and carbon fibre tows over six orders of strain rate,” *Carbon*, vol. 164, pp. 407–421, Aug. 2020, doi: 10.1016/j.carbon.2020.03.051.
- [69] M. Hudspeth, X. Nie, and W. Chen, “Dynamic failure of Dyneema SK76 single fibers under biaxial shear/tension,” *Polymer*, vol. 53, no. 24, pp. 5568–5574, Nov. 2012, doi: 10.1016/j.polymer.2012.09.020.
- [70] T. Lässig *et al.*, “A non-linear orthotropic hydrocode model for ultra-high molecular weight polyethylene in impact simulations,” *Int. J. Impact Eng.*, vol. 75, pp. 110–122, Jan. 2015, doi: 10.1016/j.ijimpeng.2014.07.004.
- [71] P. B. McDaniel *et al.*, “The effect of fiber meso/nanostructure on the transverse compression response of ballistic fibers,” *Compos. Part Appl. Sci. Manuf.*, vol. 94, pp. 133–145, Mar. 2017, doi: 10.1016/j.compositesa.2016.12.003.
- [72] S. Sockalingam, D. T. Casem, T. Weerasooriya, and J. W. Gillespie, “High Strain Rate Transverse Compression Response of Ballistic Single Fibers,” in *Dynamic Behavior of Materials, Volume 1*, J. Kimberley, L. Lamberson, and S. Mates, Eds., Cham: Springer International Publishing, 2018, pp. 51–55. doi: 10.1007/978-3-319-62956-8_9.
- [73] M. A. Iqbal, K. Senthil, P. Bhargava, and N. K. Gupta, “The characterization and ballistic evaluation of mild steel,” *Int. J. Impact Eng.*, vol. 78, pp. 98–113, Apr. 2015, doi: 10.1016/j.ijimpeng.2014.12.006.
- [74] K. Karthikeyan, B. P. Russell, N. A. Fleck, H. N. G. Wadley, and V. S. Deshpande, “The effect of shear strength on the ballistic response of laminated composite plates,” *Eur. J. Mech. - A Solids*, vol. 42, pp. 35–53, Nov. 2013, doi: 10.1016/j.euromechsol.2013.04.002.
- [75] S. Chocron, N. King, R. Bigger, J. D. Walker, U. Heisserer, and H. van der Werff, “Impacts and Waves in Dyneema® HB80 Strips and Laminates,” *J. Appl. Mech.*, vol. 80, no. 3, p. 031806, May 2013, doi: 10.1115/1.4023349.
- [76] M. K. Hazzard, P. T. Curtis, L. Iannucci, S. Hallett, and R. Trask, “An Investigation of In-Plane Performance of Ultra- High Molecular Weight Polyethylene Composites,” p. 12, 2015.
- [77] V. A. Martin, R. H. Kraft, T. H. Hannah, and S. Ellis, “An energy-based study of the embedded element method for explicit dynamics,” *Adv. Model. Simul. Eng. Sci.*, vol. 9, no. 1, p. 12, Dec. 2022, doi: 10.1186/s40323-022-00223-x.
- [78] Dassault Systèmes, “SIMULIA User Assistance 2022: Abaqus/Constraints/Embedded Elements.” Accessed: Apr. 19, 2023. [Online]. Available: https://help.3ds.com/2022/english/dssimulia_established/SIMACAECSRefMap/simacst-c-

- embeddedelement.htm?contextscope=all&id=52ee32705e334ec3a1b69df8ffd66d50#simacs
t-c-embeddedelement-t-SpecifyingTheEmbeddedNodes-sma-topic5
- [79] Dassault Systèmes, “SIMULIA User Assistance 2020: Abaqus/Elements/Structural Elements/Truss elements/Truss elements.” Accessed: Apr. 28, 2023. [Online]. Available: https://help.3ds.com/2020/English/DSSIMULIA_Established/SIMACAEELMRefMap/simaelm-c-truss.htm?contextscope=all
 - [80] J. P. Foreman, D. Porter, D. Pope, and F. R. Jones, “Predicting the Material Properties of A Polyurethane Matrix (A Composite within a Composite),” p. 5, 2012.
 - [81] F. Thomas, S. Sockalingam, S. L. Alexander, and T. Weerasooriya, “Influence of Dynamic Multiaxial Transverse Loading on Ultra High Molecular Weight Polyethylene (UHMWPE) Dyneema SK76 Single Fiber Failure,” DEVCOM Army Research Laboratory, Dec. 2020.
 - [82] X. G. Tan and A. Bagchi, “Computational Modeling of Combat Helmet Performance Analysis Integrating Blunt and Blast Loadings,” p. 9, Nov. 2020.
 - [83] 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, May 2017, doi: 10.1002/cnm.2823.

