

SANDIA REPORT

SAND2020-11444

Printed September 2020



Sandia
National
Laboratories

Blunt Impact Brain Injury using Cellular Injury Criterion

Ryan J. Terpsma, Chad B. Hovey

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: <http://www.osti.gov/scitech>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: <https://classic.ntis.gov/help/order-methods>



ABSTRACT

The Advanced Combat Helmet (“ACH”) military specification (“mil-spec”) requires a helmeted magnesium (“Mg”) Department of Transportation (“DOT”) headform be dropped vertically, with an impact speed of 3.1 m/s (10 ft/s), onto a steel hemispherical target. The pass/fail criteria are based on translational acceleration (150 G) alone, absent of any rotational component.

Without a rotational component, the specification’s injury risk application is limited to skull fracture and peripheral hematomas (subdural, subarachnoid), since this translational acceleration injury risk assessment is based on the Wayne State Tolerance Curve (“WSTC”). To provide a more comprehensive view of injury for the entire brain, an alternative approach is needed.

To meet this need, we worked with a larger group called PANTHER, a collaboration between national laboratories, industry, and academia. Collaborations specific to research and results presented here come from efforts led by Mr. Ron Szalkowski and Mr. Sushant Malave,¹ Ms. Alice Fawzi,² and Dr. Christian Franck.³

We have developed a prototypical injury risk criterion based on the neuronal response to abrupt changes in general motion (translation, rotation, or both). The cellular-based mild traumatic brain injury (“cbmTBI”) criterion utilizes both the strain and strain rate of brain tissue to account for the stretch and rate of stretch that occurs throughout the brain as a result of blunt impact to the head.

We conducted physical experiments of an ACH-fitted magnesium headform, which produced repeatable headform peak accelerations. Then, we developed a simulation of the experiment, and validated the simulation output with the experimental data. We then substituted the magnesium headform with a human headform, consisting of skin, muscle, bone, gray matter, white matter, cerebral-spinal fluid, membranes, vasculature, intravertebral discs, airway and sinus.

We quantified brain injury risk using the cbmTBI criterion, using the current mil-spec test and a modified test. The modified mil-spec test used an inclined anvil target that was located posterior to the crown of the helmet in the axial plane. While the current mil-spec test produced brain deformation from head translation alone, the modified test produced brain deformation from head translation and rotation, which is closer to most real world and combat theater impacts (*e.g.*, such as occur in tertiary blast exposure).

Compared to the current mil-spec test, the modified test produced elevated strains in the human digital twin. These data, mapped to the cbmTBI criterion, suggest increased injury risk for blunt impacts that cause rotation and translation, rather than just translation alone. Moreover, these data may lead to a rotational performance metric, which is rooted in the biology and pathology of the brain’s response to impact and blast, and which should be used to improve next-generation helmet designs.

¹Team Wendy Inc., Cleveland OH.

²Brown University, Providence RI.

³University of Wisconsin-Madison, Madison WI.

ACKNOWLEDGMENT

- The authors gratefully acknowledge support from the Office of Naval Research (Dr. Timothy Bentley) under PANTHER grant N0001418IP00054.
- We thank all of our PANTHER team members, with particular gratitude for the consistent and recurring collaboration with Ms. Alice Fawzi, Dr. Christian Franck, Mr. Sushant Malave., and Mr. Ron Szalkowski.
- We thank Mr. Douglas Dederman for his leadership with the ONR contract and management of Department 5421.
- We thank Dr. Jeremy Laflin for his collaboration with Python and rigid body dynamics calculations for quasi-rigid (slightly deformable) bodies.
- We thank Dr. Scott Miller for his SSM collaboration and implementation of numerous tensor quantities pertaining to the Green-Lagrange strain and strain rate.
- We thank Mr. Anirudh Patel for his numerous Python collaborations, challenging us to elevate the quality and effectiveness of our software authorship.
- We thank Dr. Adam Sokolow for his collaboration with an efficient VisIt post-processing workflow.

CONTENTS

1. Introduction	12
1.1. Framework Development	12
1.2. Framework Validation	13
1.3. Framework-Derived Insight	13
2. Experiments	15
3. Simulations	22
4. Results	27
4.1. Unhelmeted Configurations	27
4.2. Helmeted Configurations	31
4.3. Injury Risk Curve	42
5. Discussion	52
6. Conclusions	55
References	56
Appendices	60
A. Overview	60
B. Acronyms	61
C. Geometry	62
C.1. Solid Model	62
C.1.1. Historical Validations and Use	62
C.1.2. Present Use	63
D. Materials	76
D.1. Material Models	76
D.1.1. Finite Deformation Elastic	76
D.1.2. Finite Deformation Elastic-Plastic	76
D.1.3. Finite Strain Neo-Hookean	77
D.1.4. Swanson Viscoelastic	82
D.2. Material Properties	87
D.2.1. Bone	88

D.2.2. Disc	89
D.2.3. Vasculature	89
D.2.4. Airway/Sinus	90
D.2.5. Membrane	90
D.2.6. Cerebral Spinal Fluid (CSF)	91
D.2.7. White Matter	92
D.2.8. Gray Matter	93
D.2.9. Muscle	94
D.2.10. Skin	94
D.2.11. Helmet Shell	95
D.2.12. Foam	96
D.2.13. Steel	99
E. Methods	100
E.1. Quasi-Rigid Body Dynamics	100
E.1.1. Formulation	100
E.1.2. Unit Test Validation	101
E.2. Similarity	110
E.2.1. Vector Dot Product	110
E.2.2. Series Dot Product	111
E.2.3. Limitation of Cross-Correlation	112
E.2.4. p-norm	112
E.2.5. Normalized Cross-Correlation	113
E.2.6. Normalized Cross-Correlation with Time Shift	114
E.2.7. From Deterministic to Stochastic	117
E.3. Error and Error Rate	119
E.3.1. Error	119
E.3.2. Accumulated Error	119
E.3.3. Error Rate	120
E.4. Strains	122
E.4.1. Seth-Hill Strain Family	122
E.5. Strain Rates	128
E.5.1. Green-Lagrange Strain Rate	128
E.5.2. The Mises Invariant	130
E.5.3. Implementation	131
E.6. Toward Eigenvalues	131
E.6.1. Maximum Principal Strain using E	132
E.6.2. Implementation	132
F. Code	134
F.1. SSM Input Data	134
F.1.1. Green Lagrange Strain Rate and Mises Invariant	134
F.1.2. Green Lagrange Strain Eigenvalues	137
F.1.3. Green Lagrange Mises Invariant	138
F.1.4. First Piola-Kirchhoff Stress	139

F.1.5. Translational Case (Bob-063f)	141
F.1.6. Translational/Rotational Case (Bob-066b)	165
F.2. Post-Processing	190
F.2.1. Notices	193

LIST OF FIGURES

Figure 2-1.	Mil-spec experimental setup, with hemispherical anvil.	16
Figure 2-2.	Modified mil-spec experimental setup, with flat, inclined anvil.	17
Figure 2-3.	Team Wendy DOT sensor configuration.	18
Figure 2-4.	Team Wendy helmet pad configuration.	19
Figure 2-5.	Team Wendy DOT unhelmeted drop test configuration.	20
Figure 3-1.	Symmetric mesh used for DOT helmeted simulation.	23
Figure 3-2.	Tracer locations used in DOT helmeted simulation (rotated view). One tracer was placed in the center of the helmet shell thickness (“hel”), and three tracers were placed at the inferior (“inf”), middle (“mid”), and superior (“sup”) aspects of the DOT.	24
Figure 3-3.	Frontal view of helmeted human model, impacting the hemispherical anvil.	25
Figure 4-1.	Comparison of deceleration pulse from experiment and simulation.	28
Figure 4-2.	Comparison of deceleration pulse from simulation of DOT headform with Mg human head.	29
Figure 4-3.	Comparison of deceleration pulse from simulations of the unhelmeted Mg head form and the human head form.	30
Figure 4-4.	Comparison of deceleration pulses: four drop test experiments and their average.	31
Figure 4-5.	Comparison of deceleration pulses: DOT experimental average and DOT simulation.	32
Figure 4-6.	Comparison of deceleration pulses: DOT simulation and Mg head simulation.	33
Figure 4-7.	Comparison of deceleration pulses: Mg headform simulation and human headform simulation.	34
Figure 4-8.	Comparison of deceleration pulses: human headform translation and rotation.	35
Figure 4-9.	Comparison of deceleration pulses: human headform translation and rotation, with rigid skull boundary conditions.	36
Figure 4-10.	Comparison of angular acceleration curves: experiment versus simulation.	37
Figure 4-11.	Comparison of angular velocity curves: experiment versus simulation.	38
Figure 4-12.	Comparison of angular displacement curves: experiment versus simulation.	39
Figure 4-13.	Maximum principal strain (MPS) of the Green-Lagrange strain and log strain in the human head simulations of the mil-spec (Bob-063f), and the modified mil-spec (Bob-066b).	40
Figure 4-14.	Maximum principal strain rate (MPSR) of the Green-Lagrange strain rate and rate of deformation in the human head simulations of the mil-spec (Bob-063f), and the modified mil-spec (Bob-066b).	41

Figure 4-15. Reproduction of strain versus strain rate cellular injury risk curve of [Summey, 2020d], Figure 5.8.	44
Figure 4-16. Update of the Summey thesis figure.	45
Figure 4-17. Update of the Summey thesis figure, linear strain rate scale.	45
Figure 4-18. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for the mil-spec simulation (Bob-063f) at 5.8 ms.	46
Figure 4-19. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for the mil-spec simulation (Bob-063f) at 10.0 ms.	47
Figure 4-20. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for the mil-spec simulation (Bob-063f) at 11.2 ms.	48
Figure 4-21. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for modified mil-spec simulation (Bob-066b) at 8.4 ms.	49
Figure 4-22. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for modified mil-spec helmeted simulation (Bob-066b) at 13.6 ms.	50
Figure 4-23. Sequence of axial sections, showing the maximum eigenvalue of Green-Lagrange strain for the mil-spec and modified mil-spec cases, at respective times indicated in Table 4-1.	51
Figure C-1. NIH U.S. National Library of Medicine axial section, showing approximate A-A section indicated in Figure C-2.	65
Figure C-2. Frontal view of finite element human digital twin with A-A axial section indicated, the approximate location shown in Figure C-1.	66
Figure C-3. Left side view.	67
Figure C-4. Front, left, top isometric view of model.	68
Figure C-5. Front, left, top isometric view of model, with mid-sagittal exposure.	69
Figure C-6. Front, left, top isometric view of model, skin removed.	70
Figure C-7. Front view with left-side skin, muscle, and bone removed to show brain at Section B-B, shown in Figure C-8.	71
Figure C-8. Mid-sagittal B-B section view indicated in Figure C-7, with anatomical structures labelled and color coded.	72
Figure C-9. Mid-sagittal view, showing three points \mathbf{P} , \mathbf{Q} , and \mathbf{R} used for the three-point angular velocity (tpav) algorithm, and the center of mass, node 2,210,779.	73

Figure C-10. Element density versus element scale for the Sandia digital human twin model (red triangles), in context of other models (orange and magenta circles), summarized in [Giudice et al., 2019]. Triangles with bold, black borders include skin. Triangles with fine, black borders exclude skin. Triangles pointed up are the 1-mm (fine) models, with and without skin. Triangles pointed down are the 2-mm (coarse) models, with and without skin.	75
Figure D-1. Neo-Hookean volumetric strain energy density $W_{\text{VOL}}(J)$ normalized by $K/2$ for infinitesimal and finite cases.	81
Figure D-2. Swanson volumetric elastic strain energy compared to Neo-Hookean volumetric strain energy density for infinitesimal and finite cases. All strain energy density functions $W_{\text{VOL}}(J)$ are normalized by $K/2$	85
Figure D-3. Two-term viscous shear response for two different relaxation times.	86
Figure D-4. Half-symmetry view of model used for foam compression material validation.	96
Figure D-5. Team Wendy helmet pad used in compression tests.	97
Figure D-6. Test configuration for compression of the Team Wendy helmet pad.	98
Figure E-1. Reproduction of Figure 1 from [Jog and Motamarri, 2009].	102
Figure E-2. Reproduction of Figure 2 from [Jog and Motamarri, 2009].	103
Figure E-3. Rigid body phase diagram reference standard.	104
Figure E-4. Finite element mesh of deformable pendulum, used for unit testing of the three point angular velocity algorithm.	105
Figure E-5. Comparison of rigid body angular velocity reference to deformable body three point angular velocity algorithm.	106
Figure E-6. Comparison of rigid body angular position reference to deformable body three point angular velocity algorithm.	107
Figure E-7. Comparison of rigid body tip x position reference to deformable body three point angular velocity algorithm.	108
Figure E-8. Comparison of rigid body tip y position reference to deformable body three point angular velocity algorithm.	109
Figure E-9. One dimensional strain as a function of stretch ratio.	126
Figure E-10. One dimensional strain difference of strain function minus the natural logarithmic strain as a function of stretch ratio.	127

LIST OF TABLES

Table 3-1.	Summary of Simulations.	22
Table 3-2.	Human Digital Twin Model Properties.	24
Table 4-1.	Mil-spec and modified mil-spec simulations peak strain data and post-impact gross head motion description.	42
Table B-1.	Abbreviations used throughout the text.	61
Table C-1.	High-Fidelity Human Surrogate (Bob) Materials Details.	74
Table C-2.	Material color scheme.	74
Table D-1.	Material Properties Summary. All material models are finite deformation models. Elastic (E), elasto-plastic (EP), Swanson viscoelastic (SVE).	87
Table D-2.	Bone material properties.	88
Table D-3.	Vertebral discs material properties.	89
Table D-4.	Airway and sinus material properties.	90
Table D-5.	Membrane (falx and tentorium) material properties.	90
Table D-6.	Cerebral spinal fluid (CSF) material properties.	91
Table D-7.	White matter material properties.	92
Table D-8.	Gray matter material properties.	93
Table D-9.	Muscle material properties.	94
Table D-10.	Helmet shell material properties.	95
Table D-11.	Steel material properties.	99
Table E-1.	Five-point verification of sliding dot product form (E.43).	116
Table E-2.	Strains obtained from the Seth-Hill family.	123
Table E-3.	Reasonable requirements on the stretch function.	124

1. INTRODUCTION

Were there a one-sentence introduction capable of capturing, in a general sense, the scope and results of this work, it could be the following:

After we developed a simulation framework that can

- (i) represent, with high-fidelity, the anatomy and biomechanical response of the human head, and
- (ii) reproduce trusted results of drop test experiments, we demonstrated the need for modification of the current mil-spec to include rotation, based on cellular-based strain and strain rate responses of brain.

In the following sections, we have attempted to introduce and motivate our main finding—that the mil-spec must be modified to include rotation for it to have improved relevance to brain injury. While the current mil-spec has applicability to skull fracture and peripheral hematomas, it lacks accommodation for brain injury in general. To arrive at this finding, framework development and validation were required. These two constituents are introduced, in turn, below.

1.1. Framework Development

First, we undertook the large task of developing a blunt impact simulation framework that represented the human head to a high degree of fidelity. We resolved length scales of the human head to cubic volumes of 1-mm side length. For a human head with approximate radius of one decimeter (1E-1 m), we achieved two orders of magnitude for spatial resolution, using millimeters (1E-3 m). Neuronal cells have a micrometer (1E-6 m) length scale. Thus, a minimum of three more orders of magnitude would be required to resolve the spatial resolution between our continuum model and the underlying cellular anatomy. See Chapter [C](#) for additional details.

We endowed the model with constitutive descriptions of biological materials, with emphasis on the Swanson viscoelastic model for gray and white matter. See Chapter [D](#) for details.

While development of this framework required a significant time investment, it represented only the foundation of this work. As such, we have elected to place most of the framework discussion in the Appendices, in the hopes of keeping the reader focused on the main biomechanical concept—that rotation is particularly relevant to brain injury and, by corollary, the current mil-spec is inadequate for comprehensive brain injury assessment.

1.2. Framework Validation

Second, we used the current mil-spec experiments as a baseline to validate our simulation framework.

For the purposes of this work, we have defined a numerical framework to be **valid** if it reproduces simulation results that are indistinguishable from *repeatable* and *reliable* experimental results. That is, the concept of validation requires that the experiment be consistent and correct, and the simulation match the experiment to within a small tolerance. Specifically, we defined an L2-norm error rate, Equation (E.62), as our validation measure. See Sections E.2 and E.3 for details.

Our computational results replicated the mil-spec DOT headform experimental results to a high degree of accuracy (see Section 4.1). Trust in this step allowed us to sole-substitute the Sandia high-fidelity human digital twin (often colloquially referred to as “Bob”) for the DOT headform used in the mil-spec. This extrapolatory step provided insight into the deformation—in particular, strain and strain rates—of the human brain, were a human head subjected to the mil-spec experiment.

1.3. Framework-Derived Insight

Finally, we altered the mil-spec, from its in-line hemispherical target (see Figure 2-1) to an offset, inclined anvil target (see Figure 2-2), all else being equal. This updated target configuration, called the “modified mil-spec” herein, caused post-impact head rotation in excess of 10 degrees (see Figure 4-12) compared to the mil-spec baseline, which did not cause the helmeted DOT headform to rotate appreciably. Impact time durations were on the order of 10-milliseconds, with head angular velocities reaching nearly 17.5 rad/s (1,000 deg/s), and head angular accelerations peaking near 3,000 rad/s² (see Figures 4-10 and 4-11.)

Both the mil-spec and modified mil-spec configurations used the same kinetic energy dose, manifest by a helmeted human head of 5-kg, moving with an initial speed of 3.1 m/s (10 ft/s) toward a stationary target. While the input of kinetic energy for both configurations was the same, the output in terms of deformation of the brain was profoundly different. Relative to the mil-spec case, the modified mil-spec diverted more of the kinetic energy input into brain deformation output, which stemmed from a global rotational response of the skull.

The brain has significant preference to undergo shear deformation rather than volumetric deformation. The bulk modulus (K) of the brain exceeds its initial and long-term shear moduli (G_0 and G_∞ , respectively) by approximately five-to-six orders of magnitude. Bulk modulus is typically measured in GPa (1E9 Pa), whereas shear moduli are typically measured in kPa (1E3 Pa) or 10s of kPa (1E4 Pa). This disparity can also be viewed as

arising from the brain's near incompressibility, seen through the Lamé relationship for Poisson ratio (ν) as

$$\nu = \frac{3K - 2G}{2(3K + G)} \stackrel{K \gg G}{\approx} \frac{3K}{6K} = \frac{1}{2}, \quad (1.1)$$

where $\nu = 0.5$ denotes incompressibility. The brain's nearly incompressible response occurs because it is primarily composed of water, essentially an incompressible material over the pressure and temperature ranges of interest here.

A consequence of the bulk-to-shear disparity is the generalization that rotational head motion, relative to translational head motion, causes greater brain deformation through shear.

When the deformation, measured through strain, and the deformation temporal onset, measured through strain rate, is sufficient to exceed the normative physiological thresholds, tissue injury can result.

To assess potential for injury, we utilized a cbmTBI criterion, based on strain and strain rate of the brain. Our results demonstrated elevated injury potential for the modified mil-spec configuration, relative to the mil-spec baseline configuration. These results are the basis for claiming that the current mil-spec is in need of modification to include a rotational component, for better relevance to brain injury secondary to blunt impact.

2. EXPERIMENTS

The physical experiments were conducted by our collaborators Ron Szalkowski and Sushant Malave of Team Wendy, Inc., Cleveland, Ohio. The two main experiments were the mil-spec configuration (Figure 2-1) and the modified mil-spec configuration (Figure 2-2).

The experimental DOT headform consisted of a DOC-C head, a drop arm with clamp, and acceleration sensors (Figure 2-3). The experiments were performed using the Team Wendy seven-pad helmet liner configuration (Figure 2-4).

Prior to experiments having helmeted configurations, experiments using the instrumented DOT in an *unhelmeted* configuration were performed, with impact onto a MEP target (Figure 2-5). The purpose of the unhelmeted precursor was two-fold:

First, it allowed our Team Wendy collaborators to demonstrate consistency and correctness of the experimental system. Specifics of the experimental system, *e.g.*, the acquisition system specifications, are beyond the scope of this report, and we defer to Mr. Szalkowski and Mr. Malave for these details. The main experimental output of interest in this report was the headform acceleration time histories, both translational and (when relevant) rotational.

Second, it allowed us to exercise our numerical framework and assess the degree of validation with the least-complex physical system first, absent of helmet shell and helmet pads. The Team Wendy pads, composed of a hard (protective) foam layer adjacent to the helmet shell and a soft (comfort) foam layer adjacent to the head, had a sophisticated nonlinear mechanical response in compression, characterized by initial stiffness, plateau stiffness, and crush-up stiffness. We undertook significant effort to model the foam (see Section D.2.12). Our overarching validation philosophy was to complete component (unit) tests first, followed by system (integration) tests in order of increasing complexity.

Therefore, we introduced foam into the system only after it was validated on its own, and the *unhelmeted* DOT system was validated without foam, the most sensitive part of the helmeted system.



Figure 2-1. Mil-spec experimental setup, with hemispherical anvil.



Figure 2-2. Modified mil-spec experimental setup, with flat, inclined anvil.

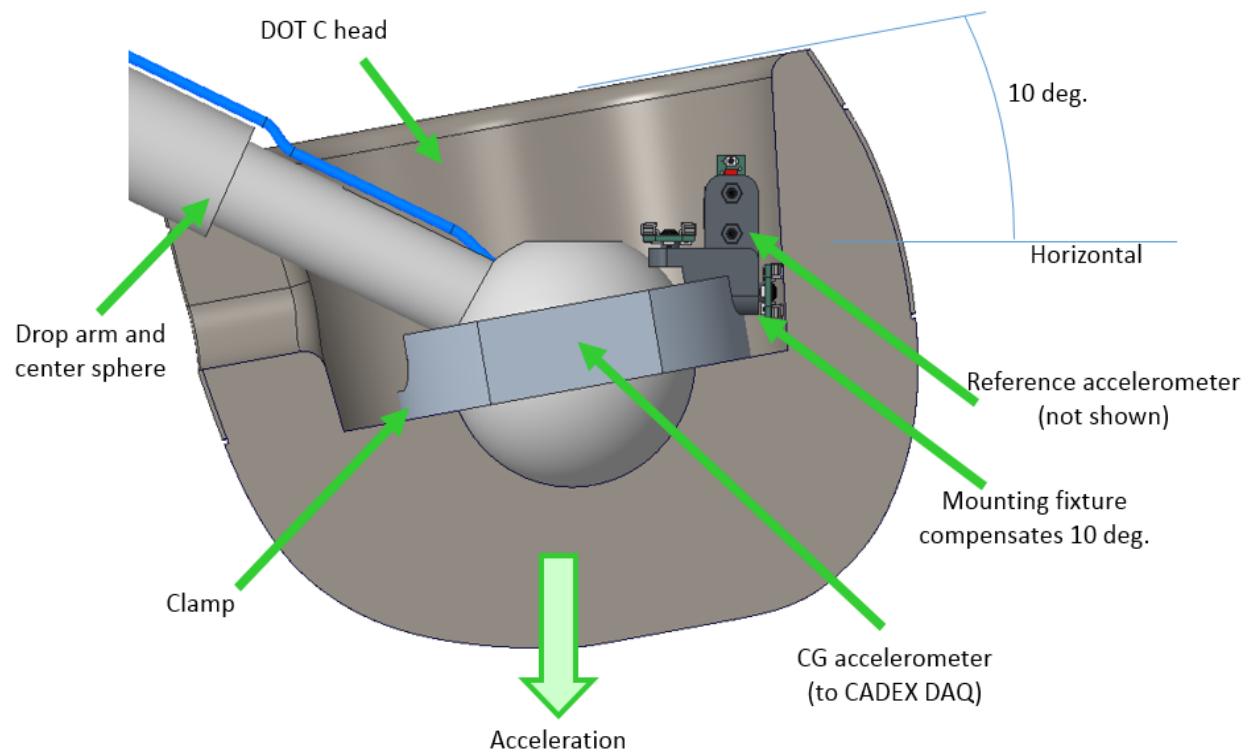


Figure 2-3. Team Wendy DOT sensor configuration.



Figure 2-4. Team Wendy helmet pad configuration.

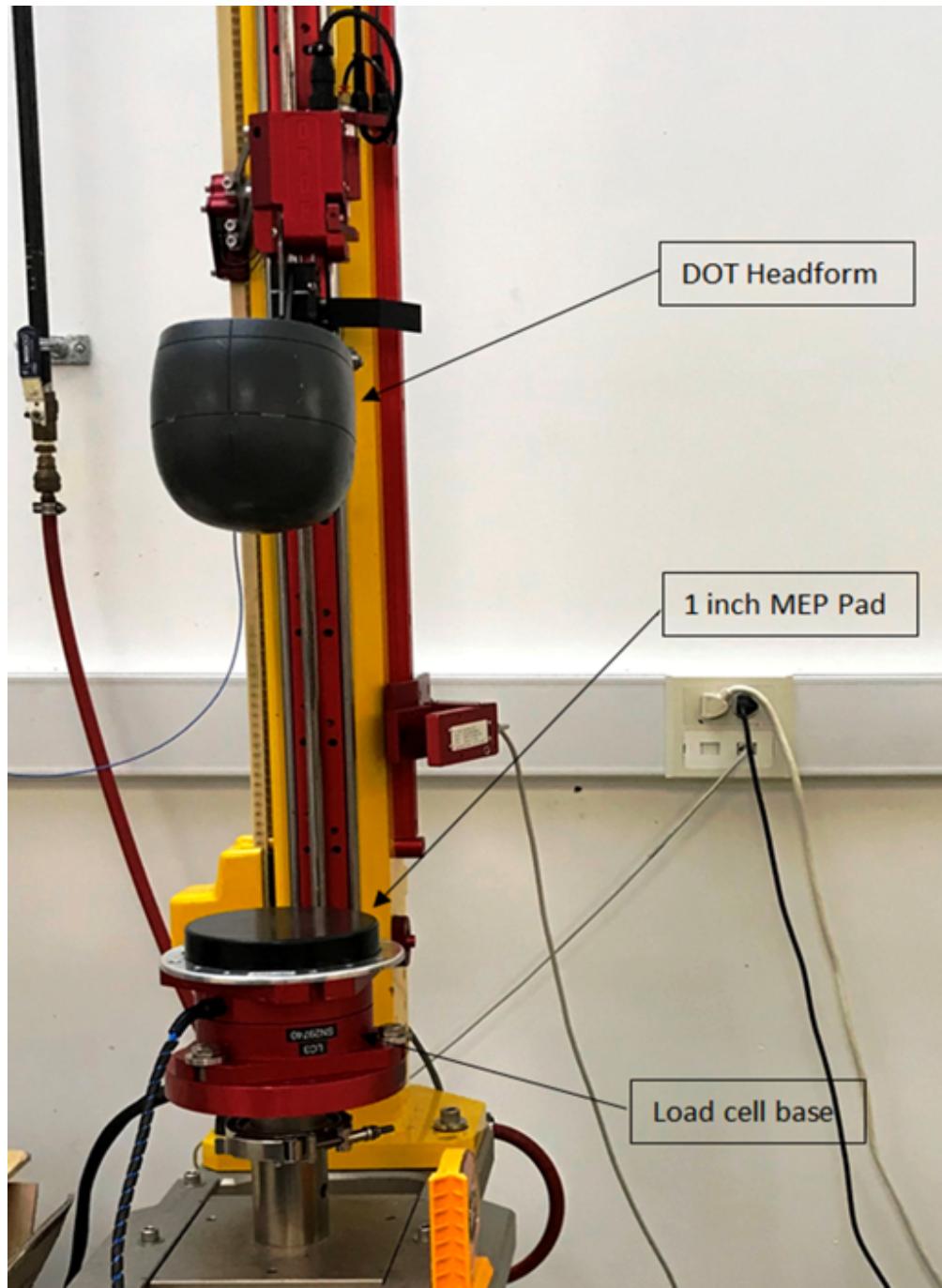


Figure 2-5. Team Wendy DOT unhelmeted drop test configuration.

Having introduced the experiments, we next introduce the simulations of those experiments in Chapter 3. The results from experiments and simulations are presented together, in Chapter 4.

3. SIMULATIONS

There are seven (7) total configurations that describe the simulations performed for this study, delineated by

- **helmet status:** unhelmeted *versus* helmeted,
- **headform composition:** DOT *versus* Mg Bob *versus* Bob, and
- **target:** MEP *versus* hemisphere *versus* inclined anvil.

Table 3-1 shows a summary of these simulations presented herein, with attributes to describe the configuration, the simulation reference number (used for internal organizational purposes), the figure containing the summary acceleration response results, and whether the basis for comparison is either direct experimental or relative simulations.

Table 3-1. Summary of Simulations.

Configuration	Simulation Reference	Acceleration Metric	Comparison Basis
Unhelmeted			
DOT onto MEP	Dot-025	Fig. 4-1	direct experimental
Mg Bob onto MEP	Bot-026	Fig. 4-2	relative simulations
Bob onto MEP	Bob-033c	Fig. 4-3	relative simulations
Helmeted			
DOT onto hemi	Dot-062	Fig. 4-5	direct experimental
Mg Bob onto hemi	Bot-064b	Fig. 4-6	relative simulations
Bob onto hemi	Bob-063f	Fig. 4-7	relative simulations
Bob onto inclined anvil	Bob-066b	Fig. 4-8	relative simulations

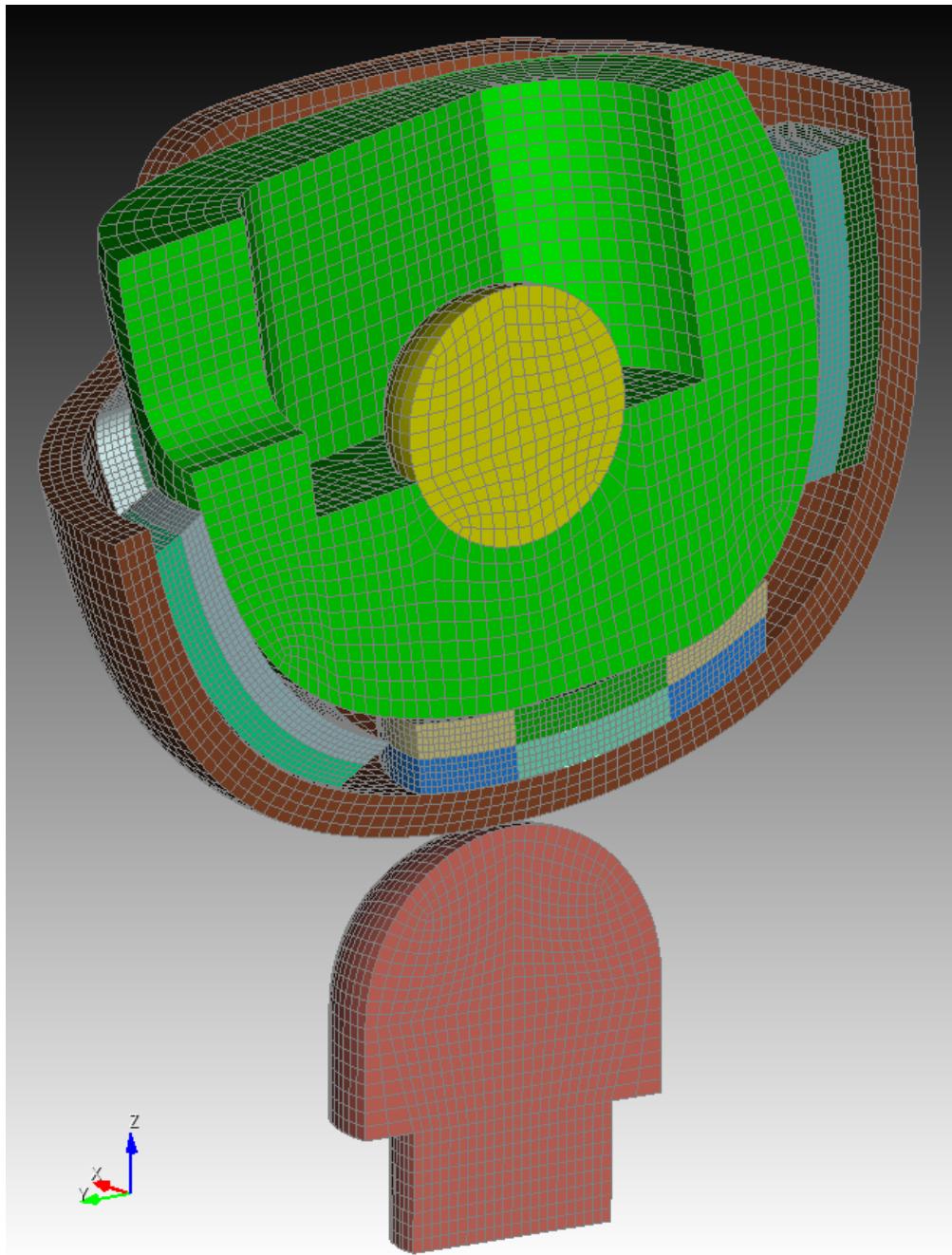


Figure 3-1. Symmetric mesh used for DOT helmeted simulation.

Figures 3-1 and 3-2 show details of the DOT-related simulation geometry. The Z-axis aligned with the inferior-superior (“inf-sup”) anatomical axis. The Y-axis aligned with the anterior-posterior (“AP”) anatomical axis. The X-axis aligned with the lateral left-right axis. For all drop tests, the drop direction was aligned with the Z-axis, the unless otherwise indicated.

An overview of the Bob simulation geometry is shown in Figure 3-3 and Table 3-2, with details of the model presented in Chapters C and D.

tracers

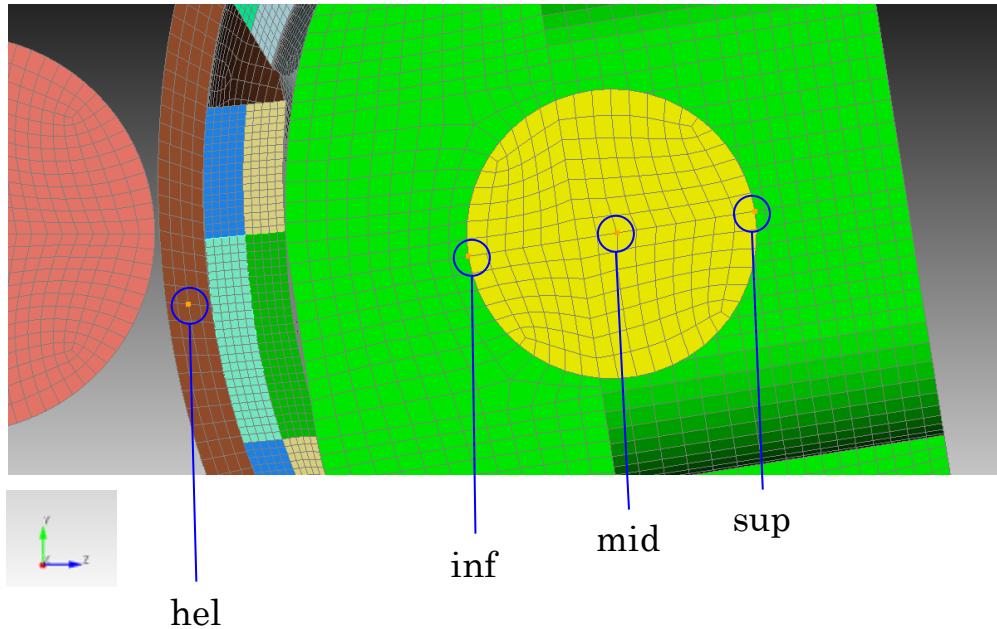


Figure 3-2. Tracer locations used in DOT helmeted simulation (rotated view). One tracer was placed in the center of the helmet shell thickness (“hel”), and three tracers were placed at the inferior (“inf”), middle (“mid”), and superior (“sup”) aspects of the DOT.

Table 3-2. Human Digital Twin Model Properties.

Geometry	# Elements	Mass (g)	Volume (cc)
High-fidelity human surrogate (Bob)	4,631,316	5,010	4,453
helmet shell	90,950	1,868	1,297
helmet pad assembly	29,820	49	843
hemispherical anvil	17,040	4,413	552
Total	4,769,126	11,340	7,145

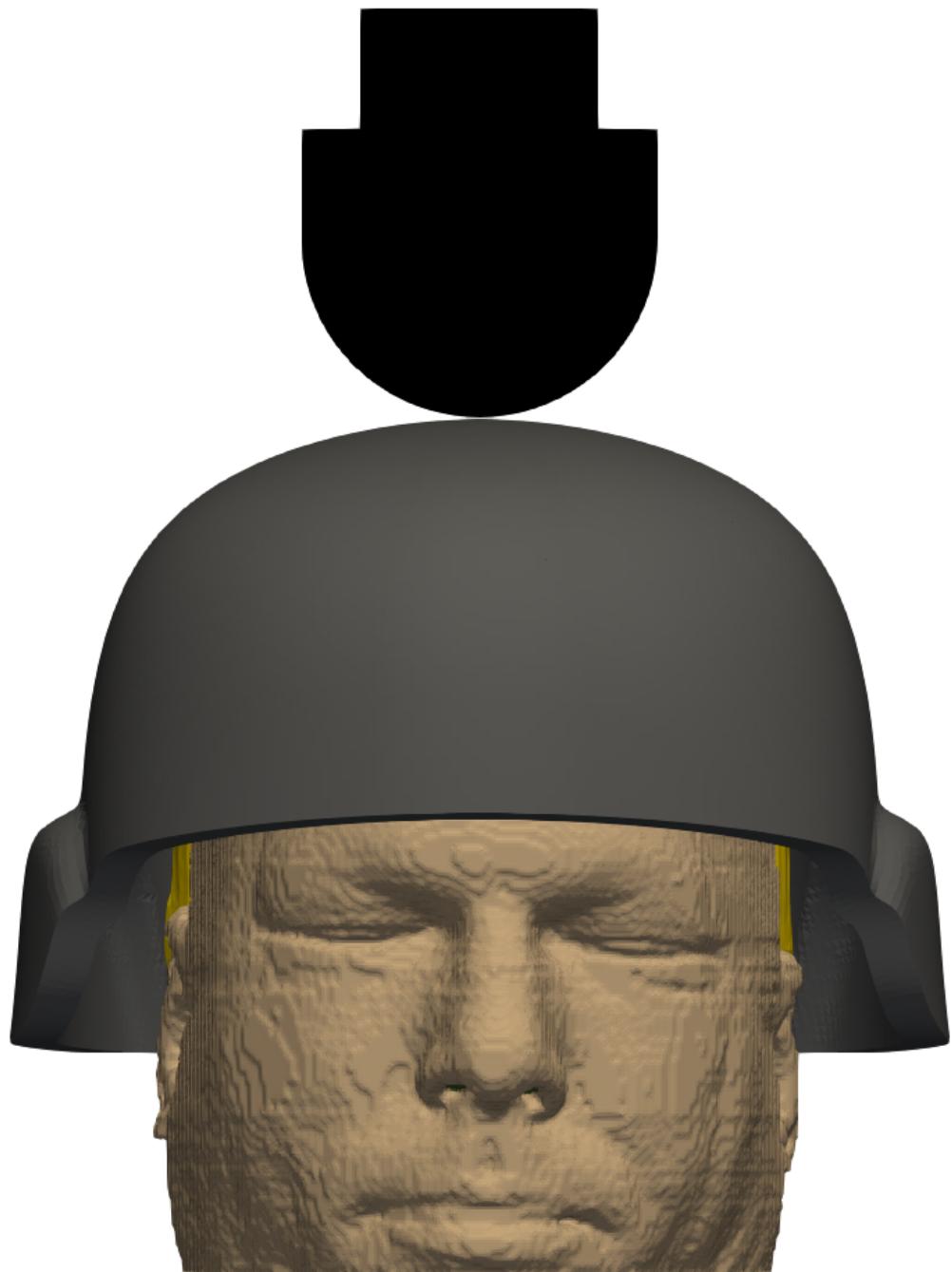


Figure 3-3. Frontal view of helmeted human model, impacting the hemispherical anvil.

For mesh pre-processing, we used Sandia National Laboratories' Cubit and Sculpt programs [Owen et al., 2011, Owen et al., 2014, Owen et al., 2019]. For processing, we used Sandia National Laboratories' Sierra Solid Mechanics (SSM) nonlinear explicit finite element solver [Merewether et al., 2020]. For Exodus file post-processing, we used Sandia National Laboratories' SEACAS and Lawrence Livermore National Laboratory's VisIt [Childs et al., 2012]. For generalized post-processing, we developed and maintained a significant code repository [Hovey, 2020] of Python [Van Rossum et al., 2007].

While the purpose of this chapter was to provide an introduction and overview to the simulations, Chapters [C](#) through [F](#) provide the reader with comprehensive details of the constituents used to build, run, and post-process the simulations.

4. RESULTS

In this chapter, we present results for the simulations, and when applicable, for the underlying experiments the simulations intended to describe. We then apply the simulation results to the cellular-based mild traumatic brain injury (“cbmTBI”) criterion, which is based on experimental results as well.

Because the mil-spec quantifies results in terms of acceleration, we have chosen to report this metric as well. Acceleration versus time data filtered with a 4th-order Butterworth low-pass filter, with cutoff frequency of 1,650 Hz, unless otherwise noted. In addition to acceleration, strain and strain rate of brain tissues are reported for the high-fidelity digital twin (“Bob”) model.

4.1. Unhelmeted Configurations

Figure 4-1 shows the center of gravity acceleration versus time response for the unhelmeted DOT impact into MEP, experiment versus simulation.¹

- The experimental acceleration peak acceleration magnitude of 242.8 G occurred at 4.9 milliseconds. For better visual comparison, we translated the experimental curve back in time by 0.002 s, making the peak appear at 2.9 ms, the same time for the simulation peak.
- The simulation peak acceleration magnitude of 243.6 G occurred at 2.9 milliseconds. The difference between experimental and simulation peak acceleration magnitude was 0.3 percent. The L2-norm error rate was 0.199852.

¹The simulation acceleration measurement of the DOT (`dot-5mm-non-0305-mep-025.i`) is at (0.58558, 0.00000, -8.06117) cm (the `a_mid` tracer), with mass of 4.990e+03 gram.

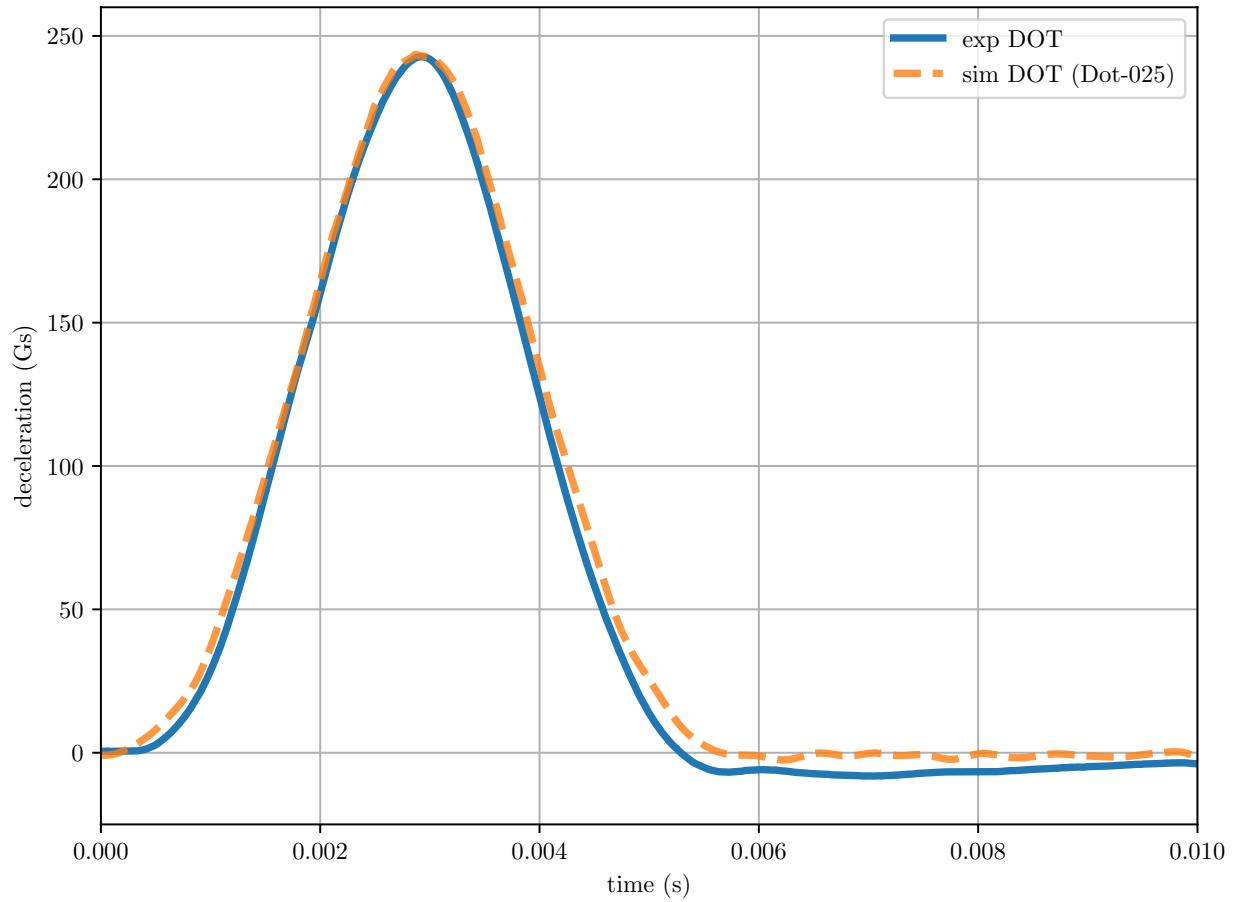


Figure 4-1. Comparison of deceleration pulse from experiment and simulation.

Figure 4-2 shows the comparison of two simulations, with the DOT headform and the Mg human head.²

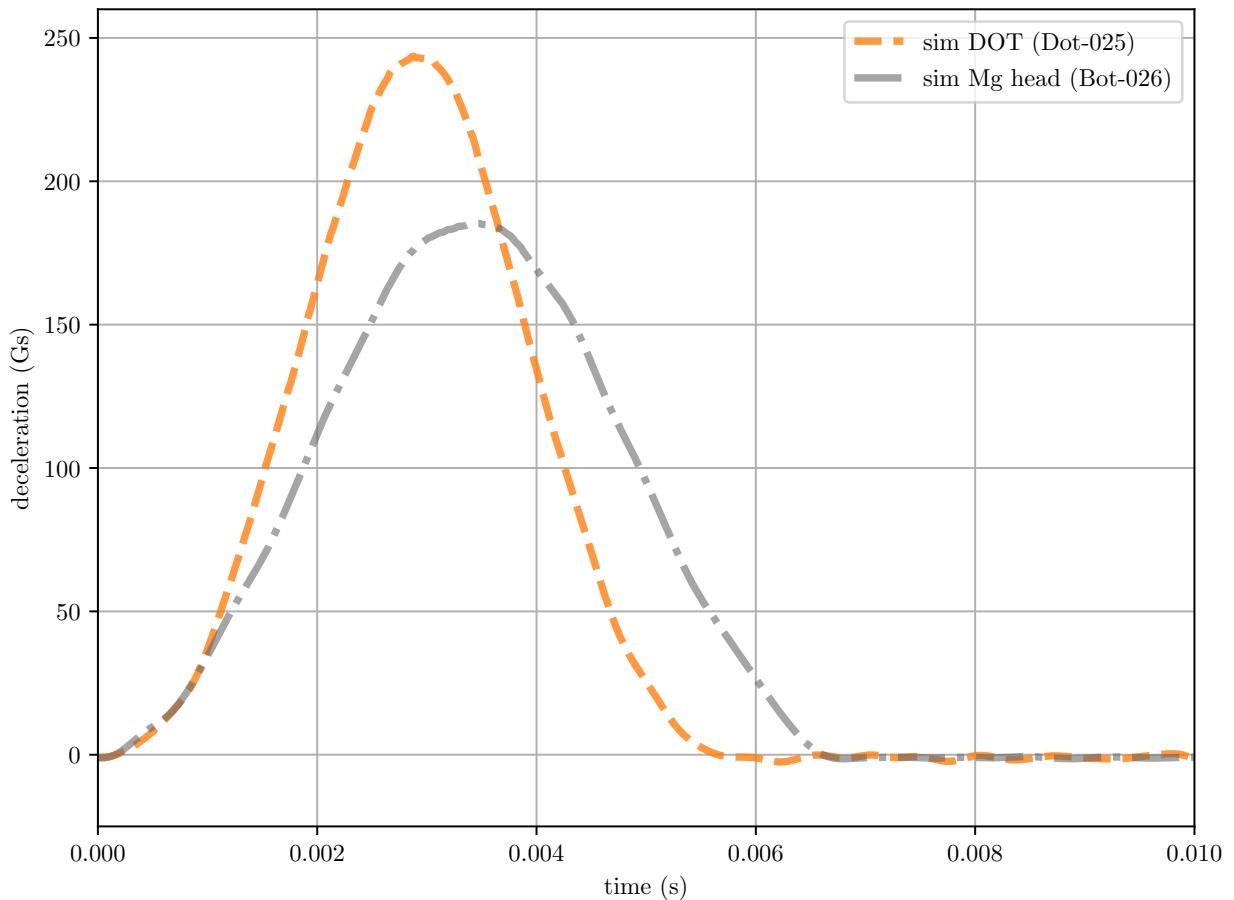


Figure 4-2. Comparison of deceleration pulse from simulation of DOT head-form with Mg human head.

²The center of mass of Bot (bot-1mm-5kg-non-0305-mep-059.i) is at (33.6440, 21.3955, 12.4797) cm, with a mass of 4.99999e+03 gram.

Figure 4-3 shows the comparison of two simulations, with the Mg human head and the human head.³ The peak acceleration magnitude of 184.4 G occurred at 4.1-milliseconds.

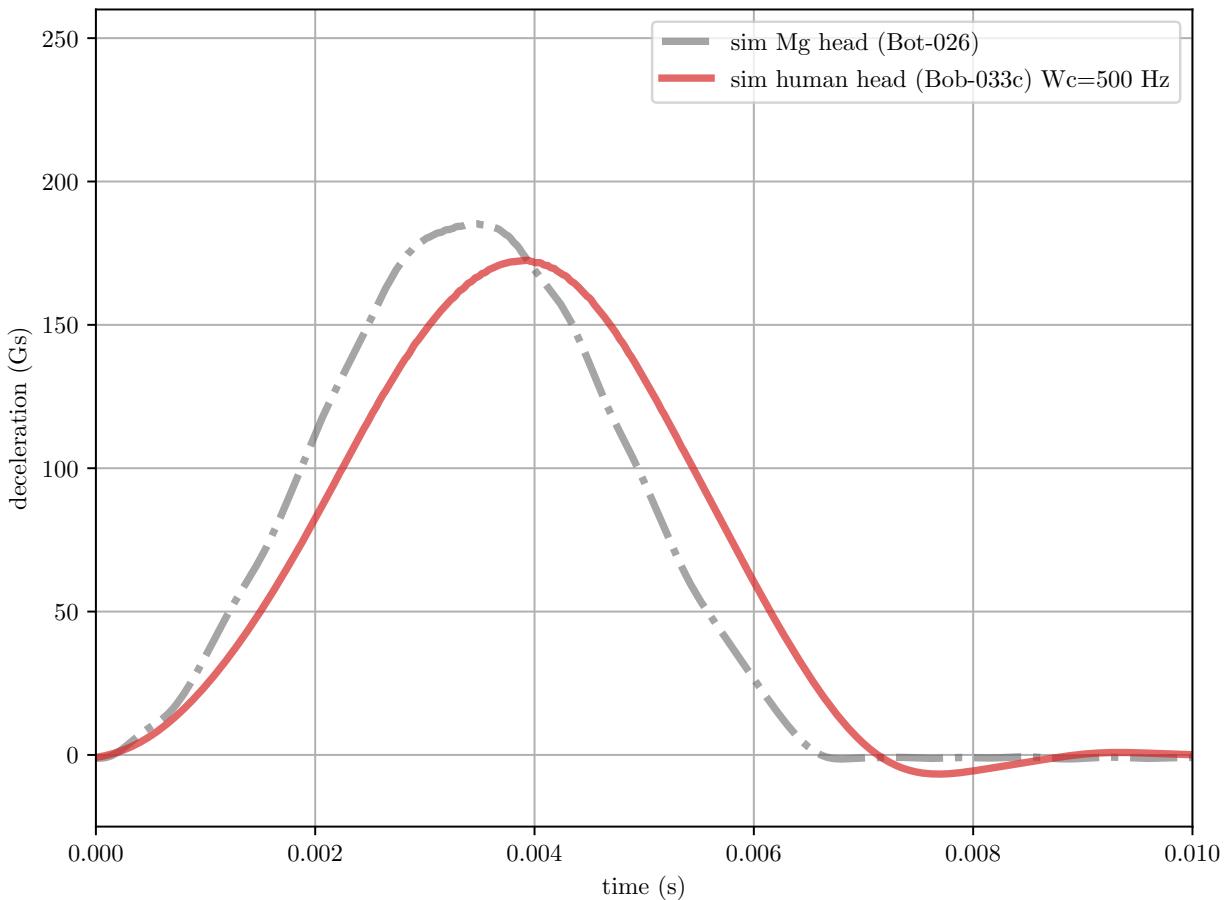


Figure 4-3. Comparison of deceleration pulse from simulations of the unhelmeted Mg head form and the human head form.

³The center of mass of Bob (bob-1mm-5kg-non-0305-mep-033.i) is at (36.6315, 21.4892, 12.3570) cm, with mass of 5.01030e+03 gram.

4.2. Helmeted Configurations

Figure 4-4 shows experimental results for four DOT drop tests experiments. The simple average is also shown.