Appendix A

Derivation of a One-Dimensional Neo-Hookean Truss Element Material Model

The neo-Hooke material model gives a stress strain relation that is dependent on the material properties and the deformation gradient as shown below.

$$\boldsymbol{\sigma} = \frac{3\lambda + 2\mu}{3}(J - 1)\mathbf{I} + \mu J^{(-5/3)} \left(\mathbf{B} - \frac{1}{3}I_1\mathbf{I} \right)$$

Here bold values are 3x3 matrices, λ and μ are the Lamé constants of the material. For a deformation gradient \mathbf{F} , $\mathbf{B} = \mathbf{F}\mathbf{F}^T$, $I_1 = \text{trace}(\mathbf{B})$, and the Jacobian, (which also represents the change in volume of the material), is $J = \det(\mathbf{F})$. Truss elements only support tension and compression along their axis. However, there is not much information on how this material model should be applied to one-dimensional elements so it was derived based on the description of the nonlinear truss element from Bonet and Wood and the normal 3D neo-Hookean model.

The common measure of deformation in a truss element is the stretch. For this one-dimensional element, the deformation “tensor” becomes $F = \Lambda$, where Λ is the ratio of the current element length to the reference length.

The rest of the deformations follow from this in a 1D sense: $B = FF^T = \Lambda^2$, $I_1 = \text{trace}(B) = \Lambda^2$, $J = \det(F) = \Lambda$. Then it should be a matter of simply replacing the matrixies in the truss equation with these scalars. These stress results were checked against a single truss element in Abaqus, with the assumption that Abaqus knows how this 3D model should be applied to a 1D element, even if the documentation does not explain how that works. As shown in , this was the wrong approach and did not agree with Abaqus at all.

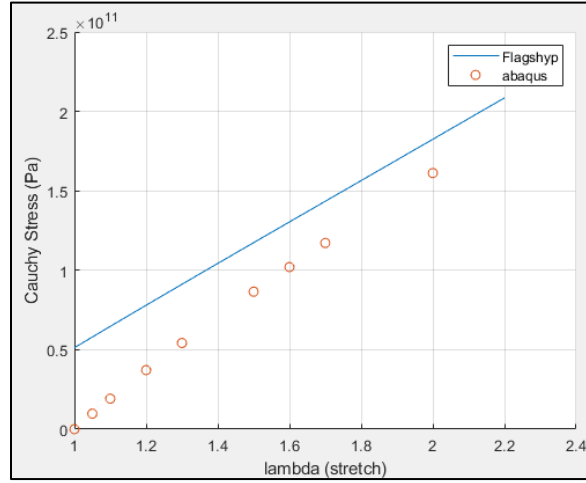


Figure A- 1: Stress results of a single truss element in Abaqus and FLaGSHyP where FLaGSHyP uses an assumed 1D neo-Hooke material formulation.

In their textbook, *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*, Bonet and Wood explain that the Jacobian of a nonlinear truss element should be $J = \Lambda^{1-2\nu}$ where ν is the Poisson's ratio of the material. This comes from the fact that when strains are large, the cross-sectional area of the truss will change a non-negligible amount. With the assumption that the change in radius of the truss can be expressed as a linear function of the axial stress: $\epsilon_r = -\nu\epsilon_a$ (ν can be calculated from Lamé's constants), the change in volume of the element can then be calculated.

In this case, nonlinear truss elements do actually have multi-dimensional strain and stress, since the radial strain is $\epsilon_r = -\nu\epsilon_a$. Then, the deformation measures are

$$\mathbf{F} = \begin{bmatrix} \Lambda & 0 & 0 \\ 0 & \Lambda^{-\nu} & 0 \\ 0 & 0 & \Lambda^{-\nu} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \Lambda^2 & 0 & 0 \\ 0 & \Lambda^{1-2\nu} & 0 \\ 0 & 0 & \Lambda^{1-2\nu} \end{bmatrix}, \quad I_1 = \Lambda^2 - 2\Lambda^{-2\nu}$$

(As a check, $\det(\mathbf{F}) = J = \Lambda^{(1-2\nu)}$ which is what was initially assumed). The deformation gradient in this case is oriented so that the $[1,1]$ term is the deformation in the direction of the truss

axis, [2,2] and [3,3] are the radial deformation (assuming radial symmetry). The other terms are zero since the truss element can't describe shear motion.

With this representation, a 3D stress tensor can be calculated for the truss. But since we are only interested in the axial stress of the truss (technically still a 1D element), the primary stress is in the [1,1] spot and we can take that as the truss stress. This method matches Abaqus almost exactly. shows analytical formulation compared to Abaqus for several different material types. The agreement is almost exact, except for when the material is nearly incompressible.