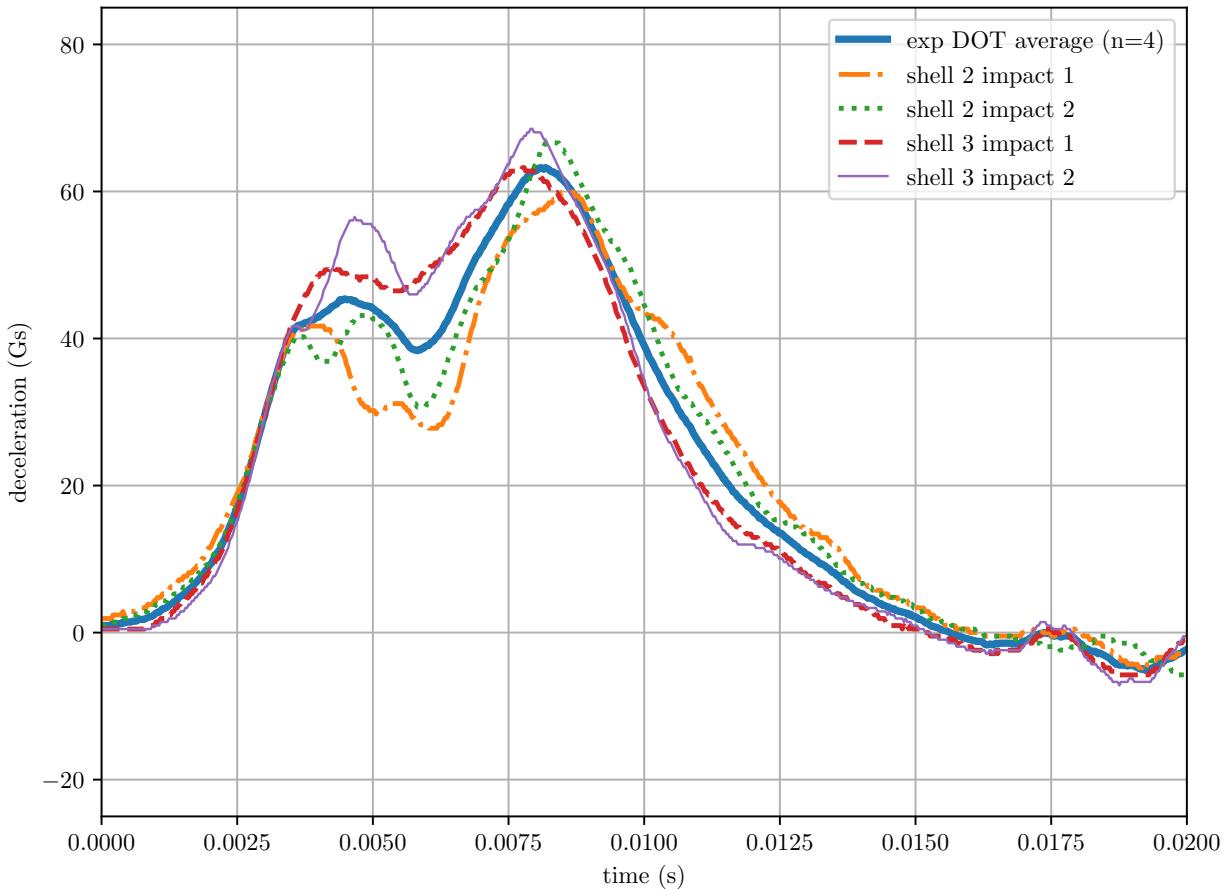


Figure 4-4. Comparison of deceleration pulses: four drop test experiments and their average.

Figure 4-5 shows a comparison of the experimental average with the simulation of the same experiment. We shifted the experimental curve in time by -1.45 ms so that the deceleration onset matched that of the simulation curve. The simulation oscillates about the experiment, which shows the high-frequency “helmet rattle” artifact that did not exist in the experiments since experiments have the chin strap tight coupling between the DOT and helmet.⁴

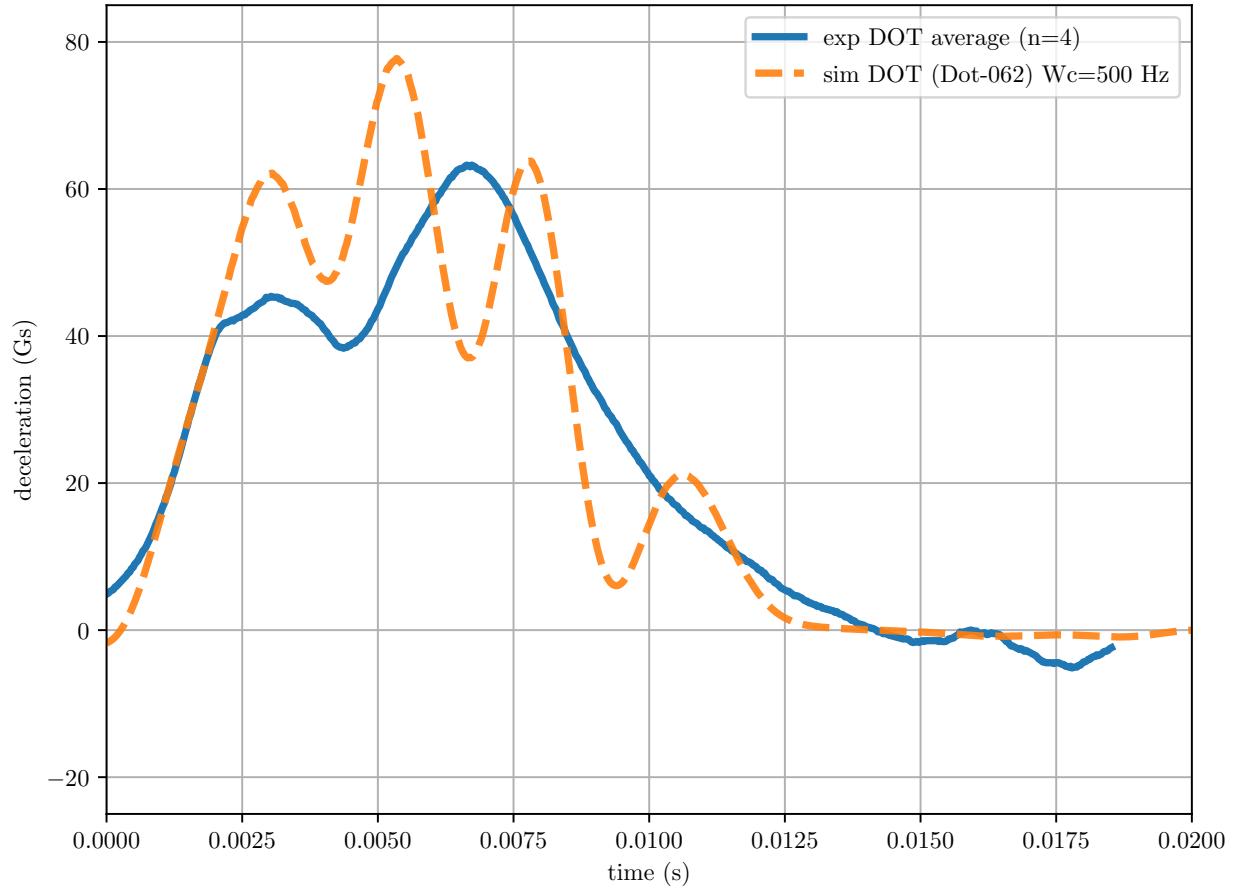


Figure 4-5. Comparison of deceleration pulses: DOT experimental average and DOT simulation.

⁴An order-of-magnitude estimate for this high-frequency is 400 Hz, taken as three cycles beginning just after 0.0025 s and ending just after 0.0100 s, 3 cycles / (0.0100 - 0.0025) s = 400 Hz.

Figure 4-6 shows a comparison of two simulations: the DOT headform with the Mg human head. The “helmet rattle” effect is visible in both simulations.

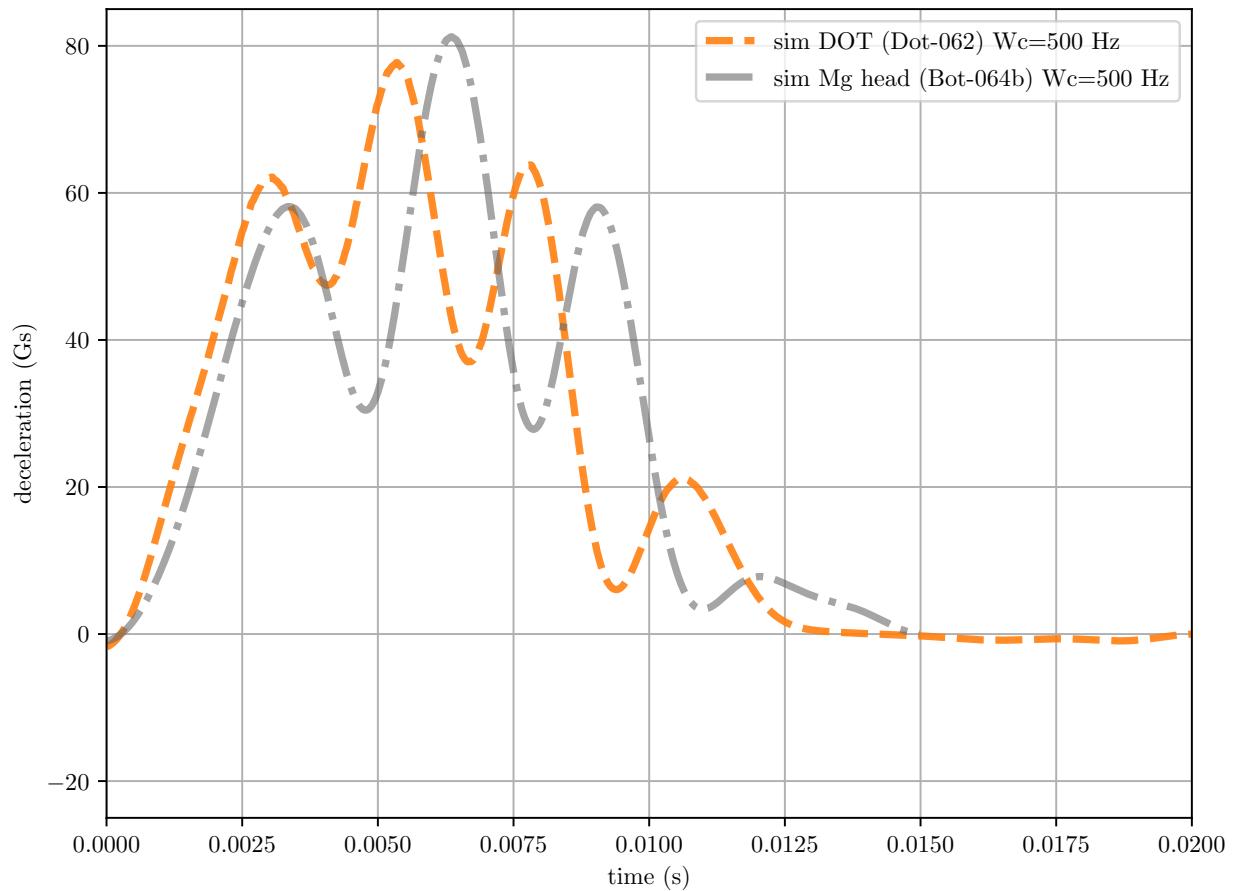


Figure 4-6. Comparison of deceleration pulses: DOT simulation and Mg head simulation.

Figure 4-7 shows a comparison of the Mg head simulation with the human head simulation. Slight attenuation from the viscoelastic response of the human head is present.

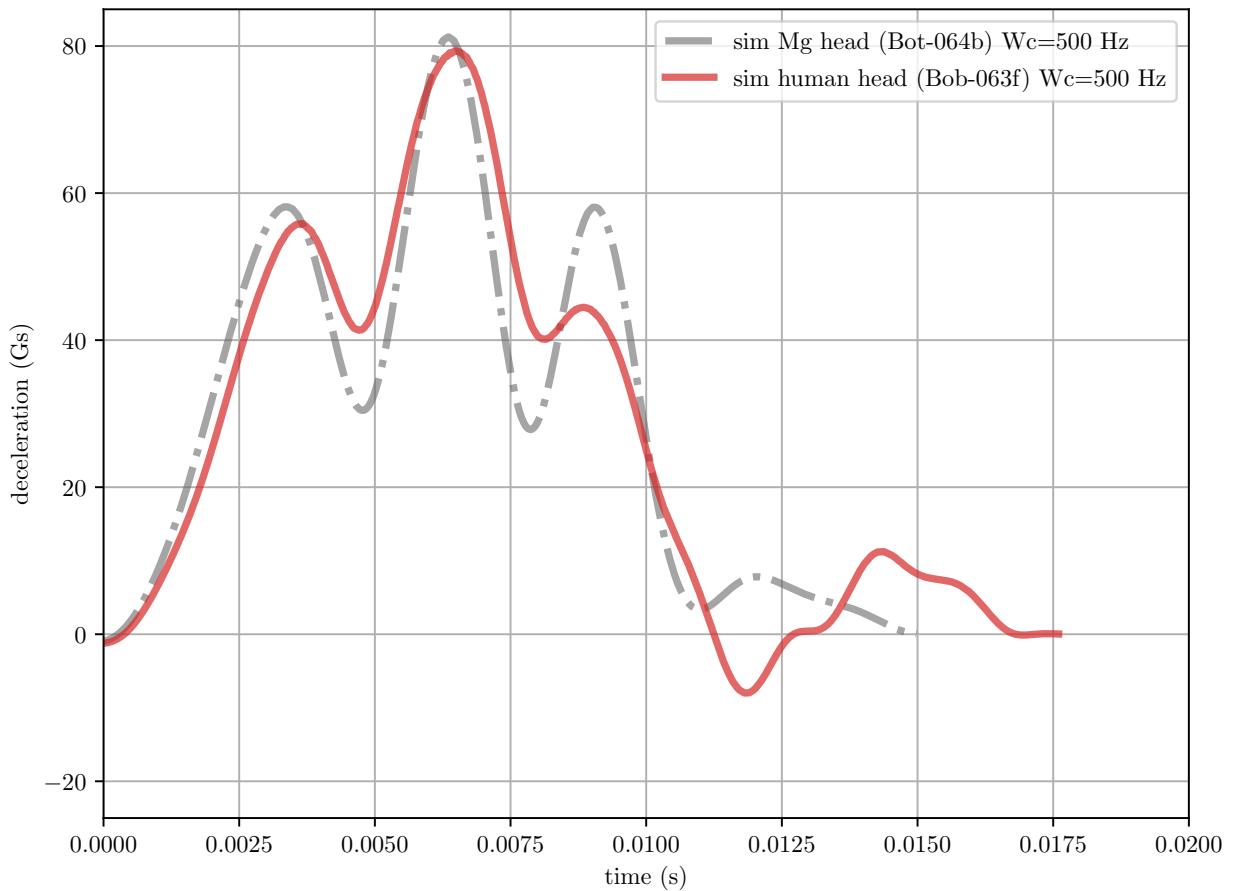


Figure 4-7. Comparison of deceleration pulses: Mg headform simulation and human headform simulation.

Figure 4-8 compares the acceleration response of the human subjected to the mil-spec, compared to the modified mil-spec. The modified mil-spec acceleration magnitudes are less than those for the mil-spec because the impact energy in the former case gets divided into a translation and rotation response.

The modified mil-spec presents a double peak acceleration, similar to the patterns seen in the experiment, since the “helmet rattle” phenomenon is evoked less than in the mil-spec case.

Finally, the modified mil-spec lateral deceleration (Y) is negative because the inclined anvil impact accelerates the head laterally.

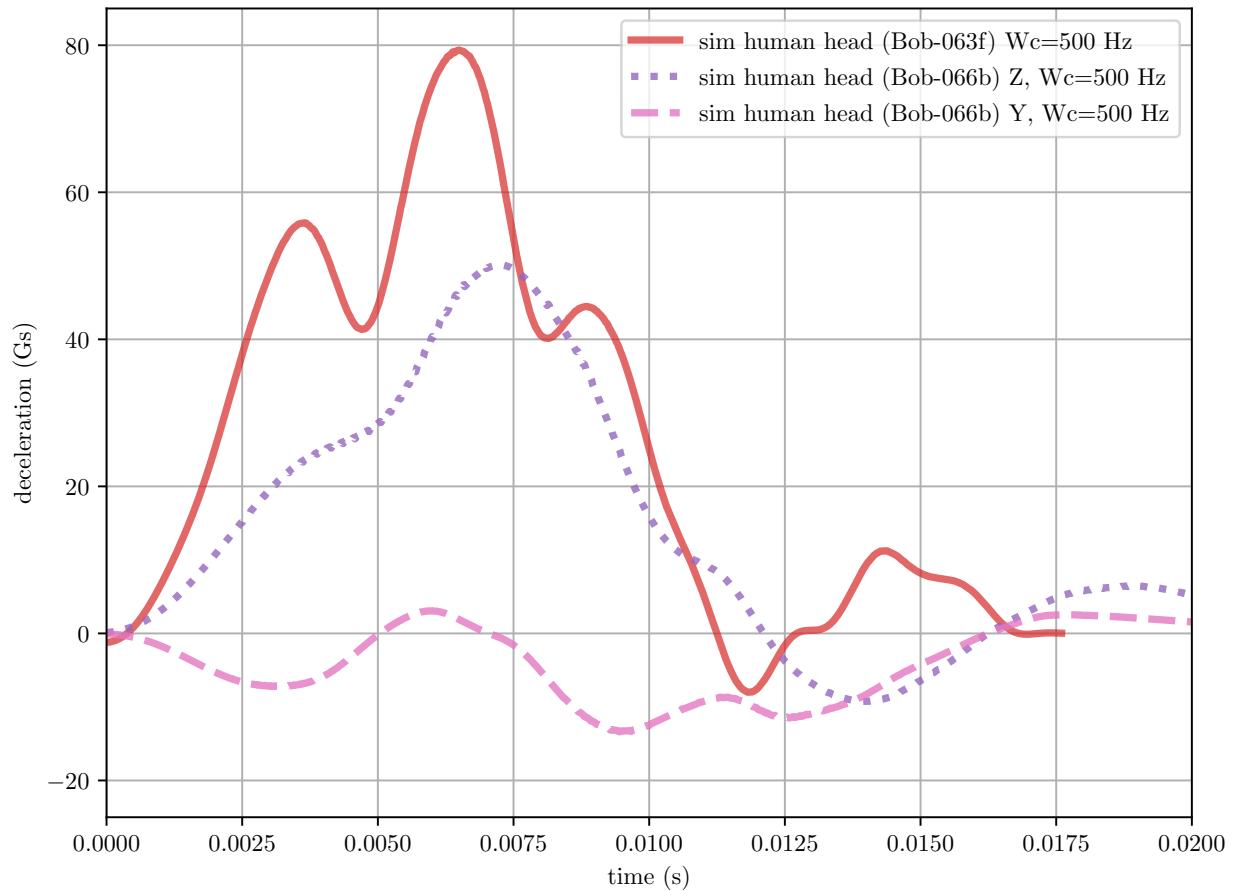


Figure 4-8. Comparison of deceleration pulses: human headform translation and rotation.

Figure 4-9 adds the acceleration traces of the skull, for comparison with the brain acceleration traces from Figure 4-8. The main result is that brain acceleration traces deviate from the skull acceleration traces at onset of skull rotation. After this deviation, brain accelerations appear as amplifications of the skull acceleration, indicating brain response is related, but not equivalent to, skull response in the presence of skull rotation.

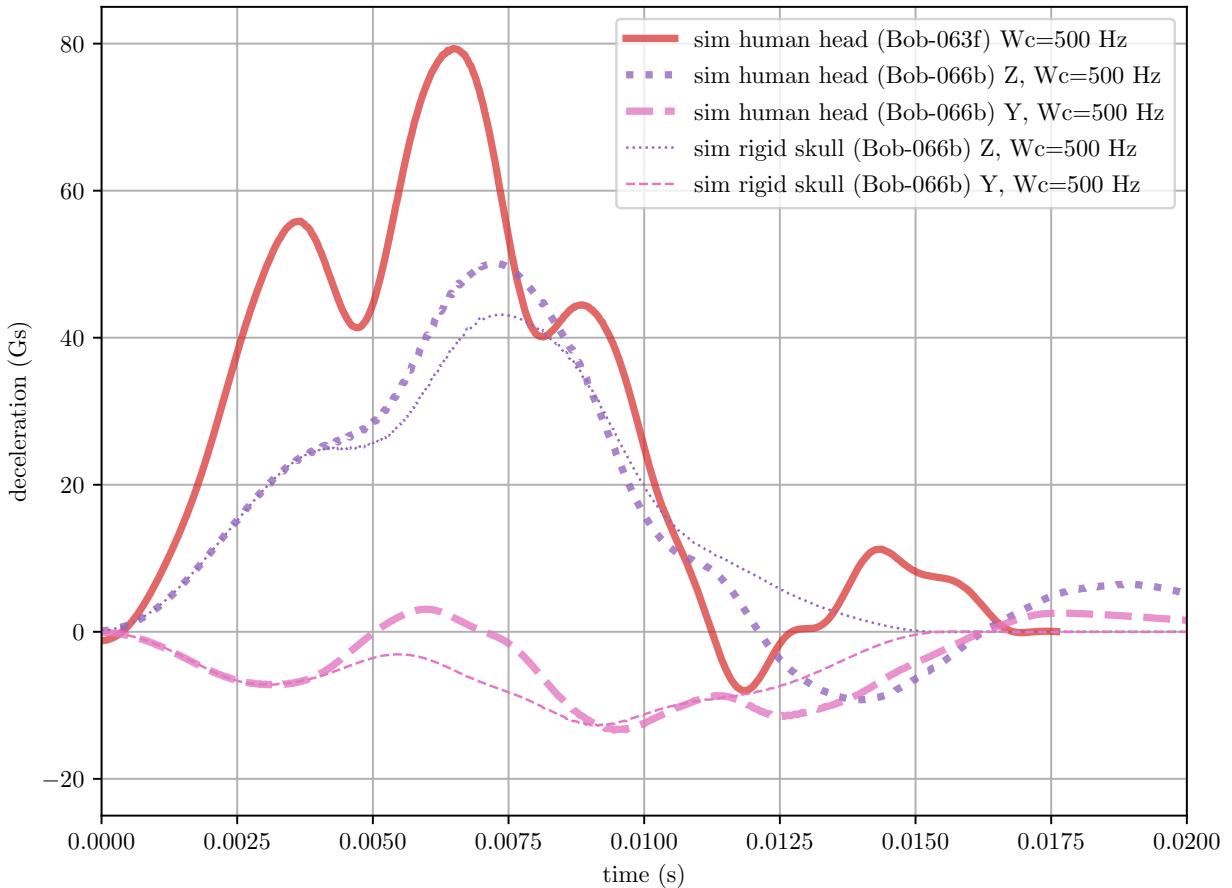


Figure 4-9. Comparison of deceleration pulses: human headform translation and rotation, with rigid skull boundary conditions.

For the modified mil-spec simulation, we elected to impose boundary conditions on the skull to match the experimental velocity time histories (Figure 4-11), obtained from numerical integration of the acceleration time history data (Figure 4-10).⁵

⁵These figures show the rotational time history plots. For the time history data, both rotational and translational, see the code repository [[Hovey, 2020](#)].

This approach allowed us to simplify the model and significantly decrease simulation wall clock time since the entire helmet assembly, as well as the skin and muscle materials, were suppressed. The modified mil-spec configuration also allowed for validation, above and beyond the already-conducted unit tests, of the three-points angular velocity (tpav) algorithm.⁶ Finally, this approach eliminated the “helmet rattle” phenomenon observed in the mil-spec simulation. The absence of a chin strap in the mil-spec simulation, a model simplification, allowed the helmet to move quickly relative to the head and anvil. In contrast, the mil-spec experiment tightly coupled the headform to the helmet, suppressing the “helmet rattle” response. This gap in model fidelity to the underlying physical system resulted in an inexact match of simulation acceleration to experiment acceleration (Figure 4-5). In contrast, with the approach adopted for the modified mil-spec configurations, the simulation and experiment accelerations matched identically, by design.

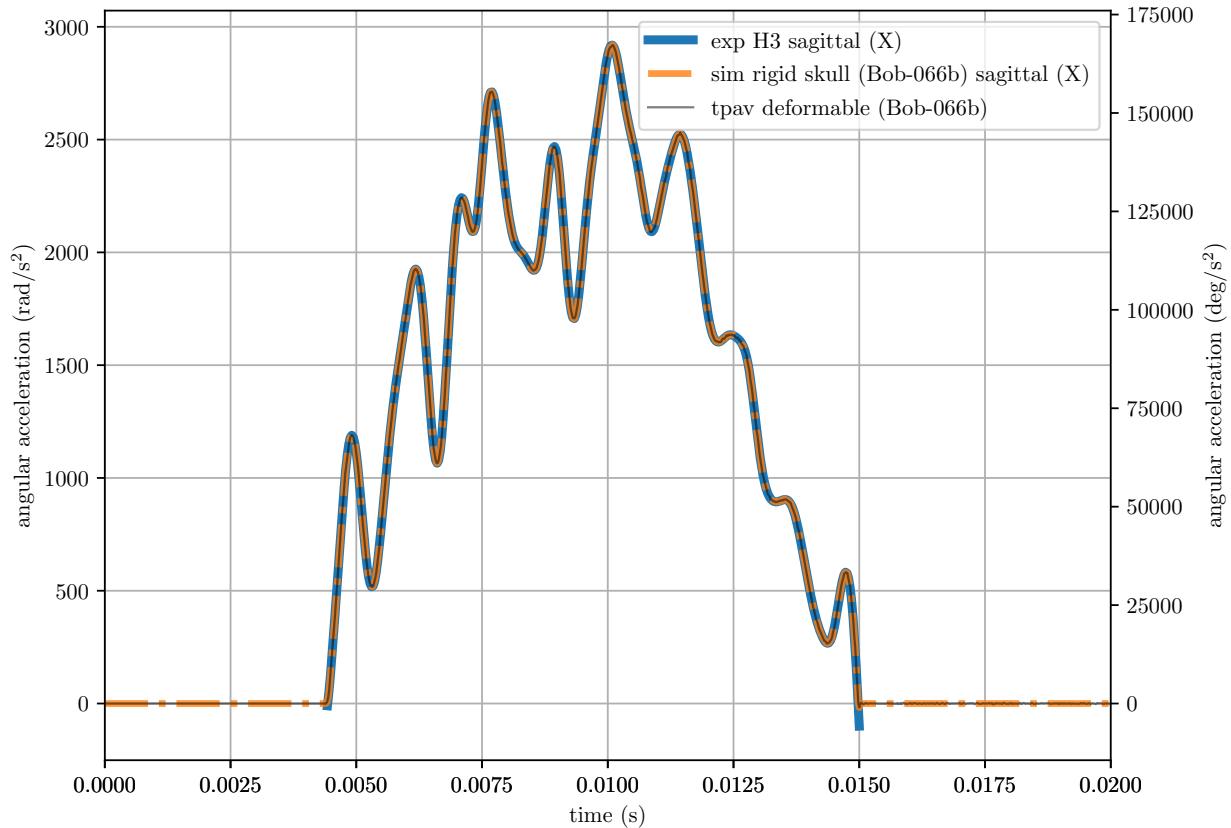


Figure 4-10. Comparison of angular acceleration curves: experiment versus simulation.

⁶See Section E.1 for details.

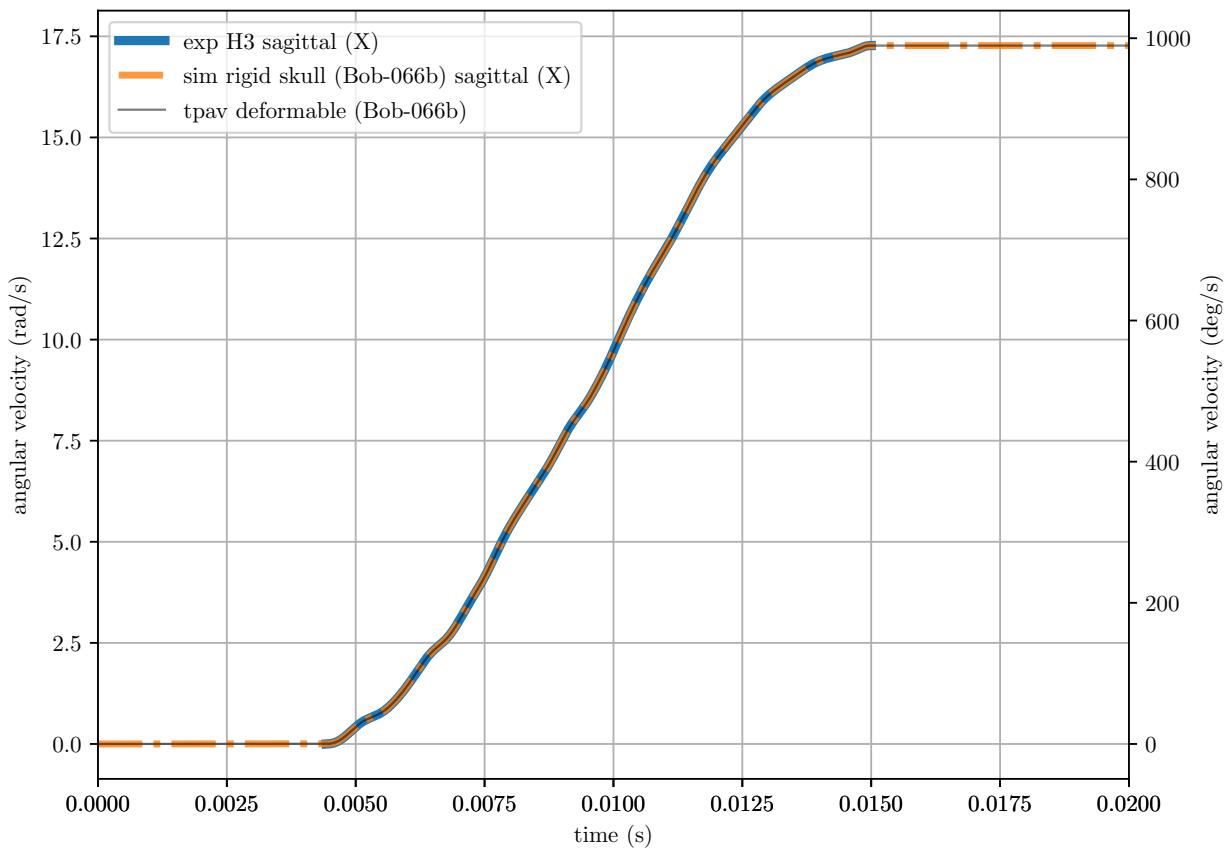


Figure 4-11. Comparison of angular velocity curves: experiment versus simulation.

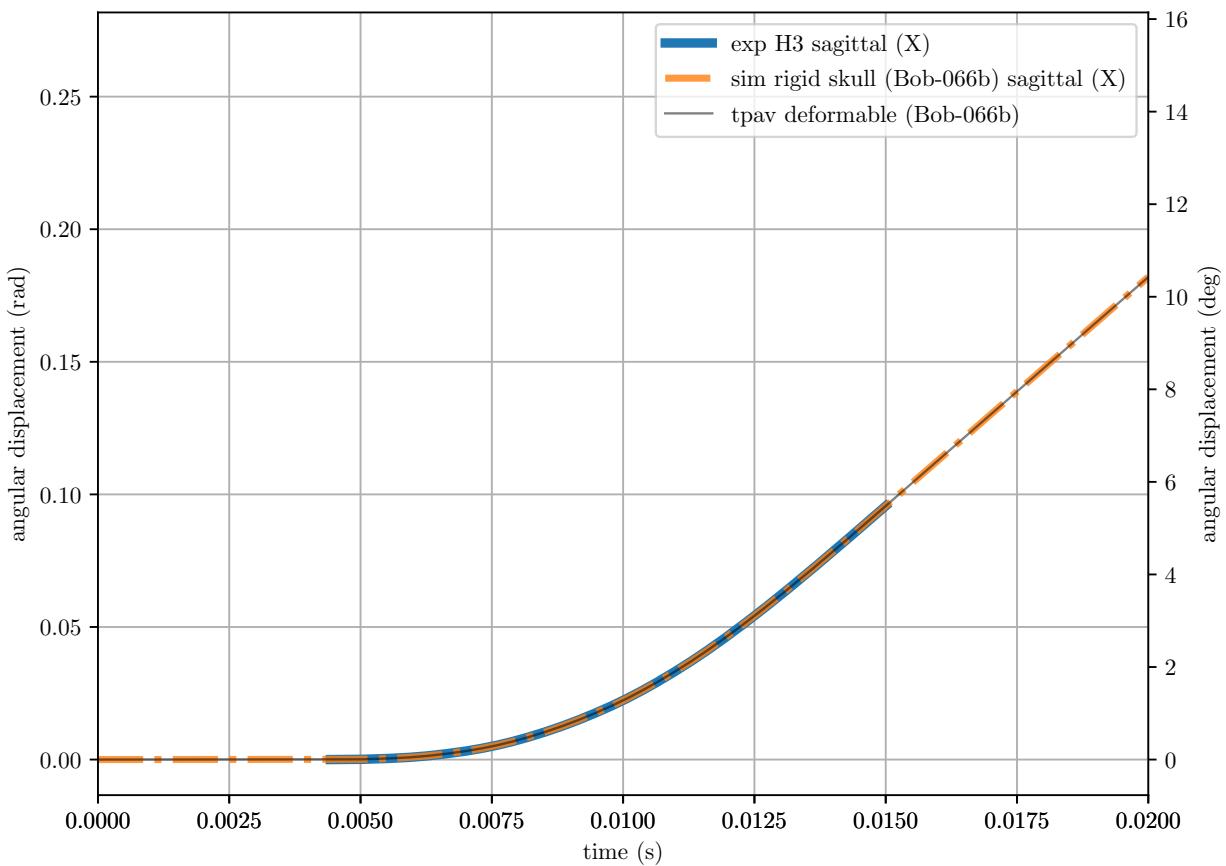


Figure 4-12. Comparison of angular displacement curves: experiment versus simulation.

Figures 4-13 and 4-14 track the 95th percentile of Green-Lagrange strain and Green-Lagrange strain rate, respectively, reported in the brain, over time.⁷

Thus, this metric is *not* defined at a tracer location; rather, it is a population metric, allowing the position at which the metric is reported to vary over time due to population dynamics. Importantly, data presented in Figure 4-14 should not be interpreted as a time derivative of data in Figure 4-13, despite the time-derivative relationships of the two dependent axes.

Figure 4-13 (respectively, Figure 4-14) can be interpreted as the 95th percentile maximum of 1,294,607 individual strain (respectively, strain rate) time histories for each finite element in the brain, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1).

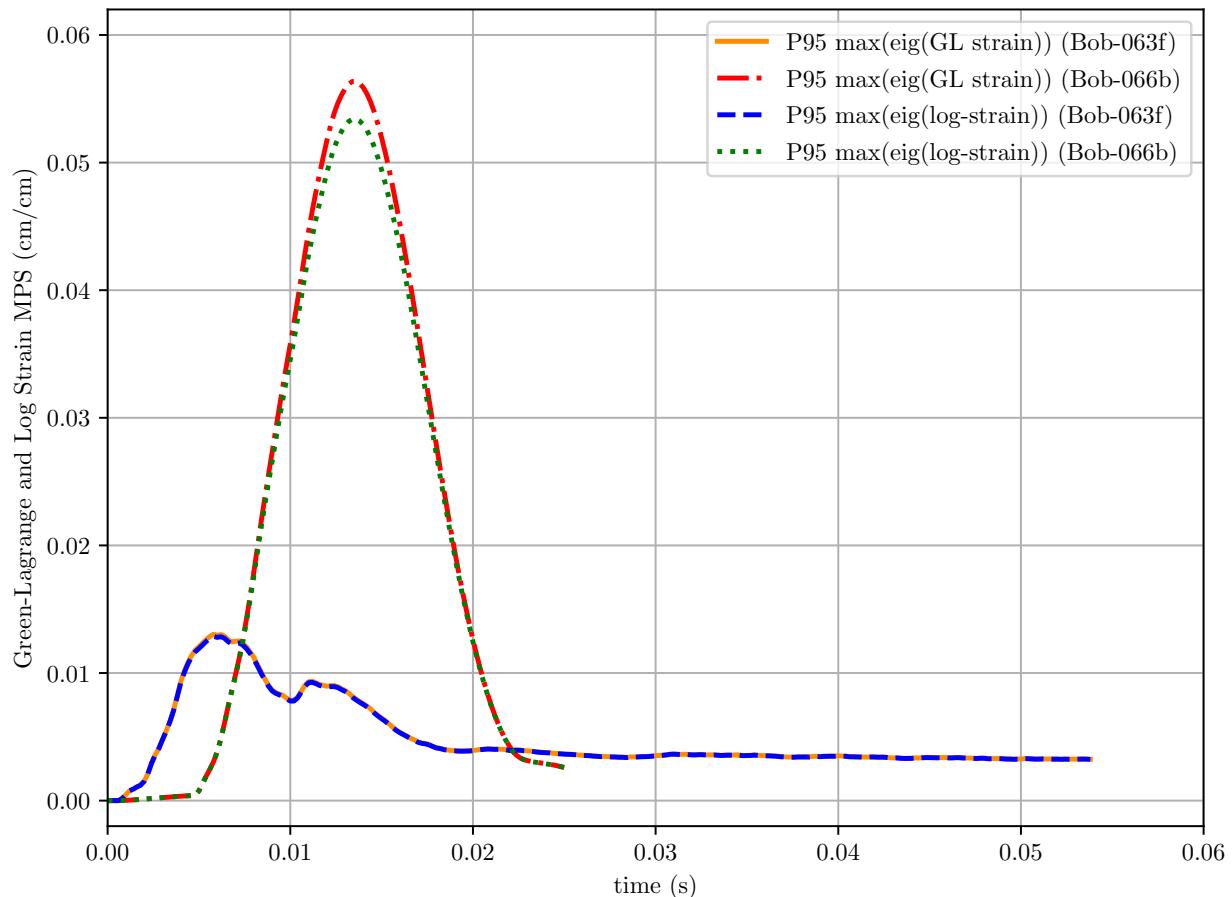


Figure 4-13. Maximum principal strain (MPS) of the Green-Lagrange strain and log strain in the human head simulations of the mil-spec (Bob-063f), and the modified mil-spec (Bob-066b).

⁷Log strain and rate of deformation were also reported for comparison with other strain metrics, as discussed in Sections 4.3 and E.4.

The mil-spec (Bob-063f) caused a translation-only post-impact response. The modified mil-spec (Bob-066b), however, caused a post-impact translation plus rotation response. The results allowed us to isolate a particular time during the simulation where strain (and possibly, strain rate) in the brain was most severe.⁸ Table 4-1 shows the two time points of interest for the mil-spec and modified mil-spec simulations.

Figures 4-18 through 4-22 show point clouds of each finite element in the brain's gray and white matter, at times where either peak strain or strain rate occur. Population 95th percentile values are shown for both the strain and strain rate metrics.

Figure 4-23 compares the peak strain results for the mil-spec simulation to the modified mil-spec simulation.

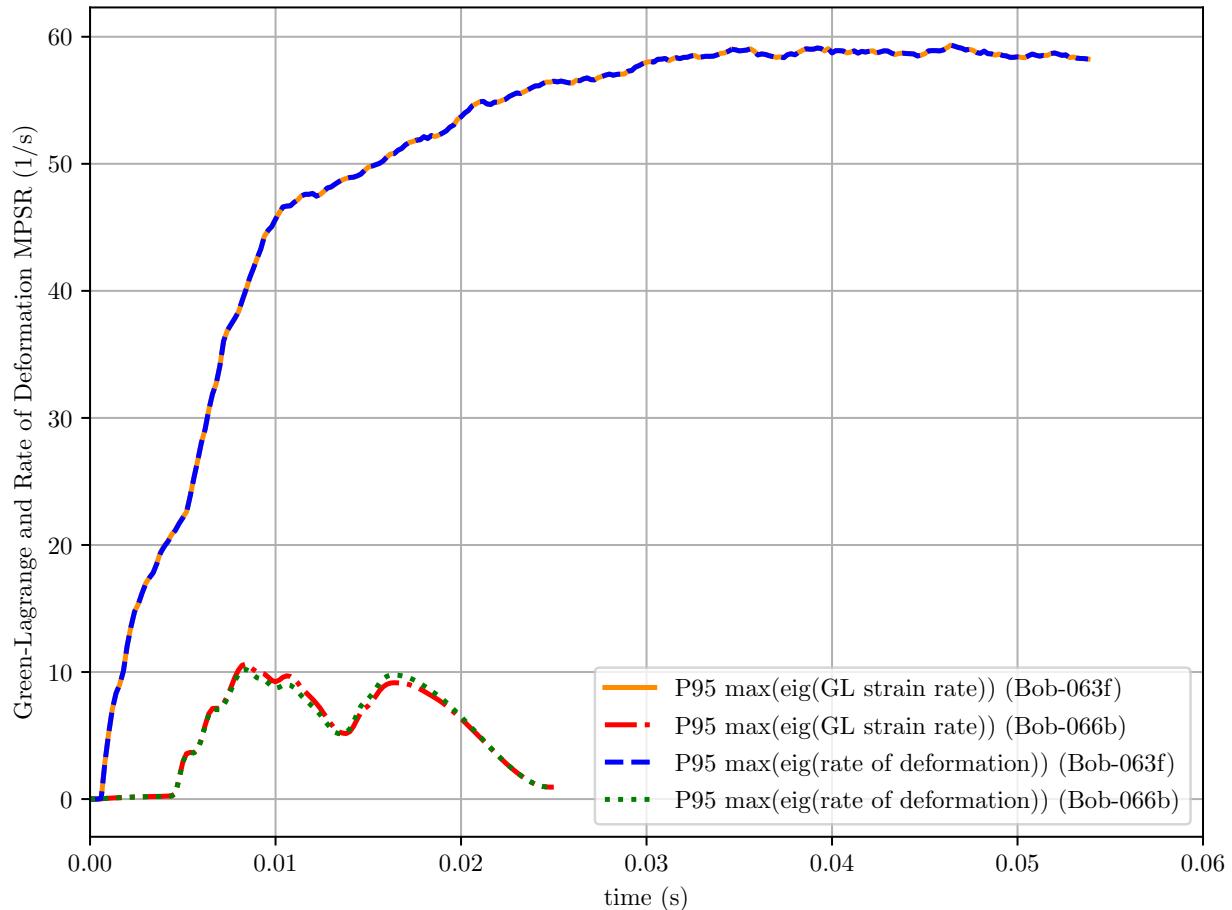


Figure 4-14. Maximum principal strain rate (MPSR) of the Green-Lagrange strain rate and rate of deformation in the human head simulations of the mil-spec (Bob-063f), and the modified mil-spec (Bob-066b).

⁸While we originally considered both strain and strain rate metrics, we later focused our efforts to strain alone. See Chapter 5 for a detailed discussion.

Table 4-1. Mil-spec and modified mil-spec simulations peak strain data and post-impact gross head motion description.

	Mil-Spec	Modified Mil-Spec
anvil	hemi-	45° flat
anvil configuration	Fig. 2-1	Fig. 2-2
simulation	Bob-063f	Bob-066b
max(eig(\mathbf{E})) (cm/cm)	1.30E-2	5.64E-2
at time (ms)	5.800	13.600
at time step (int)	30	69
post-impact head motion	—	—
translation	yes	yes
rotation	no	yes

4.3. Injury Risk Curve

Figure 4-15 shows the cellular-based injury risk curve newly developed in [Summey, 2020d].⁹ In that work, a neuronal cell culture embedded in a collagen hydrogel was subjected to a simple shear deformation. The bulk shear deformation of the hydrogel caused individual neurons embedded therein to undergo local deformation. This local deformation was characterized along cell neurites, with a local tangent, normal, and bi-normal (Frenet-Serret) basis.

Local mean tension, compression, and shear metrics were developed.¹⁰ The nomenclature $\langle E_t \rangle$ and $\langle e_t \rangle$ represented *tensile* strain along the neurite long axis, measured by the Green-Lagrange and Almansi-Euler tensors, respectively. Similar metrics for compression (*i.e.*, $\langle E_c \rangle$ and $\langle e_c \rangle$) and shear (*i.e.*, $\langle E_s \rangle$ and $\langle e_s \rangle$) were developed but not used for the injury risk curve because experiments showed cell death was correlated to local tension, but not local compression or local shear [Summey, 2020b]. Between the two strain tensors, the Almansi-Euler strain and its strain rate provided stronger correlation.¹¹

A nominal threshold delineating predictions of cell injury from cell non-injury was estimated as

$$\langle e_t \rangle (\dot{e}_t) = 0.128 \langle \dot{e}_t \rangle^{-0.156}, \quad (4.1)$$

based on the live cells and dead cells data shown in Figure 4-15. Per [Summey, 2020c], several caveats underpin this result:

1. It remains unclear how to transfer the injury risk threshold, based on outcomes for individual cells, into finite element results. In our finite element mesh, each brain element consisted of an 8-noded hexagonal finite element with side length of 1-mm in

⁹[Summey, 2020a] later updated this figure, reproduced herein as Figure 4-16 and Figure 4-17.

¹⁰See [Summey, 2020d], at 12–13 for details.

¹¹Summey, *op. cit.*, at 45.

each of the direction, X , Y , and Z . Per [Summey, 2020c], each finite element of 1-mm³ volume would contain between 2,500 and 5,000 cells. For now, we resolve to present the maximum principal strain and strain rates only as a *proxy* for cell death, with the limitation that our strain and strain rate metrics, pointwise at finite element Gauss points, do not account for viability or non-viability of cell populations.¹²

2. The experimental results are preliminary. Additional experimental data are required to provide a statistical analysis and confidence intervals to improve the threshold estimate.
3. The experimental results are binary: injured and non-injured. Moreover, the classification of injury is equated with death, the most extreme endpoint in the injury continuum. In contrast, extant injury risk curves provide estimates of injury *severity*, in a non-binary continuum, from minimally injured to maximally injured. Additional experimental data are required to provide an injury-continuum risk prediction.

¹²We remark that during the course of our joint investigations, our experimental partners first started with the Green-Lagrange strain tensors, since it presented the experimentalist with lengths in the undeformed configuration, relatively easier to measure than in the current configuration. Over the course of the experimental development, our partners moved to the Almansi-Euler strain. Unfortunately, our simulations, already completed, included only quantities for Green-Lagrange, not Almansi-Euler. We were unable to rerun our simulations due to time and budgetary constraints. Nonetheless, it can be shown that for an Almansi-Euler strain on the order of 10%, the approximate prevailing strain magnitude found in the data, the Green-Lagrange strain at the same stretch ratio is only approximately 13%. We note this discrepancy as current limitation and an opportunity for future refinement. See Section E.4.1 for more details.

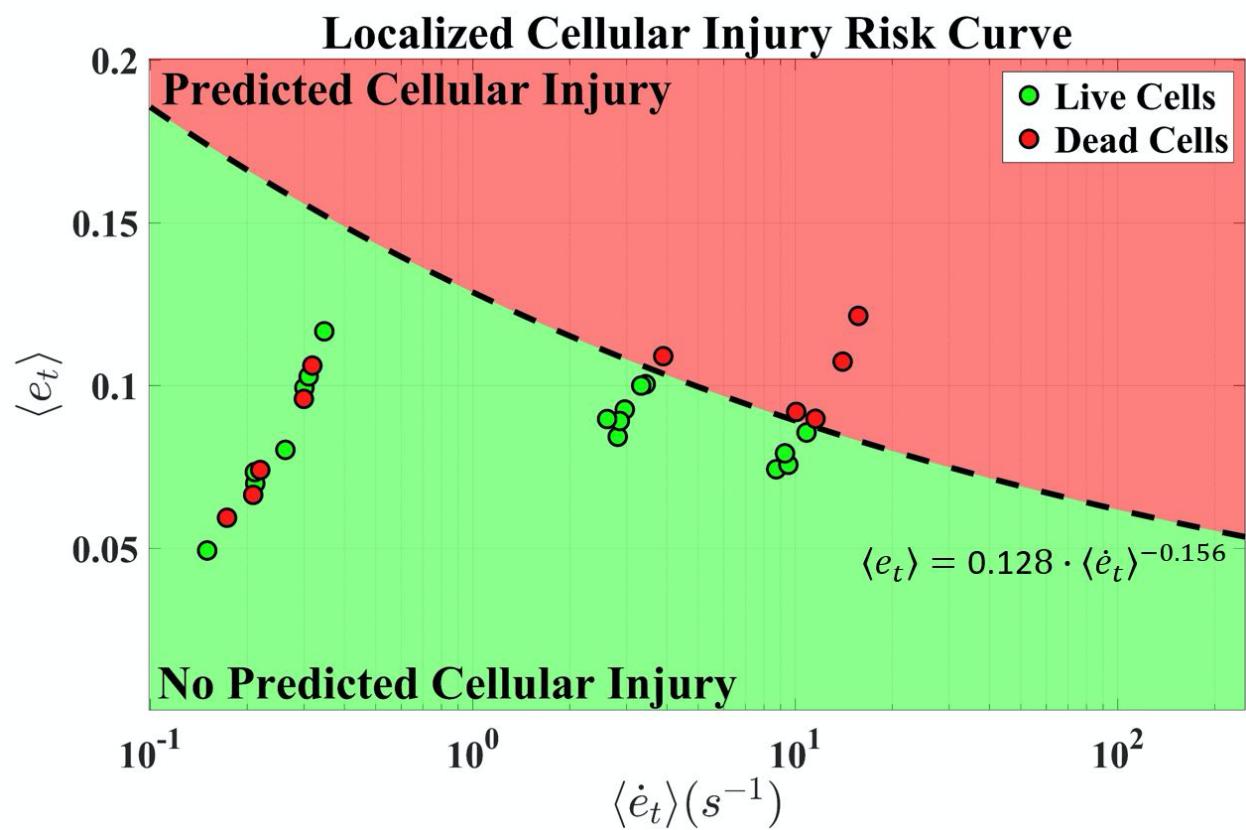


Figure 4-15. Reproduction of strain versus strain rate cellular injury risk curve of [\[Summey, 2020d\]](#), Figure 5.8.

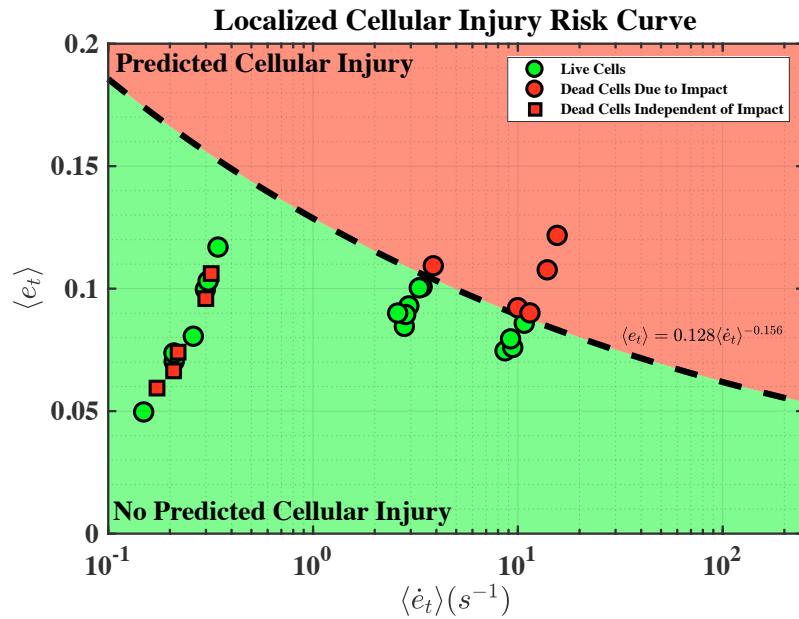


Figure 4-16. Update of the Summey thesis figure.

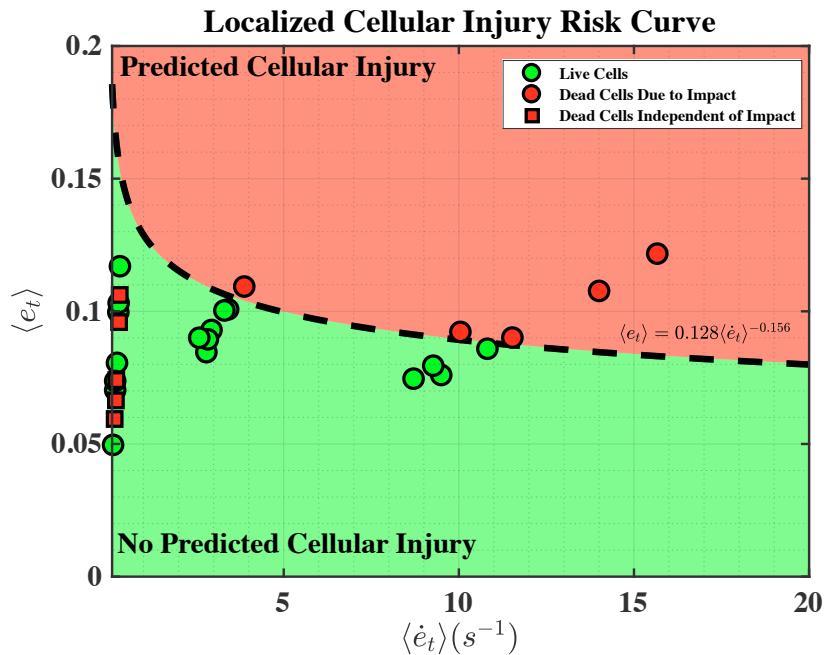


Figure 4-17. Update of the Summey thesis figure, linear strain rate scale.

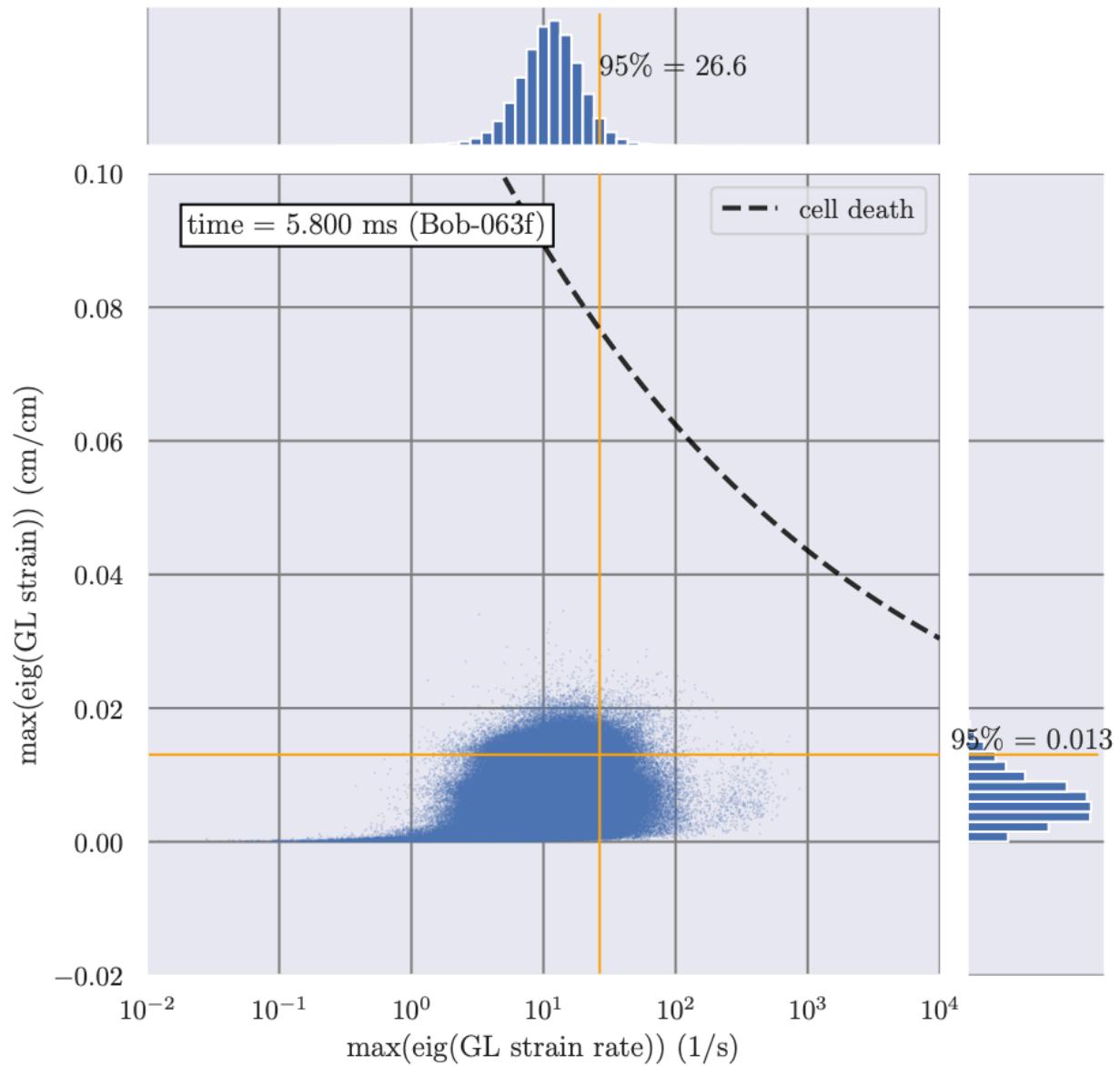


Figure 4-18. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for the mil-spec simulation (Bob-063f) at 5.8 ms.