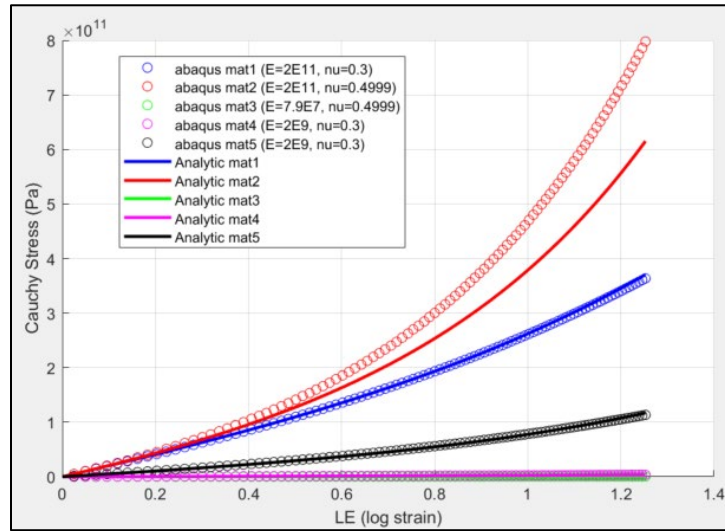
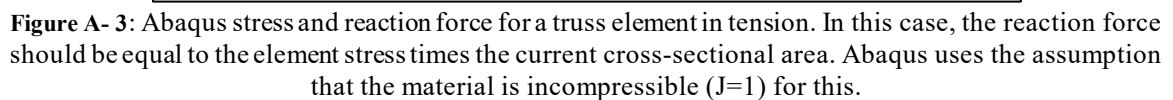


Figure A- 2: Stress results from Abaqus and FFlagSHyP for a neo-Hookean truss element in tension. Several different material types were tested to show that the final version of FFlagSHyP works for different properties.

According to the Abaqus documentation, incompressibility is dealt with by using an augmented stress equation. This would take some time to understand and there is currently no need to model incompressible materials, so the truss material model described above should be sufficient so long as incompressible test materials are avoided analytically.



Appendix B

Modified getForceEffective Subroutine

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}

Appendix C

Mild Steel: Johnson-Cook Material Properties

Table C- 1: Material properties used for the Abaqus model of the rectangular mild steel plates.

E (GPa)	203	Iqbal et al.
ν	0.33	Iqbal et al.
ρ (kg/m ³)	7850	Iqbal et al.
Ref rate (1/s)	0.001	Iqbal et al.
A (MPa)	304.33	Iqbal et al.
B (MPa)	422	Iqbal et al.
n	0.345	Iqbal et al.
c	0.0156	Iqbal et al.
m	0.87	Iqbal et al.
D1	0.1152	Iqbal et al.
D2	1.0116	Iqbal et al.
D3	-1.7684	Iqbal et al.
D4	-0.05279	Iqbal et al.
D5	0.5262	Iqbal et al.
G (J/m ²)	180,000	ASTM Ultrahigh- Strength Steels

Appendix D

Information on the GitHub Repository for FFlagSHyP Modified (FM)

The code for FM, its documentation, and the MATLAB scripts for embedded fiber creation can be found at <https://github.com/Valerie96/ExplicitFiniteElementEmbeddedVolumeCorrection>.

Valerie Martin Vita

Education

The Pennsylvania State University

Ph.D. in Mechanical Engineering, GPA: 3.92

University Park, PA

Dec 2023

The Pennsylvania State University

Master of Science in Mechanical Engineering, GPA: 3.92

University Park, PA

Dec 2021

Minor: Computer Science

Cedarville University (ABET Accredited)

Bachelor of Science in Mechanical Engineering, GPA: 3.93

Cedarville, OH

May 2018

Minor: Mathematics, Bible

Experience

Penn State University | *High-Rate Mechanics and Failure Group, Research Assistant*

University Park, PA

Oct 2018 - Present

The Air Force Institute of Technology | *Summer Research Fellow*

Dayton, OH

May 2018 - July 2018

Applied Optimization | *Mechanical Engineering Intern*

Dayton, OH

May 2017 - April 2018

Penn State University | *Toshiba-Westinghouse Fellow*

University Park, PA

May 2016 – Aug 2016

Publications

V. A. Martin, R. H. Kraft, T. H. Hannah, and S. Ellis, “An energy-based study of the embedded element method for explicit dynamics,” *Adv. Model. and Simul. in Eng. Sci.*, vol. 9, no. 1, p. 12, Dec. 2022, doi: 10.1186/s40323-022-00223-x.

V. Martin, T. Hannah, S. Ellis, and R. Kraft, “Towards Verification and Validation of Modeling Dyneema Using the Embedded Finite Element Method,” in *Volume 9: Mechanics of Solids, Structures, and Fluids; Micro- and Nano-Systems Engineering and Packaging; Safety Engineering, Risk, and Reliability Analysis; Research Posters*, Columbus, Ohio, USA: American Society of Mechanical Engineers, Oct. 2022, p. V009T12A043. doi: 10.1115/IMECE2022-96784.

T. Hannah, R. H. Kraft, V. Martin, and S. Ellis, “Implications of Statistical Spread to Experimental Analysis in a Novel Miniature Kolsky Bar,” in *Volume 3: Advanced Materials: Design, Processing, Characterization, and Applications*, Virtual, Online: American Society of Mechanical Engineers, Nov. 2020, p. V003T03A019. doi: 10.1115/IMECE2020-23976.

T. Hannah, R. H. Kraft, V. Martin, and S. Ellis, “Impact of Imperfect Kolsky Bar Experiments Across Different Scales Using Finite Elements,” presented at the ASME 2022 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers Digital Collection, Feb. 2023. doi: 10.1115/IMECE2022-96816.