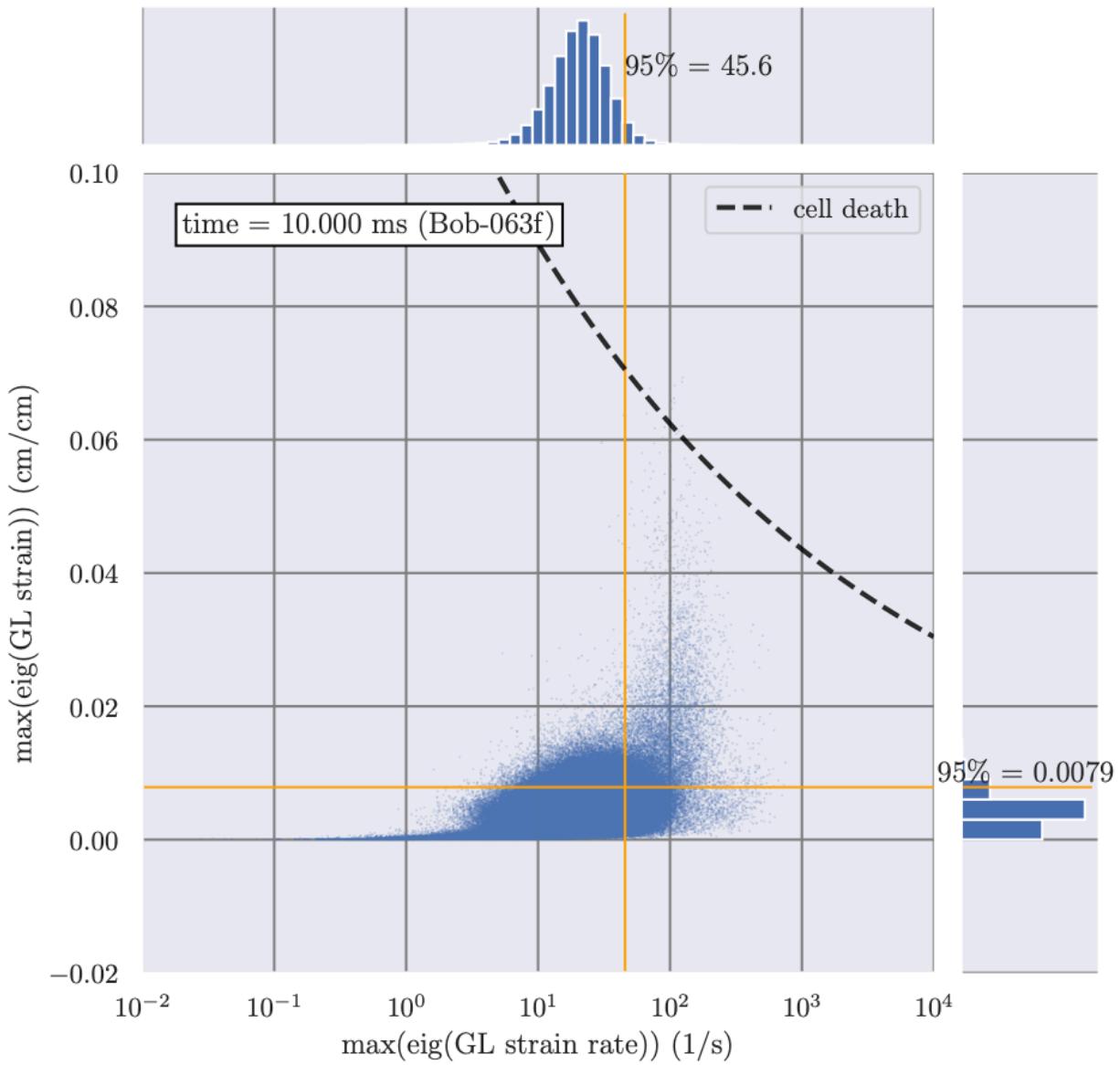


Figure 4-19. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for the mil-spec simulation (Bob-063f) at 10.0 ms.

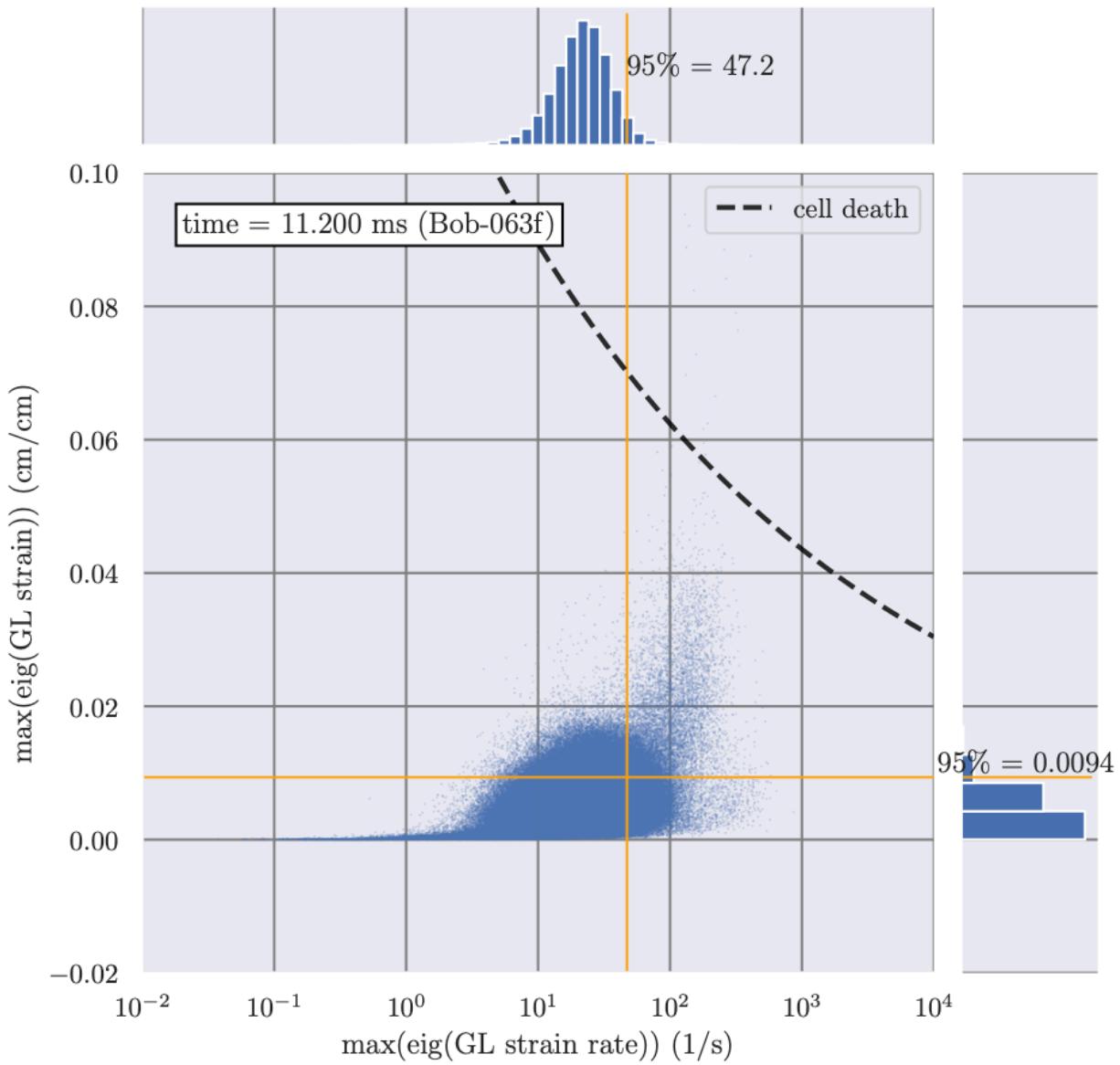


Figure 4-20. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for the mil-spec simulation (Bob-063f) at 11.2 ms.

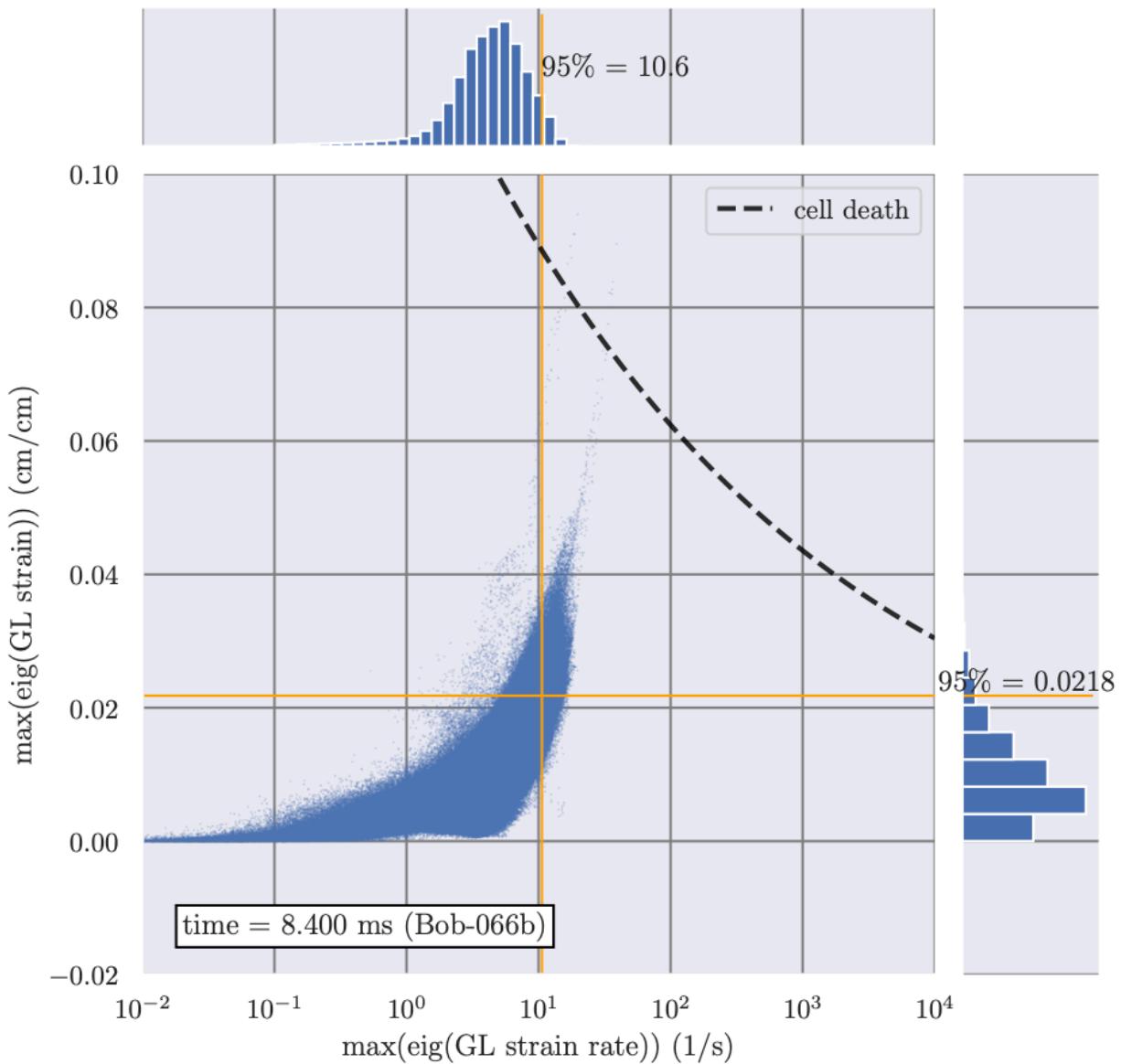


Figure 4-21. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for modified mil-spec simulation (Bob-066b) at 8.4 ms.

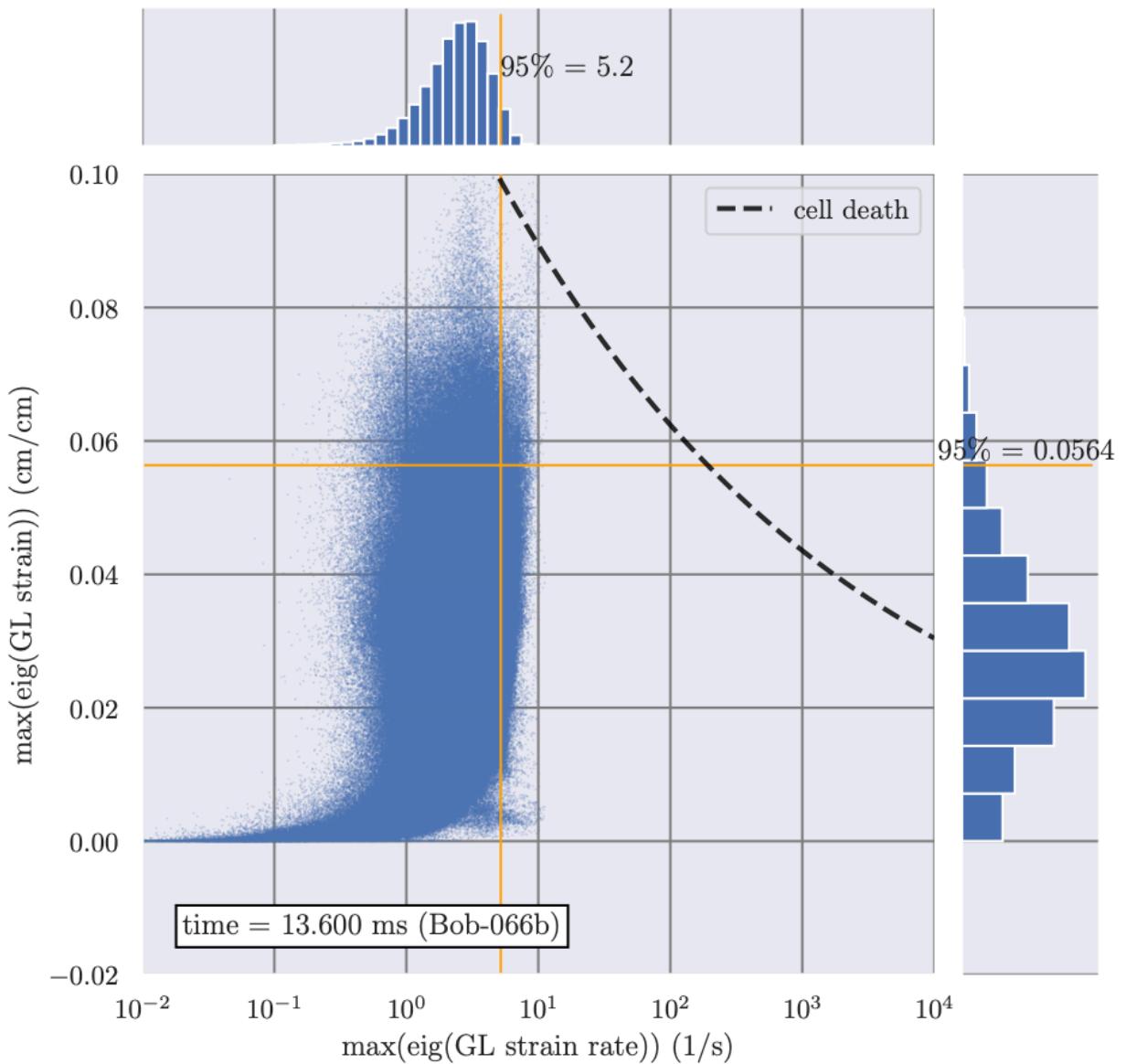


Figure 4-22. Strain and strain rate of 1,294,607 brain finite elements, composed of 504,505 white matter finite elements and 790,102 gray matter finite elements (see Table C-1) for modified mil-spec helmeted simulation (Bob-066b) at 13.6 ms.

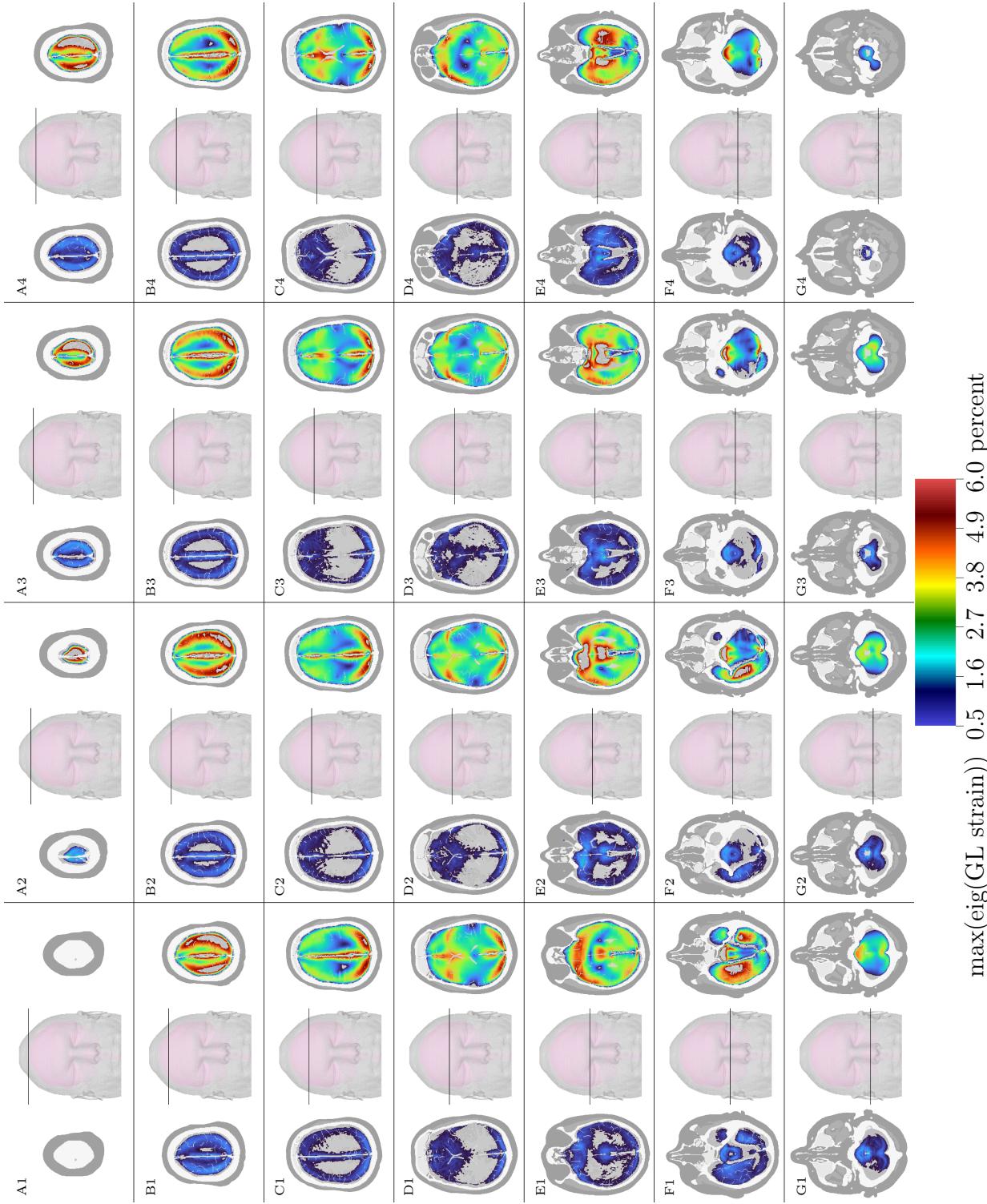


Figure 4-23. Sequence of axial sections, showing the maximum eigenvalue of Green-Lagrange strain for the mil-spec and modified mil-spec cases, at respective times indicated in Table 4-1.

5. DISCUSSION

Reviewing Figure 4-2, we see the comparison of the acceleration time history for the experimental DOT into MEP impact with the simulation of the same configuration showed excellent agreement, as quantified by the L2-norm error rate of less than 0.2. Acceleration rise time, peak, and duration all matched well, indicating framework validation, as defined in Section 1.2.

Several iterations of input decks were used to obtain this match. A simple objective function composed, defined as the L2-norm error rate as a function of elastic modulus (E) and Poisson ratio (ν) parameter space was constructed.¹ The steepest descent gradient was determined, which gave rise to successive (E, ν) candidate pairs, and was used iteratively until the error norm was sufficiently small.

With the simulation parameters matched to experimental results, we then performed a sole substitution of the DOT with a Mg Bob headform. The purpose was to investigate the effect of changing the geometry of the headform impactor, all else held equal, including bullet mass.

Figure 4-3 shows the Mg Bob headform to produce an acceleration profile that has a decrease in rise time, peak and an increase in pulse duration. The Mg Bob headform generally is larger than the DOT headform. As a result, the Mg Bob headform impact recruited a larger amount of the MEP surface and material, lessening the impact severity.

Next, we retained the Bob geometry, but replaced the Mg singleton material with the multi-materials of human Bob. The two acceleration traces, shown in Figure 4-3, indicate that the material effects further work to reduce acceleration rise time, peak, and extend the pulse duration. This result was expected, as the soft, viscoelastic materials of human tissue were prognosticated to have more dissipative effects than magnesium, an elastic, non-viscous material.

Comparing Figures 4-2 with 4-3, we see geometric effects are more profound than material effects, though both cause the same trend over the baseline.

In general, all unhelmeted configurations had characteristic peak values of 150 to 250 Gs, and pulse durations of 5 to 7 ms. The gross metrics stand in stark contrast to the helmeted configuration cases, which had significantly lower peaks, and higher durations.

Experimental helmeted DOT impacts typically demonstrated peak acceleration levels in the 40 to 60 G corridor, with pulse durations of about 15 ms.

¹The MEP was modeled as a finite deformation elastic material. See Section D.1.1 for details.

Figure 4-4 shows four experimental impacts, with their average. A characteristic double-peak, with the amplitude of the second exceeding that of the first, is seen. The first peak is understood as the initial arrest of the impacting mass, particularly of the helmet system, as the helmet foam works to absorb the impact energy. The dip between the peaks is understood as the helmet rebound away from the target back toward the headform. Finally, the second peak occurs as the inbound headform pushes back on the rebounded helmet, and compresses the helmet foam, until the internal forces balance the inertial forces, and full scale rebound of the headform occurs. A two-degree of freedom oscillator is a sufficient surrogate to give explain this behavior. The component (*e.g.*, helmet) proximate to the impact surface has a higher stiffness and lower mass than its lagging counterpart (*e.g.*, the head). The helmet's oscillatory behavior thus has a higher frequency than the that of the head, giving rise the helmet rebounding back into the head prior to the arrival of the head pushing the system fully into the target.

Compared to the experimental average (see Figure 4-5), our simulation predicted three main peaks, rather than two. While the gross trends were the similar, the relatively higher frequency present in the simulations arose from the so-called "helmet rattle" effect, caused in the simulations by the lack of a chin strap to attach the helmet to the head. In contrast, the experiments tightly coupled the DOT headform to the helmet and were absent of "helmet rattle" behavior. When investigative experiments were run without the tight coupling, they also demonstrated the "helmet rattle" effect, confirming this as the high frequency source found in the simulations.

Sole substitution of the Mg Bob headform for the DOT headform demonstrate the enlarged geometry of Bob's head shape cause distension of the pulse, as seen previously in the unhelmeted case. For the Mg Bob headform, the first, second, and fourth peaks are mildly attenuated, but the second peak is mildly amplified, relative to the DOT headform baseline (Figure 4-6).

As with the unhelmeted case, the helmeted case of Bob with human materials relative to Mg-only materials caused mild attenuation of the response signal, likely caused by the dissipative gray and white matter of the brain (Figure 4-7).

The head response from the modified mil-spec (Figure 4-8) shows a decrease in acceleration rise time and peak, with a slight increase in pulse duration. This result is expected, as not all of the inbound translational kinetic energy gets rebound as translational kinetic energy, as in the mil-spec case. Rather, the inbound kinetic energy, all from translational kinetic energy, gets split, post-collision, partly into rebounded translational kinetic energy and partly into *de novo* rotational kinetic energy.

While it may be tempting to therefore suggest the modified mil-spec, relative to the mil-spec case, is less injurious by virtue of its milder acceleration time history, exactly the opposite will be found to be true.

To address this finding, consider the strain response shown in Figure 4-13. The maximum tissue strain is on the order of 5 to 6 percent for the modified mil-spec case, compared to just over 1 percent for the mil-spec case. This result, explained in broad terms in Section 1.3, illustrates the brain's propensity to undergo deformation through shear rather

than volume changes. It also alerts us to the notion than an acceleration metric alone for the mil-spec may be insufficient as a proxy for injury risk potential.

Point clouds of strain and strain rate for every finite element of the gray and white matter are presented in Figures 4-18 through 4-22. Throughout these figures, one can observe that we are more likely to approach the cell death threshold by way of excessive strain than by way of excessive strain rate. This observation becomes even more profound once the linear scales for strain versus the log scales for strain rate are noted. That is, the strain rate can continue to increase by several factors of 10 before the injury risk curve is approached, given the injury threshold in its current formulation. The same cannot be said for increases in strain, given an expected strain rate.

We have chosen the 95th percentile value of strain and strain rate as cutoff values, in the hopes of disregarding outliers, and being conservative in our estimate of our reporting of maximum values.

Figure 4-22 shows axial sections of brain strain, comparing the maximum values observed in the mil-spec case with the modified mil-spec case. Generally, the modified mil-spec case shows elevated strain levels throughout the brain. High values of brain strain appear near the periphery of the brain, where the gray matter sits adjacent to the cerebral spinal fluid and the interior of the skull. Elevated gray matter strain is also observed near the interface with the falx. Less detectable is any effect from the gray-white interface. This result indicates that shear constitutive disparities between gray matter and skull largely saturate any relatively small disparities between gray and white matter.

6. CONCLUSIONS

We have demonstrated that our high-fidelity simulation framework can reliably reproduce experimental data of unhelmeted and helmeted impacts. We have shown that head impacts generating rotations are more deleterious to the brain than impacts causing translations alone, all else being equal.

The basis for this conclusion is the cellular-based injury criterion that accounts for brain strain and strain rate. Underpinning this criterion is the fundamental material property that the brain, composed primarily of water, strongly prefers isochoric to volumetric deformation. It is difficult to compress water sitting in a glass. It is easy, however, to stir it.

Brain deformation, and not acceleration alone, should be used as an explanatory variable for brain injury risk potential. Indeed, as shown by our comparison of the mil-spec to the modified mil-spec configurations, relatively lower accelerations may lead to relatively higher deformations, illustrating the need to distinguish between translational and rotational variables.

The final, and perhaps most actionable, conclusion from this work is our finding that the current mil-spec deserves a renewed evaluation for its relevance to risk assessment of head injury, secondary to blunt impact.

Our investigation exposes a potentially severe shortcoming of the current mil-spec: It may actually be an assessment more of helmet integrity than of brain injury.

Without a rotational component, the current mil-spec is incapable of fully predicting brain injury. The current mil-spec should be modified to include a rotational component for updated relevance to prediction and assessment of brain injury risk.

REFERENCES

[Ackerman, 1998] Ackerman, M. J. (1998). The visible human project. *Proceedings of the IEEE*, 86(3):504–511.
https://www.nlm.nih.gov/research/visible/visible_human.html.

[Brannon, 1998] Brannon, R. (1998). Caveats concerning conjugate stress and strain measures for frame indifferent anisotropic elasticity. *Acta Mechanica*, 129(1-2):107–116.

[Brundage, 2014] Brundage, A. (2014). Prediction of shock-induced cavitation in water. In *Journal of Physics: Conference Series*, volume 500, page 102002. IOP Publishing.
<https://doi:10.1088/1742-6596/500/10/102002>.

[Brundage, 2013] Brundage, A. L. (2013). Implementation of tillotson equation of state for hypervelocity impact of metals, geologic materials, and liquids. *Procedia Engineering*, 58:461–470. <https://doi.org/10.1016/j.proeng.2013.05.053>.

[Childs et al., 2012] Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K., Miller, M., Harrison, C., Weber, G. H., Krishnan, H., Fogal, T., Sanderson, A., Garth, C., Bethel, E. W., Camp, D., Rübel, O., Durant, M., Favre, J. M., and Navrátil, P. (2012). VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, pages 357–372. University of California.
"<https://escholarship.org/content/qt69r5m58v/qt69r5m58v.pdf>" and
<https://wci.llnl.gov/simulation/computer-codes/visit/>".

[Feng et al., 2010] Feng, Y., Abney, T. M., Okamoto, R. J., Pless, R. B., Genin, G. M., and Bayly, P. V. (2010). Relative brain displacement and deformation during constrained mild frontal head impact. *Journal of the Royal Society Interface*, 7(53):1677–1688.

[Giudice et al., 2019] Giudice, J. S., Zeng, W., Wu, T., Alshareef, A., Shedd, D. F., and Panzer, M. B. (2019). An analytical review of the numerical methods used for finite element modeling of traumatic brain injury. *Annals of Biomedical Engineering*, 47(9):1855–1872.

[Haniff and Taylor, 2017] Haniff, S. and Taylor, P. A. (2017). In silico investigation of blast-induced intracranial fluid cavitation as it potentially leads to traumatic brain injury. *Shock Waves*, 27(6):929–945.
<https://www.osti.gov/pages/servlets/purl/1421630>.

[Hertel et al., 1995] Hertel, E., Bell, R., Elrick, M., Farnsworth, A., Kerley, G., McGlaun, J., Petney, S., Silling, S., Taylor, P., and Yarrington, L. (1995). Cth: A software family

for multi-dimensional shock physics analysis. In *Shock Waves Marseille I*, pages 377–382. Springer. <https://www.osti.gov/servlets/purl/10180820>.

[Hovey, 2020] Hovey, C. (2020). Sandia Injury Biomechanics Laboratory (SIBL). <https://github.com/sandialabs/sibl>.

[Jog and Motamarri, 2009] Jog, C. and Motamarri, P. (2009). An energy-momentum conserving algorithm for nonlinear transient analysis within the framework of hybrid elements. *Journal of Mechanics of Materials and Structures*, 4(1):157–186.

[Merewether et al., 2020] Merewether, M. T., Plews, J. A., de Frias, G. J., Mosby, M. D., Porter, V. L., Shelton, T., Thomas, J. D., Tupek, M. R., Veilleux, M., Manktelow, K., et al. (2020). Sierra/solidmechanics 4.56 user’s guide. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States). <https://www.osti.gov/servlets/purl/1608404>.

[Merkle, 2011] Merkle, A. (2011). Personal communication.

[Moss and King, 2011] Moss, W. C. and King, M. J. (2011). Impact response of us army and national football league helmet pad systems. Technical report, Lawrence Livermore National Lab CA. <https://apps.dtic.mil/dtic/tr/fulltext/u2/a536266.pdf>.

[Nahum et al., 1977] Nahum, A. M., Smith, R., and Ward, C. C. (1977). Intracranial pressure dynamics during head impact. Technical report, SAE Technical Paper.

[Neff et al., 2016] Neff, P., Eidel, B., and Martin, R. J. (2016). Geometry of logarithmic strain measures in solid mechanics. *Archive for Rational Mechanics and Analysis*, 222(2):507–572. <https://doi.org/10.1007/s00205-016-1007-x>.

[Owen et al., 2014] Owen, S., Staten, M., and Sorensen, M. (2014). Parallel hex meshing from volume fractions. *Engineering with Computers*, 30. <https://doi.org/10.1007/s00366-012-0292-8>.

[Owen et al., 2019] Owen, S. J., Ernst, C. D., and Stimpson, C. J. (2019). Sculpt version 15.4: Automatic parallel hexahedral mesh generation. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States). <https://cubit.sandia.gov/public/documents/SculptSANDReport-5.pdf>.

[Owen et al., 2011] Owen, S. J., Staten, M. L., and Sorensen, M. C. (2011). Parallel hex meshing from volume fractions. In *Proceedings of the 20th International Meshing Roundtable*, pages 161–178. Springer.

[Pierson, 2018] Pierson, K. (2018). Private Communication. Electronic mail correspondence 06 December 6 2018, 22:38 MDT.

[Rivlin, 1956] Rivlin, R. (1956). Solution of some problems in the exact theory of visco-elasticity. *Journal of Rational Mechanics and Analysis*, 5(1):179–188.

[Simo and Hughes, 2006] Simo, J. C. and Hughes, T. J. (2006). *Computational Inelasticity*, volume 7. Springer Science & Business Media.

[Summey, 2020a] Summey, L. A. (2020a). Private Communication. Electronic mail correspondence 19 June 2020.

[Summey, 2020b] Summey, L. A. (2020b). Private Communication. PANTHER meeting 27 May 2020.

[Summey, 2020c] Summey, L. A. (2020c). Private Communication. Electronic mail correspondence 18 June 2020.

[Summey, 2020d] Summey, L. A. (2020d). Development of a simple shear impact device for evaluating cellular traumatic brain injury. Master's thesis, University of Wisconsin–Madison, Madison, Wisconsin.

[Swanson, 1985] Swanson, S. (1985). A constitutive model for high elongation elastic materials. *Journal of Engineering and Technology*, 102(2):110–115.

[Swanson et al., 1985] Swanson, S., Christensen, L., and Ensign, M. (1985). Large deformation finite element calculations for slightly compressible hyperelastic materials. *Computers & Structures*, 21(1-2):81–88.

[Taylor and Ford, 2009] Taylor, P. A. and Ford, C. C. (2009). Simulation of blast-induced early-time intracranial wave physics leading to traumatic brain injury. *Journal of Biomechanical Engineering*, 131(6).
https://www.sandia.gov/biomechanics/_assets/documents/Taylor_2009_001.pdf.

[Taylor et al., 2014a] Taylor, P. A., Ludwigsen, J. S., and Ford, C. C. (2014a). Investigation of blast-induced traumatic brain injury. *Brain Injury*, 28(7):879–895.
<https://doi.org/10.3109/02699052.2014.888478>.

[Taylor et al., 2018] Taylor, P. A., Ludwigsen, J. S., Ford, C. C., and Vakhtin, A. A. (2018). Verification and validation of simulation framework for analysis of traumatic brain injury. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States). <https://www.osti.gov/servlets/purl/1529058>.

[Taylor et al., 2014b] Taylor, P. A., Ludwigsen, J. S., Vakhtin, A. A., and Ford, C. C. (2014b). Simulation and clinical assessment of blast-induced traumatic brain injury. *IBIA Neurotrauma Lett.*, accepted.

[Tillotson, 1962] Tillotson, J. H. (1962). Metallic equations of state for hypervelocity impact. Technical report, General Dynamics San Diego CA General Atomic Division.
<https://apps.dtic.mil/dtic/tr/fulltext/u2/486711.pdf>.

[Vakhtin et al., 2013] Vakhtin, A. A., Calhoun, V. D., Jung, R. E., Prestopnik, J. L., Taylor, P. A., and Ford, C. C. (2013). Changes in intrinsic functional brain networks following blast-induced mild traumatic brain injury. *Brain Injury*, 27(11):1304–1310.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5075489/>.

[Van Rossum et al., 2007] Van Rossum, G. et al. (2007). Python programming language. In *USENIX Annual Technical Conference*, volume 41, page 36.
<https://www.python.org/>.

[Veilleux et al., 2019] Veilleux, M., Beckwith, F., Belcourt, K. N., de Frias, G. J., Manktelow, K., Merewether, M. T., Miller, S. T., Mosby, M. D., Plews, J. A., Porter, V. L., et al. (2019). Sierra/solidmechanics 4.54 user's guide: Addendum for shock capabilities. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States). <https://www.osti.gov/servlets/purl/1569142>.

[Xiao et al., 1997] Xiao, H., Bruhns, O., and Meyers, A. (1997). Hypo-elasticity model based upon the logarithmic stress rate. *Journal of Elasticity*, 47(1):51–68.

[Yang et al., 2016] Yang, H., Jekir, M. G., Davis, M. W., and Keaveny, T. M. (2016). Effective modulus of the human intervertebral disc and its effect on vertebral bone stress. *Journal of Biomechanics*, 49(7):1134–1140.
<https://doi.org/10.1016/j.jbiomech.2016.02.045>.

[Zhang et al., 2001] Zhang, L., Yang, K. H., and King, A. I. (2001). Comparison of brain responses between frontal and lateral impacts by finite element modeling. *Journal of Neurotrauma*, 18(1):21–30. <https://doi.org/10.1089/089771501750055749>.

APPENDIX A. Overview

The Appendices contain our implementation details. We have included these details to support *reproducibility* of our results, both by others as well as ourselves at a later time, when inevitably we have the general recollection that we performed a particular computation but are in need documentation to make memory of the process concrete again.

We favored inclusion over exclusion, with the goal of building a **comprehensive** summary the steps required to recreate our work product. The contents of the Appendices are grouped into the logical associations as follows:

- Appendix [B](#) helps the reader navigate the potentially opaque world of acronyms used as a technical shorthand throughout the text.
- Appendix [C](#) focuses on the Sandia high-fidelity human digital twin, explaining the model's **geometric representation** of the underlying human anatomy.
- Appendix [D](#) explains how we endow the geometric domain with a mechanical response, implemented through **material models** and **material properties**.
- Appendix [E](#) records details of several **methods** that we developed to provide objective, quantitative measures of our results.
- Appendix [F](#) memorializes the exact input decks and post-processing steps used to arrive at our results. While this section is rather long, we felt it necessary to maintain its unabridged form to promote repeatability and integrity of our methods.

APPENDIX B. Acronyms

Table B-1. Abbreviations used throughout the text.

Abbreviation	Definition
ACH	Advanced Combat Helmet
ant	anterior (anatomical direction)
AP	anterior-posterior (anatomical axis)
APL	Applied Physics Laboratory
ARA	Applied Research Associates
cbmTBI	cellular-based mild traumatic brain injury
CSF	cerebral spinal fluid
CTH	(not an acronym) the name of a shock hydrodynamic code at Sandia
DOE	Department of Energy
DOT	Department of Transportation
G-level	gravitational level on Earth, equal to 9.81 m/s^2
hel	helmet
HIC	head injury criterion
inf	inferior (anatomical direction)
inf-sup	inferior-superior (anatomical axis)
mil-spec	military specification
mTBI	mild traumatic brain injury
post	posterior (anatomical direction)
PPE	personal protective equipment
SIBL	Sandia Injury Biomechanics Laboratory
SNL	Sandia National Laboratories
SSM	Sierra Solid Mechanics
sup	superior (anatomical direction)
TBI	traumatic brain injury
WSTC	Wayne State Tolerance Curve

APPENDIX C. Geometry

C.1. Solid Model

From U.S. National Library of Medicine, Visible Human Project [[Ackerman, 1998](#)],

- manual segmentation
- CT and digital photography, scan of entire body
- full body 1,871 axial slices at 1 mm intervals
- CT: 512×512 pixels; 120-bit gray
- Photo: $4,096 \times 2,700$ pixels; 24-bit color
- MR head and neck: axial slices at 4 mm intervals; 256×256 pixels; 12-bit gray

The human digital twin geometric model was originally developed for blast and ballistic simulations in CTH. Beginning in 2018, our laboratory developed a finite element model from the finite volume mode, for use with simulations in SSM.

The human digital twin, colloquially referred to as “Bob,” has been used with several numerical experiments and validations. The Bob model consists of a geometric model, describing the anatomy of the human subject, and a collection of material models, describing the constitutive response of the model to blast, blunt impact, and ballistic loading conditions.

C.1.1. *Historical Validations and Use*

[[Taylor et al., 2018](#)] validated the model against head impact data originating from Nahum [[Nahum et al., 1977](#)] and Feng [[Feng et al., 2010](#)]. Additional, Taylor validated the model against laboratory blast results from Johns Hopkins Applied Physics Laboratory (APL) [[Merkle, 2011](#)].

The model was used extensively by Taylor and coworkers to predict the human head and torso response to field-relevant blast scenarios [[Haniff and Taylor, 2017](#)], [[Taylor et al., 2014a](#)], [[Taylor et al., 2014b](#)], [[Vakhtin et al., 2013](#)], and [[Taylor and Ford, 2009](#)]. When these historical efforts examined the topic of cavitation, it was done through the context of liquid-to-vapor content of the cerebral spinal fluid, modeled with a Tillotson equation of state modified by Brundage [[Brundage, 2013](#)], [[Brundage, 2014](#)] and based on the original work by Tillotson to model metal vaporization secondary to hypervelocity compressive impact [[Tillotson, 1962](#)].

C.1.2. Present Use

Our current blast methodology is a departure from and simplification of the foregoing Tillotson approach. Our blast simulations used the CONWEP [Veilleux et al., 2019] functionality of Sierra Solid Mechanics (SSM) [Merewether et al., 2020], a three-dimensional Lagrangian code for nonlinear analysis of solids and structures, instead of the historically-used CTH [Hertel et al., 1995], a three-dimensional Eulerian code for analysis of shock physics. Our blunt impact scenarios utilize SSM in an explicit time-stepping scheme.

Significant effort was devoted to translating the 3D solid model from the Eulerian finite volume context to the Lagrangian finite element context. We used Sculpt [Owen et al., 2011], [Owen et al., 2014], [Owen et al., 2019] to convert the solid finite volumes to hexahedral finite elements. We attempted to smooth all domains from their native cubic 1-mm \times 1-mm \times 1-mm Eulerian form into a Lagrangian form with shape reflecting the underlying anatomical geometry.

We discovered that smoothing the skin domain was readily achievable, but smoothing the domains interior to the skin was problematic, resulting in an explosion of the finite element population beyond what we considered tractable. Moreover, the many of the newly-formed non-hexahedral elements suffered from poor shape quality.¹ Finally, we encountered instances where entire anatomical volumes, in particular the CSF, would be eliminated based on Sculpt-based morphology criteria, such as decimate, loft, and pillow.

While we devoted considerable resources to smoothing the interior domains, we ultimately decided to forego smoothing, electing to leave the internals as regular hexahedral finite elements. We refer to this shape schema as “sugar cubes,” inspired by the way in which sugar can sometimes be found for coffee and tea service: sold in boxes as a collection of small, solid cubes stacked into layers of rows and columns.

The election of the sugar cube approach turned out to be serendipitous, as subsequent analyses in the finite element domain were easily comparable to historical analyses in the finite volume domain. Aside from the skin, all domains, e.g., skull, sinus/airways, CSF, gray matter, white matter, etc., have individual finite elements that are a one-to-one spatial map of their finite volume counterpart.

In what follows, we present Figure C-1 as a historical context of the anatomy underlying the development of both the finite volume and the finite element models. Figures C-2–C-9 show the new finite element model. Finally, Figure C-10 demonstrates that the new finite element model (shown with measurements for *with* and *without* skin) represents the human form with a high degree of fidelity, particularly when compared to other available human models [Giudice et al., 2019].

We believe this high number of elements, even with their sugar cube format, offers sufficient smoothing of post-processing metrics, such as stress or strain, because the

¹In contrast, hexahedral elements have idealized shape quality, since the map from local to global domains is a simple scaling, absent of degeneration.

element size is so small (cube side length of 1-mm). Moreover, our post-processing efforts report element data in the context of a population rather than a single number. We have indicated the population’s 95th percentile thresholds as a way to avoid reporting data this is an outlier, potentially caused by numerical noise. Such a practice has widespread adoption [[Giudice et al., 2019](#)].

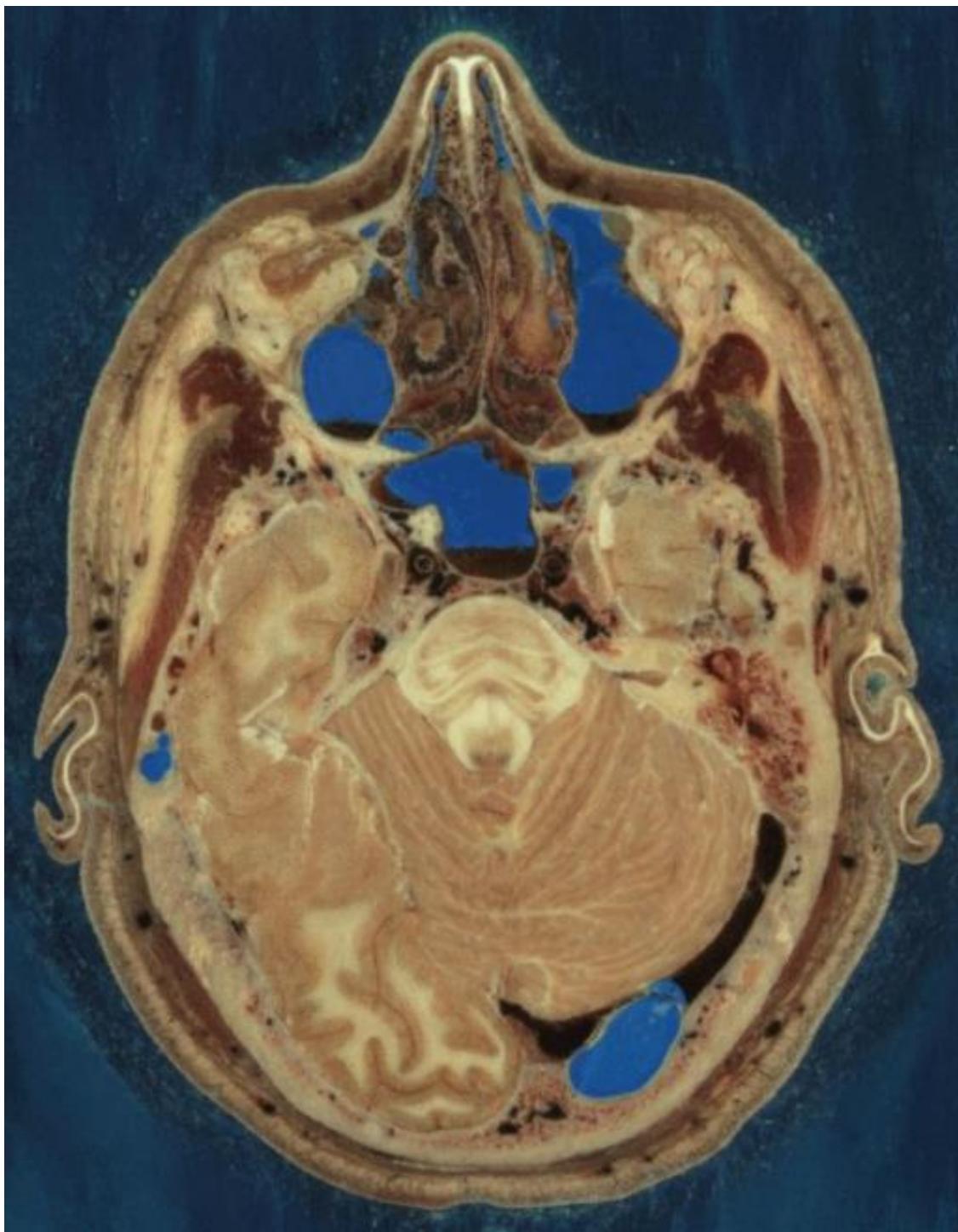


Figure C-1. NIH U.S. National Library of Medicine axial section, showing approximate A-A section indicated in Figure [C-2](#).

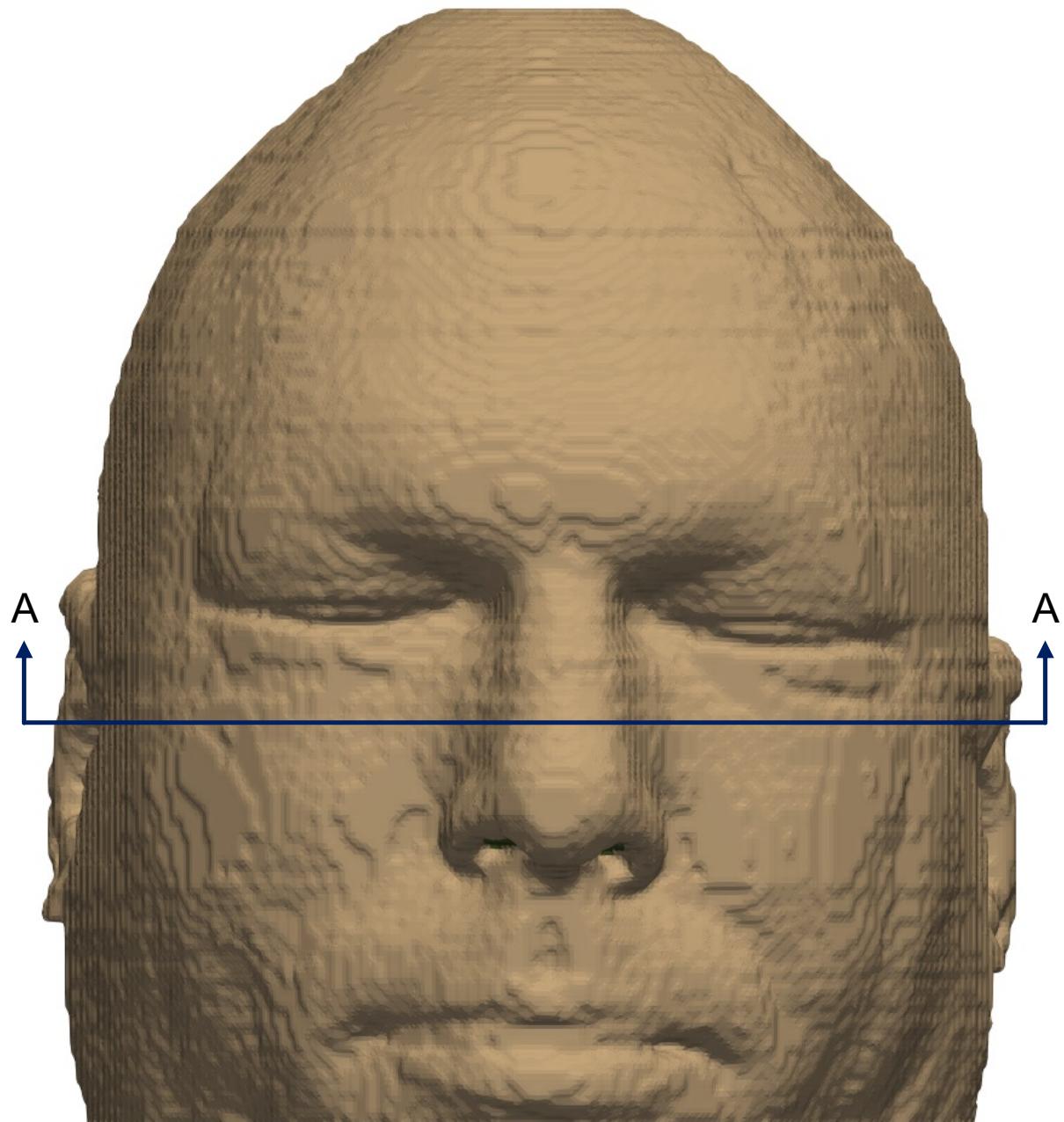


Figure C-2. Frontal view of finite element human digital twin with A-A axial section indicated, the approximate location shown in Figure C-1.

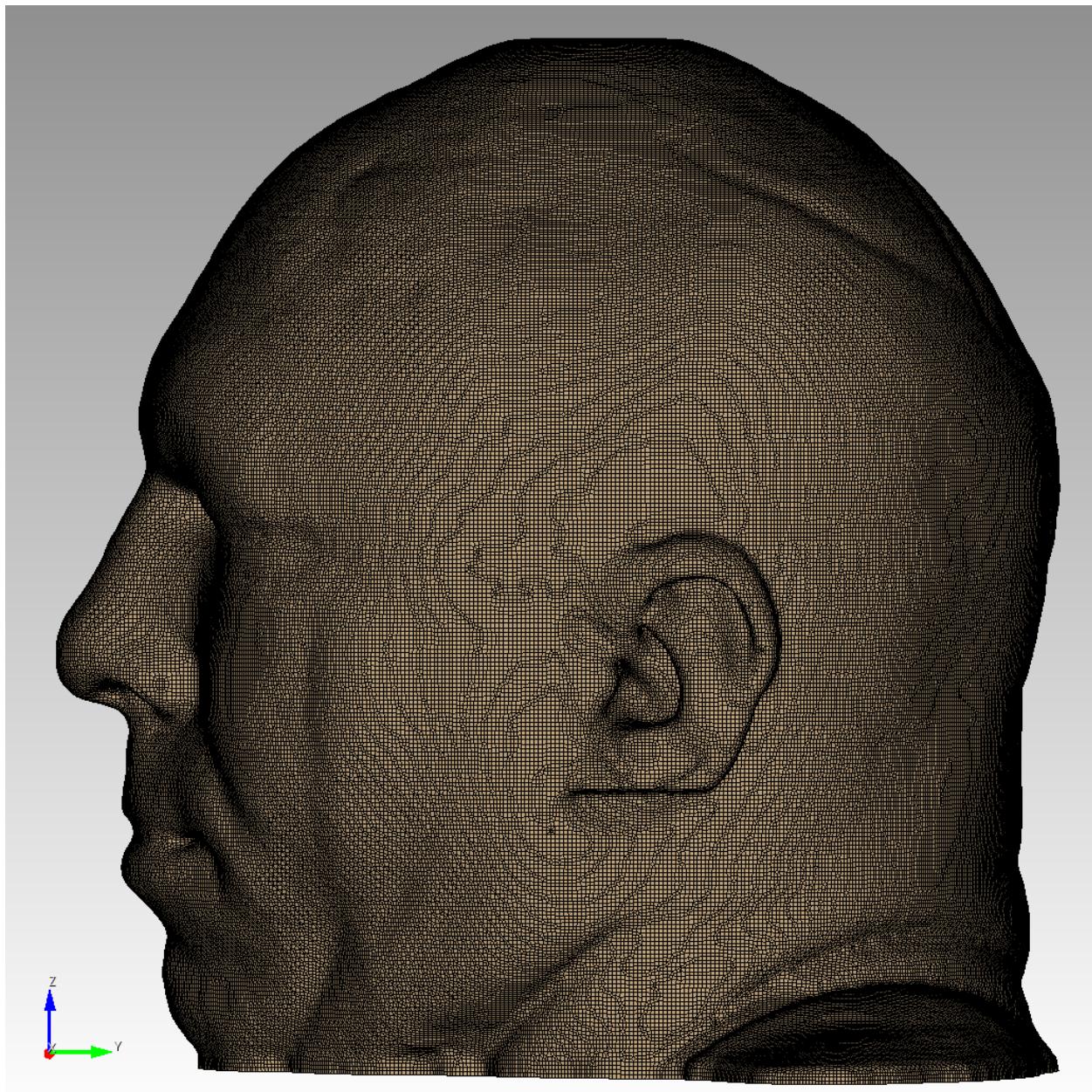


Figure C-3. Left side view.

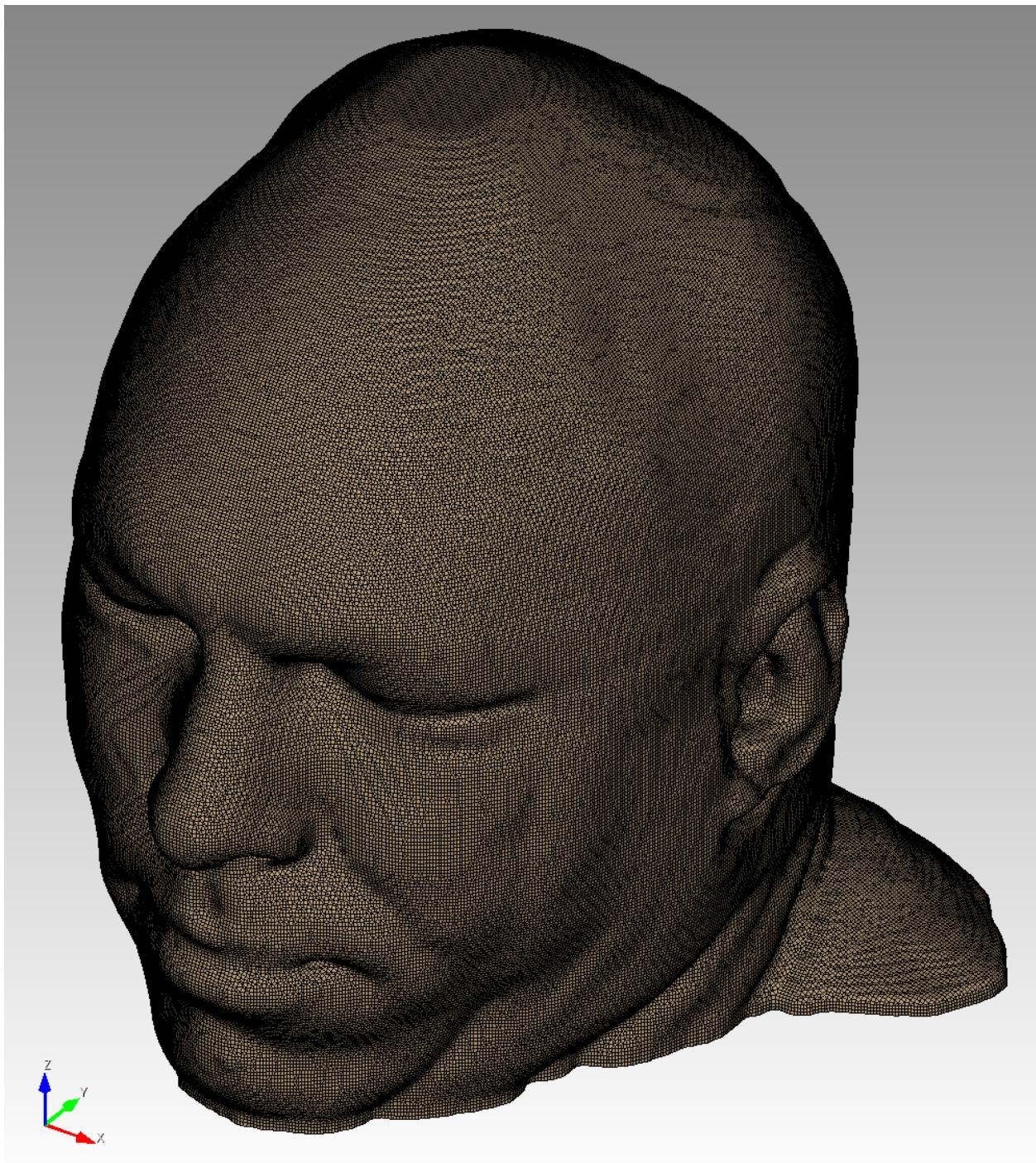


Figure C-4. Front, left, top isometric view of model.

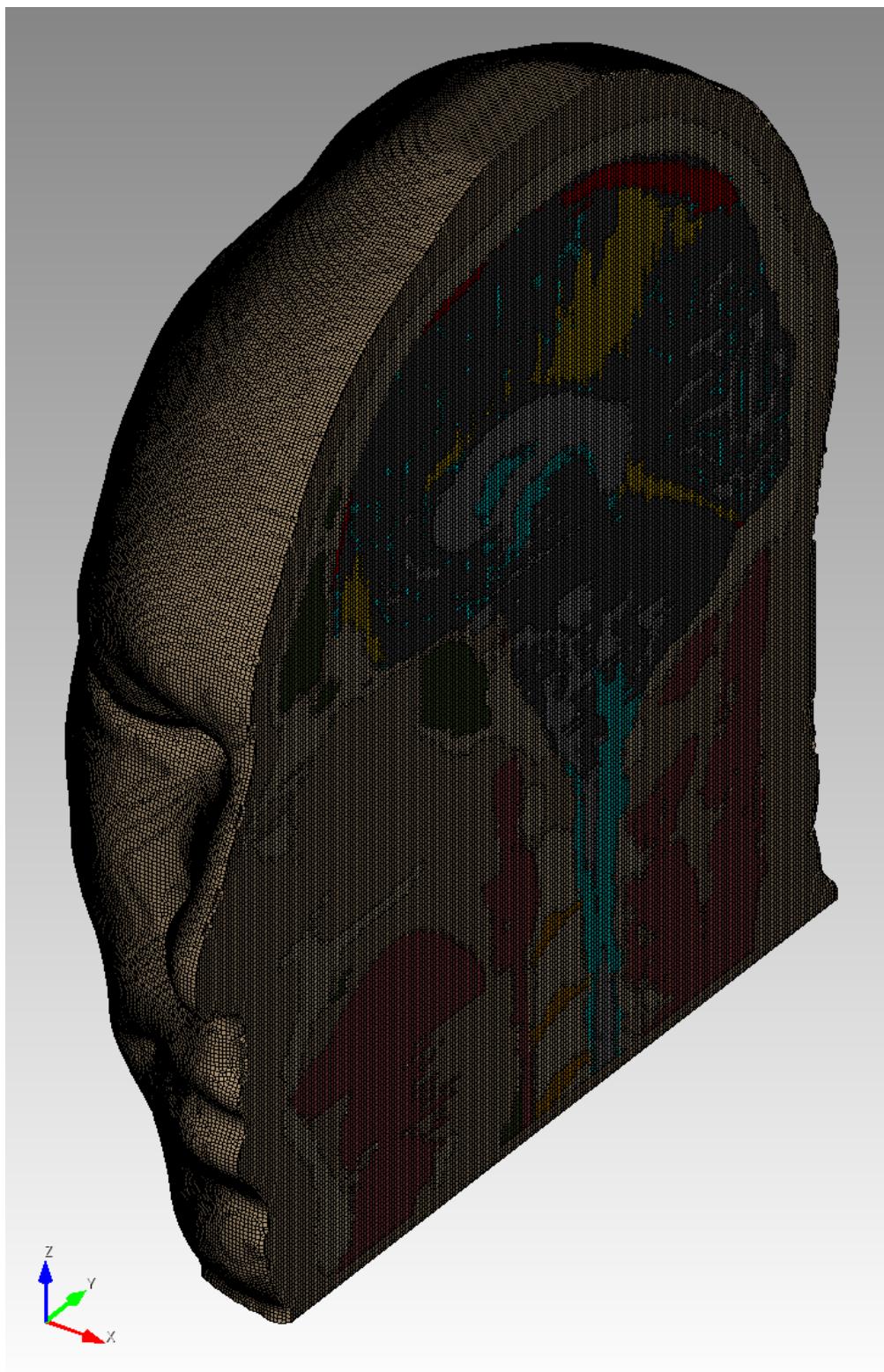


Figure C-5. Front, left, top isometric view of model, with mid-sagittal exposure.



Figure C-6. Front, left, top isometric view of model, skin removed.

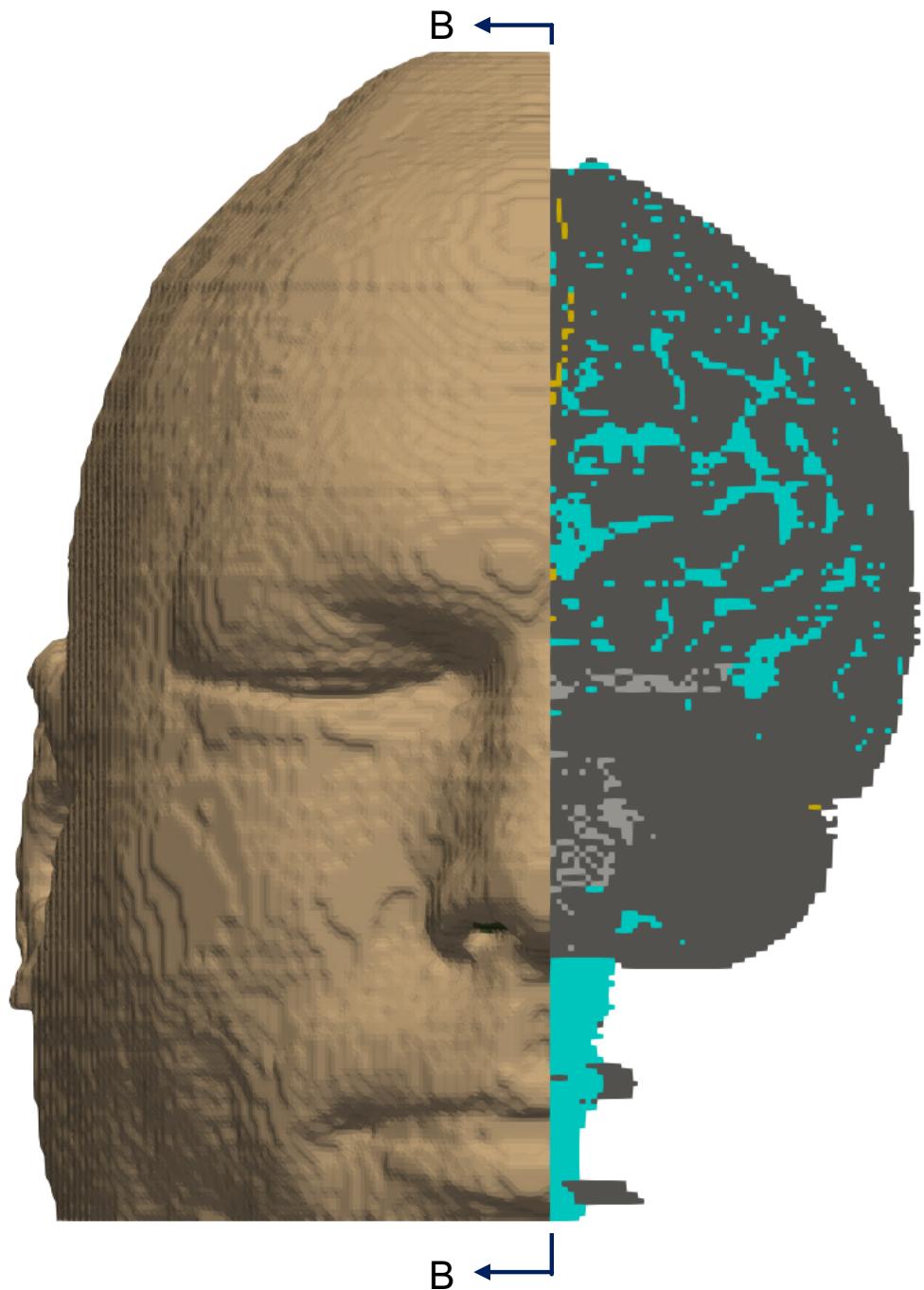


Figure C-7. Front view with left-side skin, muscle, and bone removed to show brain at Section B-B, shown in Figure [C-8](#).

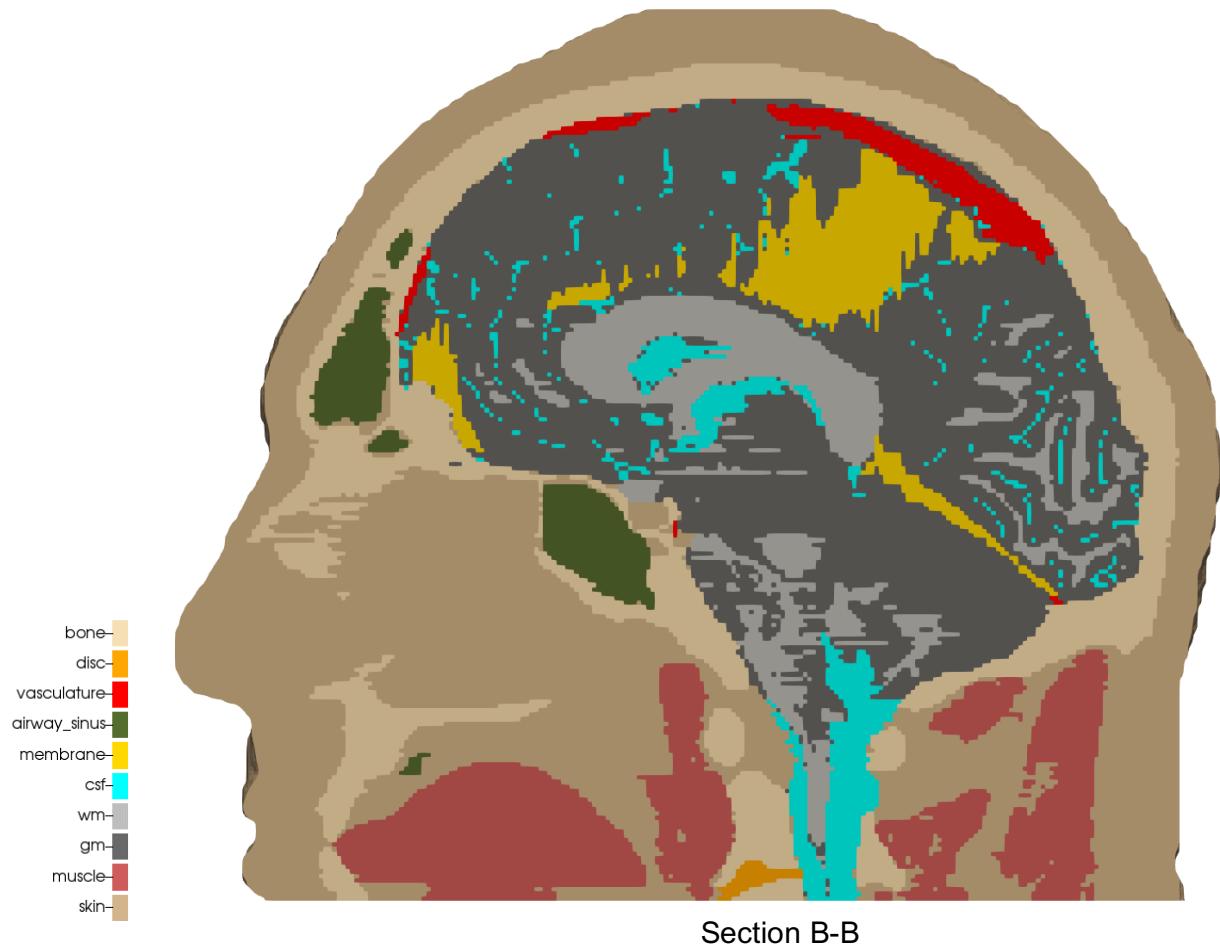


Figure C-8. Mid-sagittal B-B section view indicated in Figure C-7, with anatomical structures labelled and color coded.

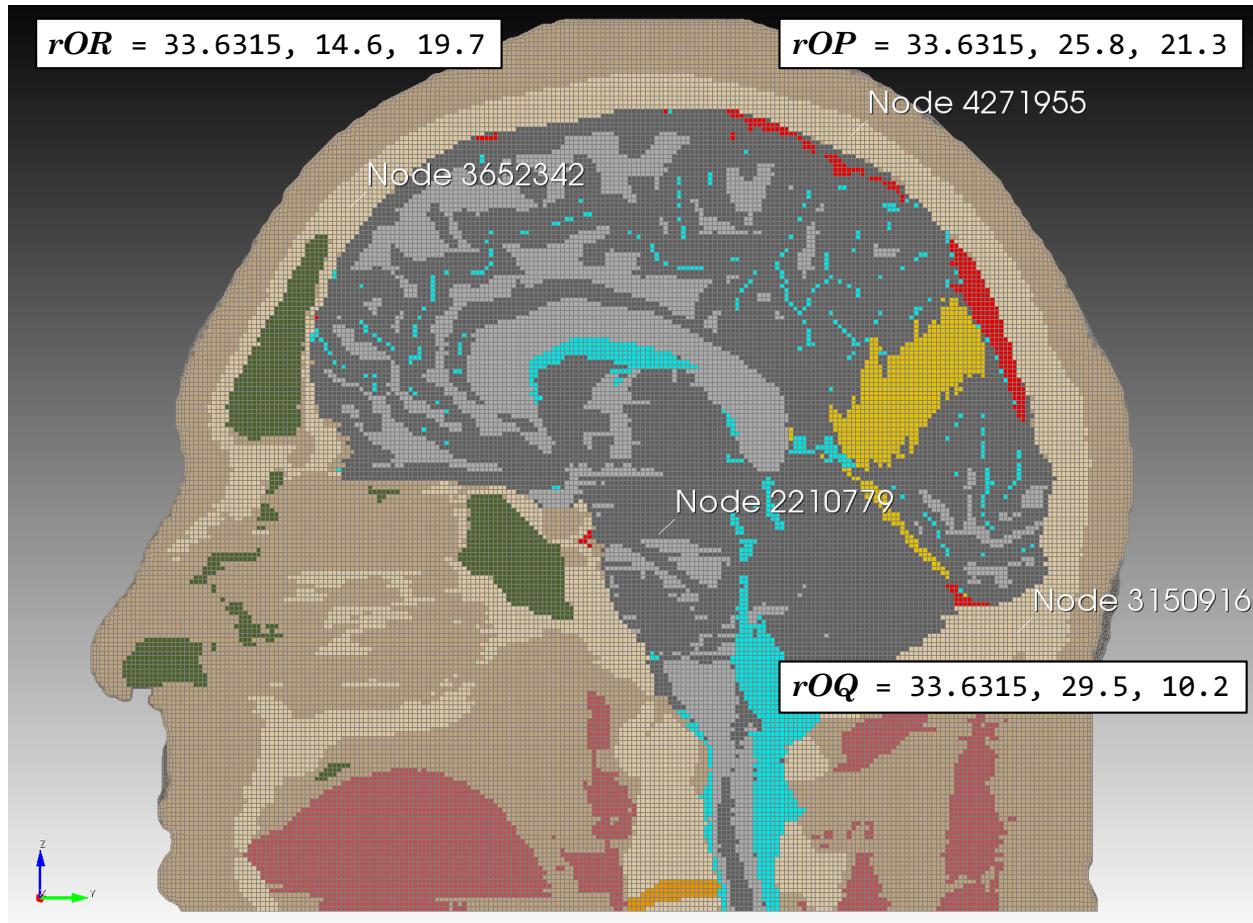


Figure C-9. Mid-sagittal view, showing three points P , Q , and R used for the three-point angular velocity (tpav) algorithm, and the center of mass, node 2,210,779.

Based on bob-1mm-5kg-helmet2-0305-hemi-066b.log,

Table C-1. High-Fidelity Human Surrogate (Bob) Materials Details.

Block # (int)	Geometry	# Elements (int)	(%)	Mass (grams)	(%)
1	bone	670,293	14.47	8.11052E+02	16.188
2	disc	2,221	0.05	2.22100E+00	0.044
3	vasculature	21,642	0.47	2.17286E+01	0.434
4	airway/sinus	93,465	2.02	1.13617E-01	0.002
5	membrane	21,234	0.46	2.40581E+01	0.480
6	CSF	73,172	1.58	7.34647E+01	1.466
7	white matter	504,505	10.89	5.24685E+02	10.472
8	gray matter	790,102	17.06	8.21706E+02	16.400
9	muscle	554,555	11.97	6.65459E+02	13.282
10	skin	1,900,127	41.03	2.06581E+03	41.231
Total		4,631,316	100.00	5.01030e+03	100.000

For consistency in color attributed to each material, we note the color scheme in Table C-2.

Table C-2. Material color scheme.

Block # (int)	Color Name	RGB triple (double, double, double)
1	wheat	0.961, 0.871, 0.702
2	orange	1.000, 0.647, 0.000
3	red	1.000, 0.000, 0.000
4	dark olive green	0.333, 0.420, 0.184
5	gold	1.000, 0.843, 0.000
6	cyan	0.000, 1.000, 1.000
7	gray	0.745, 0.745, 0.745
8	dim gray	0.412, 0.412, 0.412
9	faded red	0.804, 0.361, 0.361
10	tan	0.824, 0.706, 0.549

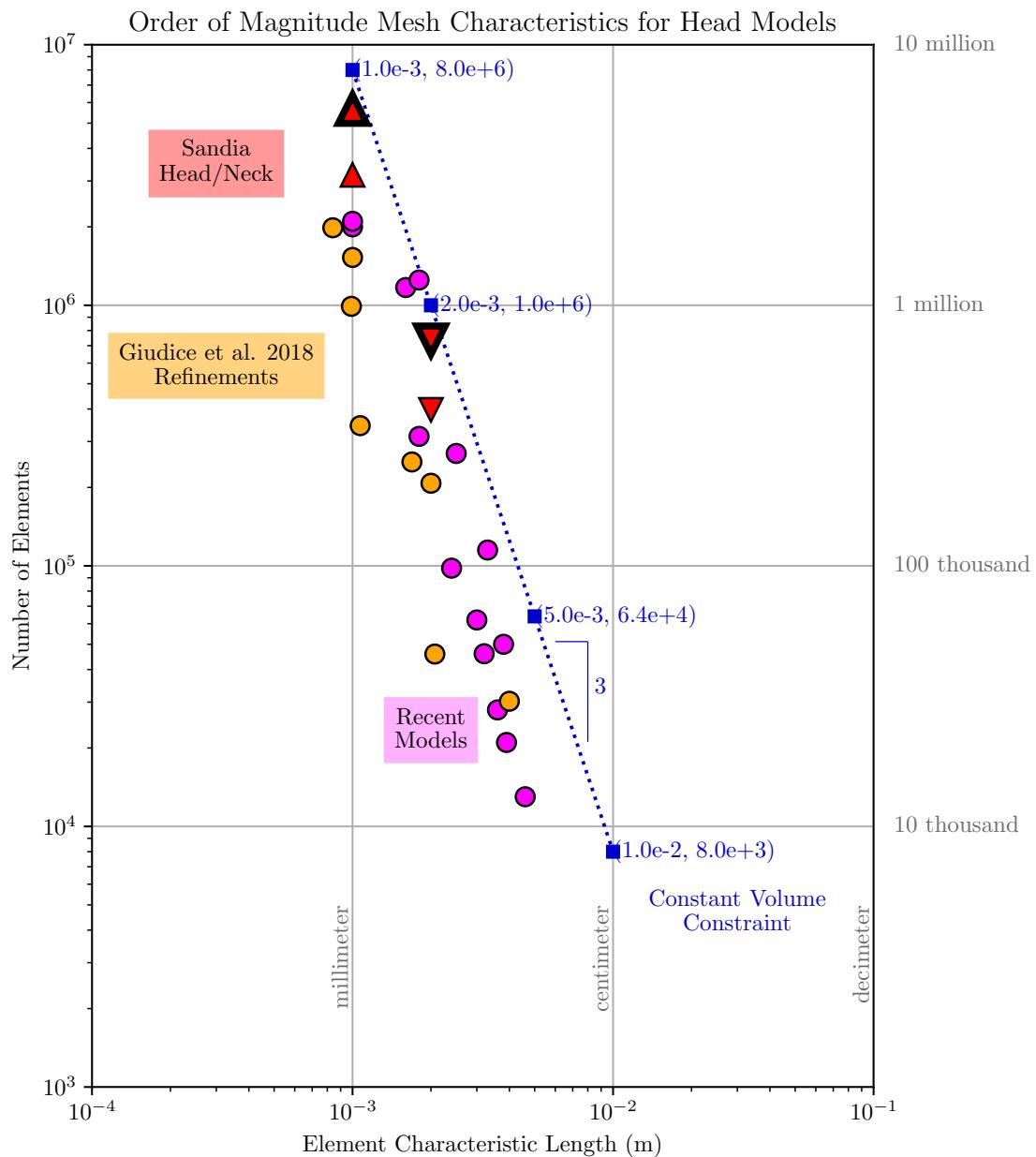


Figure C-10. Element density versus element scale for the Sandia digital human twin model (red triangles), in context of other models (orange and magenta circles), summarized in [Giudice et al., 2019]. Triangles with bold, black borders include skin. Triangles with fine, black borders exclude skin. Triangles pointed up are the 1-mm (fine) models, with and without skin. Triangles pointed down are the 2-mm (coarse) models, with and without skin.

APPENDIX D. Materials

D.1. Material Models

D.1.1. Finite Deformation Elastic

All Sierra Solid Mechanics (SSM) material models used in this study employed finite deformation kinematic measures.¹ The SSM elastic model is extended from a hypoelastic extension of isotropic, small-strain, linear elasticity model. Given Lamé constants λ and μ , the stress response function for an elastic, isotropic material is

$$\boldsymbol{\sigma} = \lambda \operatorname{tr}(\boldsymbol{\epsilon}) + 2\mu \boldsymbol{\epsilon}. \quad (\text{D.1})$$

The foregoing model is extended to a finite-deformation, hypoelastic model by making it a rate equation. Given the **rate of deformation** tensor \mathbf{d} ($\mathbf{d} = \operatorname{SYM}(\nabla \mathbf{v})$, the symmetric part of the spatial velocity gradient) the stress is replaced with the **Green-McInnis stress rate** $\overset{\square}{\boldsymbol{\sigma}}$.²

$$\overset{\square}{\boldsymbol{\sigma}} = \lambda \operatorname{tr}(\mathbf{d}) + 2\mu \mathbf{d} \quad (\text{D.2})$$

Another common stress rate is the **Jaumann stress rate**,³ which will produce the same result as the Green-McInnis rate when the principal axes of deformation are fixed. For problems where the principal axes of deformation rotate during the deformation, the two stress rates can give different answers. There is no reason to prefer one stress rate over the other as long as the constitutive relationships between material properties and measures of deformation are well-formulated [Brannon, 1998].

D.1.2. Finite Deformation Elastic-Plastic

The elastic-plastic extends the elastic formulation, allowing for permanent plastic deformation when the effective stress exceeds a yield surface.⁴ The SSM elastic-plastic model is a hypoelastic, rate-independent, linear hardening plasticity formulation. The rate form of the constitutive equation assumes an additive split of the rate of deformation tensor $\mathbf{d} = \operatorname{SYM}(\nabla \mathbf{v})$ into an elastic part and a plastic part,

$$\mathbf{d} = \mathbf{d}^{\text{elas}} + \mathbf{d}^{\text{plas}}. \quad (\text{D.3})$$

¹Internal reference: [../material/ssm_elastic.tex](#).

²Also called the Green-Naghdi and the Green-McInnis-Naghdi stress rate.

³Also called the Jaumann-Zaremba stress rate.

⁴Internal reference: [../material/ssm_elastic_plastic.tex](#)

The effective stress⁵ $\phi(\boldsymbol{\sigma}, \boldsymbol{\alpha})$ is a function of the Cauchy stress $\boldsymbol{\sigma}$ and the back stress $\boldsymbol{\alpha}$. The Green-McInnis stress rate $\overset{\square}{\boldsymbol{\sigma}}$ depends only on the elastic strain rate \mathbf{d}^{elas} as

$$\overset{\square}{\boldsymbol{\sigma}} = \mathbf{d}^{\text{elas}}. \quad (\text{D.4})$$

Given the hardening modulus H , yield stress σ_y , and equivalent plastic strain ϵ^{eps} , the linear hardening law is assumed,

$$\sigma^{\text{eff}} = \sigma_y + H\epsilon^{\text{eps}}. \quad (\text{D.5})$$

- If $\beta = 0.0$, the plasticity model uses only **kinematic hardening**, wherein the yield surface does not grow in size, but only moves in stress space.
- If $\beta = 1.0$, the plasticity model uses only **isotropic hardening**, wherein the yield surface grows in size, but does not move in stress space.
- If $\beta \in (0.0, 1.0)$, then the plasticity uses a **mixture** of both kinematic and isotropic hardening.

D.1.3. Finite Strain Neo-Hookean

Prior to considering the Swanson viscoelastic response, we explicate some details pertaining to finite strain neo-Hookean models, often used as comparison to the Swanson model.⁶ Following is a development, from [Simo and Hughes, 2006], of the finite deformation neo-Hookean model from the original developments of Rivlin.

Let I_1, I_2, I_3 be the three invariants of the two Cauchy-Green deformation tensors. The invariants are defined in terms of principal stretch ratios $\lambda_1, \lambda_2, \lambda_3$,

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \quad (\text{D.6})$$

$$I_2 = \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_3^2 \lambda_1^2, \quad (\text{D.7})$$

$$I_3 = \lambda_1^2 \lambda_2^2 \lambda_3^2 = J^2. \quad (\text{D.8})$$

Rivlin used a strain energy density function W as power series of the invariants [Rivlin, 1956],

$$W = \sum_{i+j+k=1}^{\infty} C_{ijk} (I_1 - 3)^i (I_2 - 3)^j (I_3 - 1)^k. \quad (\text{D.9})$$

For **incompressible** materials, $I_3 = 1$, and the Rivlin equation reduces to

$$W = \sum_{i+j=1}^{\infty} C_{ij} (I_1 - 3)^i (I_2 - 3)^j. \quad (\text{D.10})$$

⁵Also called the Huber-Mises stress or Mises effective stress.

⁶Internal reference: ../material/hookean.tex

Remark D.1.1. The presence of the $-3, -3$, and -1 terms in parenthesis allow for zero strain energy density when there is zero stretch.

Remark D.1.2. When a material's bulk modulus K is at least two orders of magnitude larger than the shear modulus G , the incompressibility modeling assumption may be appropriate.

Incompressible Rivlin

Using on the first two terms of the incompressible Rivlin gives the **Mooney-Rivlin** strain energy density function:

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3). \quad (\text{D.11})$$

Incompressible Neo-Hookean

Using only the first term of the Mooney-Rivlin equation gives the **Incompressible Neo-Hookean** strain energy density function:

$$W = C_{10}(I_1 - 3). \quad (\text{D.12})$$

The invariants in terms of the right Cauchy-Green tensor \mathbf{C} and left Cauchy-Green tensor \mathbf{b} are

$$I_1 = \text{tr}(\mathbf{C}) = C_{KK} = \text{tr}(\mathbf{b}) = b_{kk}, \quad (\text{D.13})$$

$$\begin{aligned} I_2 &= \frac{1}{2} (I_1^2 - \mathbf{C} : \mathbf{C}) = \frac{1}{2} (I_1^2 - C_{IK}C_{KI}) \\ &= \frac{1}{2} (I_1^2 - \mathbf{b} : \mathbf{b}) = \frac{1}{2} (I_1^2 - b_{ik}b_{ki}), \end{aligned} \quad (\text{D.14})$$

$$I_3 = \det(\mathbf{C}) = \det C_{IJ} = \det(\mathbf{b}) = \det b_{ij} = J^2. \quad (\text{D.15})$$

For small strains and incompressible Mooney-Rivlin:

$$E = 6(C_{10} + C_{01}), \quad (\text{D.16})$$

$$G = 2(C_{10} + C_{01}). \quad (\text{D.17})$$

For small strains and incompressible Neo-Hookean:

$$C_{01} = 0, \quad (\text{D.18})$$

$$C_{10} = G/2, \implies E = 3G. \quad (\text{D.19})$$

Compressible Rivlin

Next, allow the material to be compressible, characterized through bulk modulus K . Thus, $I_3 \neq 1$. The **Mooney-Rivlin equation** for compressible materials is given by

$$W = W_{\text{VOL}}(I_3) + C_{10}(I_1 - 3) + C_{01}(I_2 - 3), \quad (\text{D.20})$$

$$= W_{\text{VOL}}(I_3) + C_{10}(I_1 - 3) + C_{01}(I_2 - 3), \quad (\text{D.21})$$

where W_{VOL} is the hydrostatic (volumetric) term.

To fully uncouple the volumetric from the deviatoric responses, let the **volume preserving (isochoric) distortion invariants** be calculated as

$$\bar{I}_1 = \frac{I_1}{I_3^{(1/3)}} = \frac{I_1}{J^{(2/3)}}, \quad (\text{D.22})$$

$$\bar{I}_2 = \frac{I_2}{I_3^{(2/3)}} = \frac{I_2}{J^{(4/3)}}, \quad (\text{D.23})$$

$$J = \sqrt{I_3}. \quad (\text{D.24})$$

Then, for **compressible Rivlin**:

$$W = \underbrace{\frac{K}{2}(J - 1)^2}_{\text{VOL}} + \underbrace{C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3)}_{\text{DEV}}. \quad (\text{D.25})$$

Finally, for **compressible Neo-Hookean**:

$$W = \underbrace{\frac{K}{2}(J - 1)^2}_{\text{VOL}} + \underbrace{C_{10}(\bar{I}_1 - 3)}_{\text{DEV}}. \quad (\text{D.26})$$

In the finite compression limit, as the volume goes to zero ($J \rightarrow 0$), the $(J - 1)^2$ factor in the volumetric term tends toward unity. This behavior—finite strain energy density in the limit of compression—is not desirable. Instead, as the volume goes toward zero, the strain energy density so go toward $+\infty$. Simo and Hughes [Simo and Hughes, 2006] page 361, Eq. (10.4.19), provided an alternative form of the compressible Neo-Hookean strain energy density that exhibits this desired response,

$$W = \underbrace{\frac{K}{2} \left[\frac{J^2 - 1}{2} - \ln J \right]}_{\text{VOL}} + \underbrace{\frac{G}{2}(\bar{I}_1 - 3)}_{\text{DEV}}. \quad (\text{D.27})$$

Figure D-1 shows the volumetric term for the small strain and finite strain cases, with the relative contribution of the two terms that add to compose the finite case.

Remark D.1.3. The Simo and Hughes notation, while slightly different from above, is equivalent. For the deviatoric term, since

$$\bar{\mathbf{F}} = \frac{\mathbf{F}}{J^{(1/3)}}, \quad (\text{D.28})$$

and

$$\bar{\mathbf{C}} = \bar{\mathbf{F}}^T \bar{\mathbf{F}} = \frac{\mathbf{F}^T}{J^{(1/3)}} \frac{\mathbf{F}}{J^{(1/3)}} = \frac{\mathbf{F}^T \mathbf{F}}{J^{(2/3)}} = \frac{\mathbf{C}}{J^{(2/3)}}, \quad (\text{D.29})$$

then,

$$\text{tr}(\bar{\mathbf{C}}) = \text{tr} \left(\frac{\mathbf{C}}{J^{(2/3)}} \right) = J^{(-2/3)} \text{tr}(\mathbf{C}) = \frac{I_1}{J^{(2/3)}} = \bar{I}_1, \quad (\text{D.30})$$

which harmonizes the notation used here with the deviatoric term from Simo and Hughes.

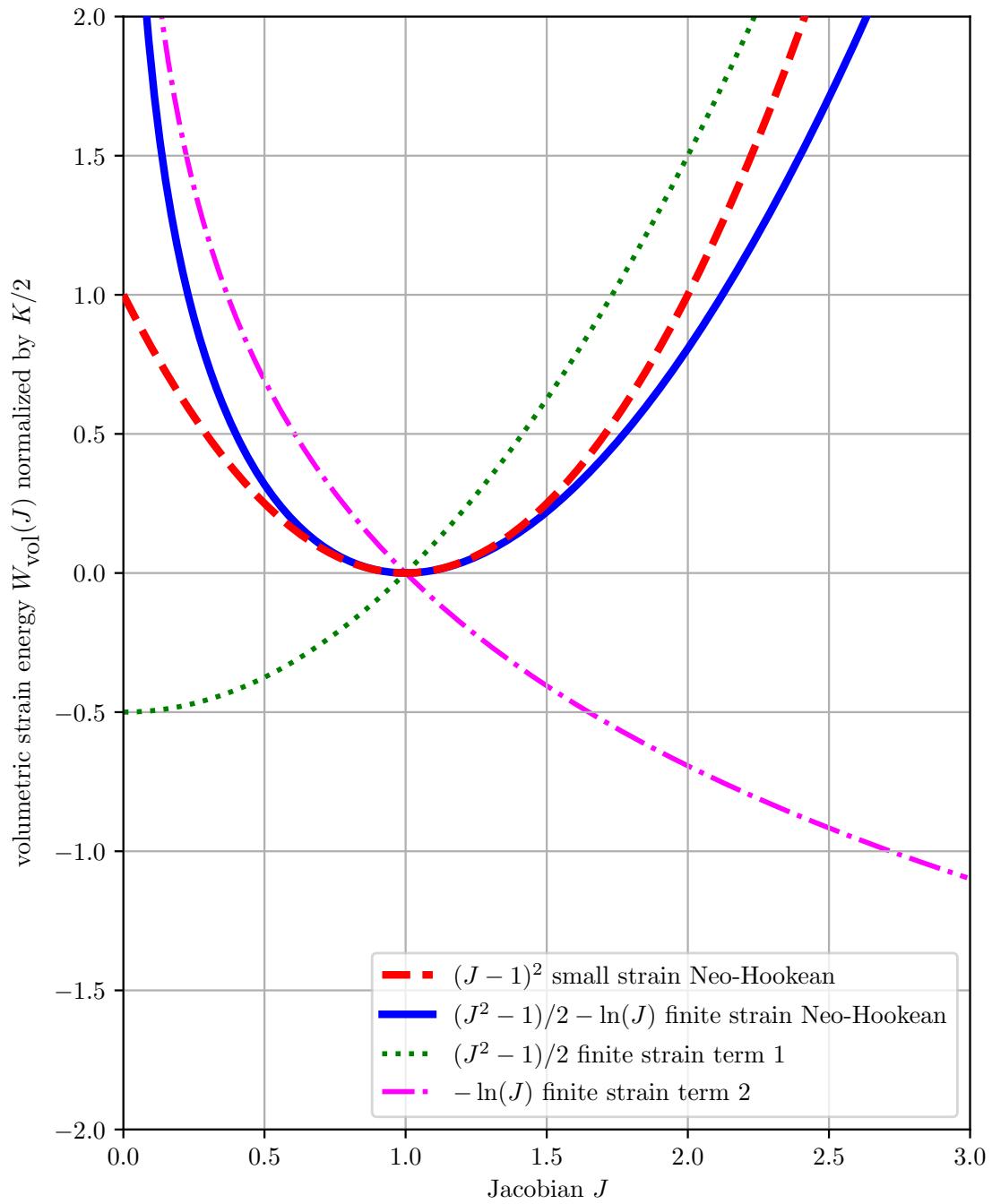


Figure D-1. Neo-Hookean volumetric strain energy density $W_{\text{vol}}(J)$ normalized by $K/2$ for infinitesimal and finite cases.

Remark D.1.4. Consider the relative effects of the $(J^2 - 1)/2$ term versus the $-\ln(J)$ term in (D.27):

- For finite compression, the $-\ln(J)$ dominates the $(J^2 - 1)/2$ term, causing the strain energy function to go toward $+\infty$, as desired.
- For finite extension, the $(J^2 - 1)/2$ term dominates the $-\ln(J)$ term, causing the strain energy function to go toward $+\infty$, as desired.
- The addition of the two terms evaluated at $J = 1$ gives $W = 0$, as desired. For zero volumetric deformation, there should be zero strain energy density.
- The slope of the strain energy density function, $dW/dJ = J - 1/J$, evaluated at $J = 1$ is zero. For zero volumetric deformation, there should be zero initial stress.
- The second derivative of the strain energy density function, $d^2W/dJ^2 = (1 + 1/J^2)$ is positive for all $J \in (0, +\infty)$. Thus, the volumetric strain energy density function W_{vol} is convex.

D.1.4. Swanson Viscoelastic

Next consider the Swanson viscoelastic response.⁷ In 1985, Swanson [Swanson, 1985], [Swanson et al., 1985] described a hyperelastic constitutive model for large deformation, slightly compressible materials.

Elastic Response

The Swanson viscoelastic model has an elastic bulk response, has a viscoelastic shear response, and is often used to simulate rubber.

The *elastic, deviatoric* contribution to the Swanson strain energy density function is given by

$$W_{\text{elas, dev}} = \frac{3}{2} \left[\frac{A_1}{P_1 + 1} \left(\frac{\bar{I}_1}{3} - 1 \right)^{P_1+1} + \frac{B_1}{Q_1 + 1} \left(\frac{\bar{I}_2}{3} - 1 \right)^{Q_1+1} + \frac{C_1}{R_1 + 1} \left(\frac{\bar{I}_1}{3} - 1 \right)^{R_1+1} \right], \quad (\text{D.31})$$

where \bar{I}_1 and \bar{I}_2 , respectively, are the first and second invariants of the isochoric left Cauchy-Green strain tensor \mathbf{b} .

The *elastic, volumetric* contribution to the Swanson strain energy density function is

$$W_{\text{elas, vol}} = K (J \ln J - J + 1). \quad (\text{D.32})$$

⁷Internal reference: ../material/swanson.tex

This volumetric contribution is compared the previously developed neo-Hookean infinitesimal and finite deformation bulk responses in Fig. D-2

The shear modulus G is calculated from the material constants A_1, P_1, B_1, Q_1 and cut off strain e_c as

$$G = A_1 \left(\frac{e_c^2}{3} \right)^{P_1} + 2B_1 \left(\frac{e_c^2}{3} \right)^{Q_1} \quad (\text{D.33})$$

Viscous Response with Prony Series

The one-dimensional relaxation test, subjecting the specimen to a step-function from zero to constant strain, gives rise to the stress-relaxation form

$$G(t) = G_\infty + \sum_{i=1}^n G_i \exp(-t/\tau_i), \quad (\text{D.34})$$

where G_∞ is the **long-term shear modulus**, G_i is a **transient shear modulus**, and τ_i is a *relaxation time*. Defining the **initial shear modulus** $G_0 \stackrel{\Delta}{=} G(t=0)$, gives

$$G(t=0) = G_\infty + \sum_{i=1}^n G_i = G_0. \quad (\text{D.35})$$

The summation thus has the form,

$$\sum_{i=1}^n G_i = G_0 - G_\infty, \quad (\text{D.36})$$

which allows (D.34) to be rewritten as

$$G(t) = G_\infty + (G_0 - G_\infty) \sum_{i=1}^n \exp(-t/\tau_i). \quad (\text{D.37})$$

In the special case where only two shear terms are used, the stress-relaxation relationship (D.37) takes the form

$$G(t) = G_\infty + (G_0 - G_\infty) \exp(-t/\tau_1). \quad (\text{D.38})$$

Example 1.

Consider the two-term special case for two different relaxation times, (1/700) s and (1/40) s, shown in Fig. D-3. Of interest is the time when the shear response reaches the midpoint between G_0 and G_∞ , referred to as t_{mid} and G_{mid} , respectively.

$$G_{\text{mid}} = G_\infty + \frac{1}{2}(G_0 - G_\infty) = \frac{G_0 + G_\infty}{2}. \quad (\text{D.39})$$

This point is reached when the factor $\exp(-t/\tau) = \frac{1}{2}$, or

$$t_{\text{mid}} = -\tau \ln\left(\frac{1}{2}\right) \approx 0.69 \tau. \quad (\text{D.40})$$

These midpoint values are shown for the $\tau = (1/40)$ s case in Fig. D-3.

...

The two relaxation times presented in the preceding example represent the initial and revised values used by Taylor [Taylor et al., 2018] for white/gray matter. Taylor wrote, at pages 23-24,

“We initially conducted the Bayly rotation experiments using a viscoelastic decay constant β of 700/s as reported by Zhang *et al.* [14] for the white and gray brain matter. This parameter determines how quickly material shear stresses relax in response to a shear deformation. Unfortunately, the β decay constant value of 700/s damped out our predicted shear strains far quicker than the experimental data displayed. As a result, we varied this parameter in our simulations until we were able to match the experimental results and arrived at a decay constant of 40/sec. When simulating the rotation experiments using this latest value of β [= 40/s], our predicted shear strains persisted out to the longer times [as] displayed by the experimental data.⁸ Consequently, we conducted the remainder of our [...] simulations...] with the modified value of 40/s for β ...”

Here, Taylor reported a decay constant β , which is the reciprocal of relaxation time $\tau = 1/\beta$.

⁸With $\beta = 40/s$, the simulation results matched experimental results.

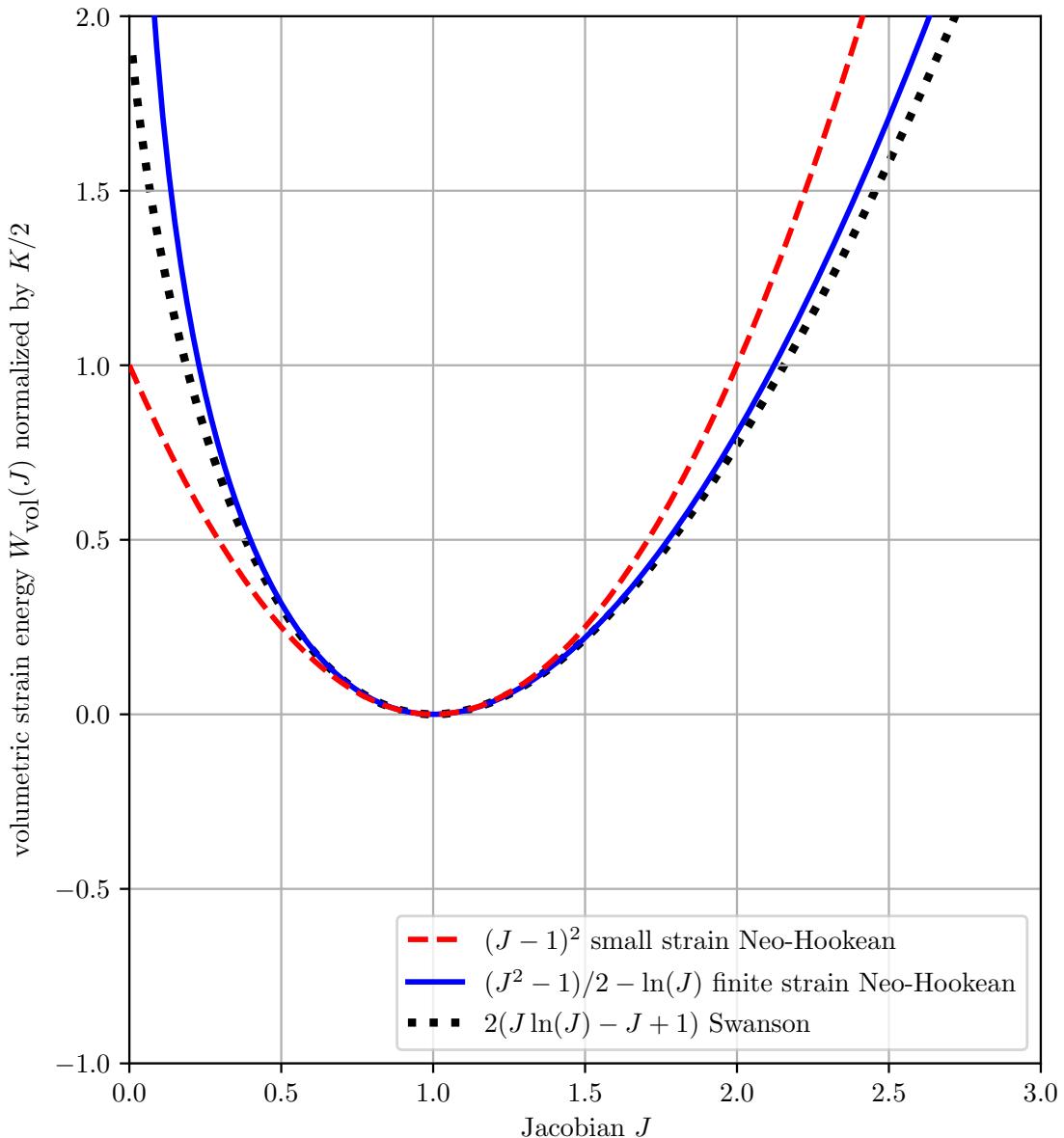


Figure D-2. Swanson volumetric elastic strain energy compared to Neo-Hookean volumetric strain energy density for infinitesimal and finite cases. All strain energy density functions $W_{\text{VOL}}(J)$ are normalized by $K/2$.

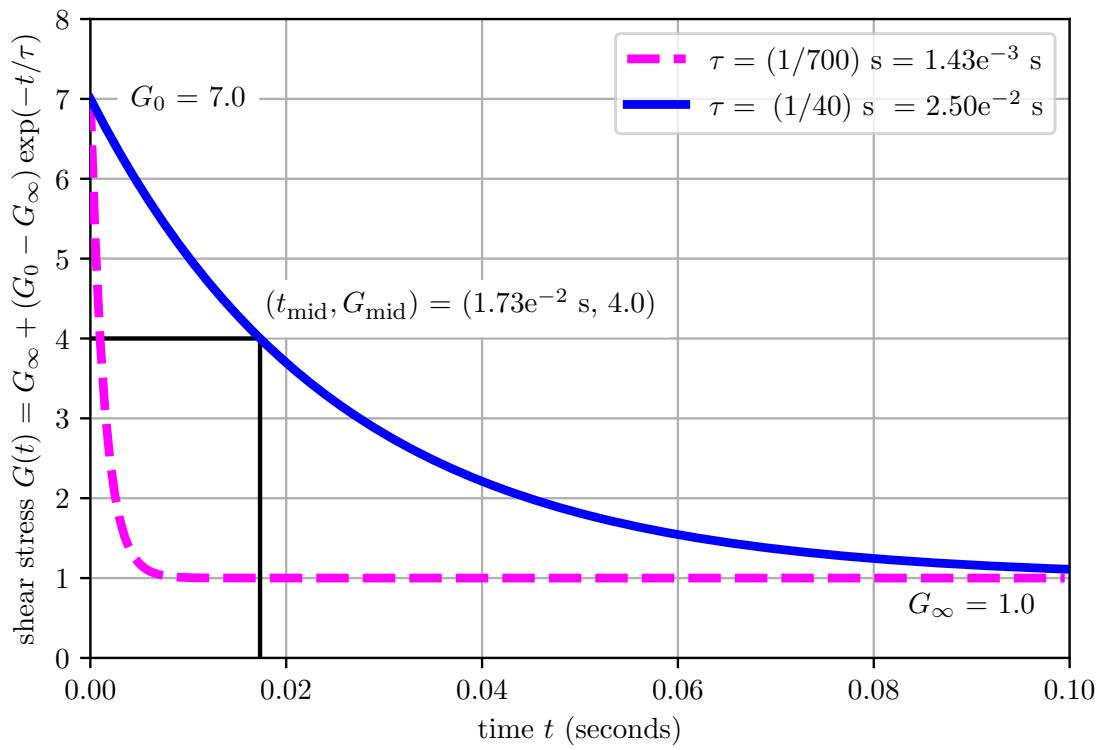


Figure D-3. Two-term viscous shear response for two different relaxation times.

D.2. Material Properties

Below we record the material properties used for the human and PPE models. The human consists of bone, vertebral discs, vasculature, airway/sinus, membrane, cerebral spinal fluid (CSF), white matter (WM), gray matter (GM), muscle, and skin. The helmet consists of the shell and pads. The pads are composed of two layers: hard and soft. The hard (protective) layer resides between the helmet shell interior and the soft layer. The soft layer resides between the hard layer and the exterior of the scalp. The hemispherical impactor, not used in the simulations herein but included for completeness, is steel.

An overview summary of the material properties is shown in the following table.⁹

Table D-1. Material Properties Summary. All material models are finite deformation models. Elastic (E), elasto-plastic (EP), Swanson viscoelastic (SVE).

material (model)	ρ_0 (kg/m ³)	K (Pa)	G_0 (Pa)	other
bone (EP)	1.210e3	4.762e9	3.279e9	$\sigma_y = 6.400e7$ Pa, $H = 0$ Pa, $\beta = 1.0$
disc (E)	1.000e3	7.117e7	1.525e7	—
vasculature (E)	1.004e3	2.371e9	5.000e2	—
airway/sinus (E)	1.218e0	2.330e7	4.691e5	—
membrane (E)	1.133e3	1.050e8	1.086e7	—
CSF (E)	1.004e3	2.371e9	5.000e2	—
WM (SVE)	1.040e3	2.371e9	4.100e4	$G_\infty = 7.8e3$ Pa, $\beta = 40$ s ⁻¹
GM (SVE)	1.040e3	2.371e9	3.400e4	$G_\infty = 6.4e3$ Pa, $\beta = 40$ s ⁻¹
muscle (E)	1.200e3	3.479e7	5.880e6	—
skin (E)	1.200e3	3.479e7	5.880e6	—
helmet shell (EP)	1.440e3	8.333e10	3.846e10	$\sigma_y = 8.200e8$ Pa, $H = 1.360e9$ Pa, $\beta = 0.5$
hard foam (EP)	5.900e1	1.942e6	2.118e6	$\sigma_y = 2.100e5$ Pa, $H = 2.500e0$ Pa, $\beta = 1.0$
soft foam (EP)	5.600e1	3.542e5	3.864e5	$\sigma_y = 4.800e4$ Pa, $H = 9.100e4$ Pa, $\beta = 1.0$
steel impactor (E)	8.000e3	1.532e11	7.481e10	—

⁹Internal reference: material/doc/materials_summary.tex.

D.2.1. Bone

We model bone with the following material properties:¹⁰

Table D-2. Bone material properties.

property		CGS	MKS
density	ρ_0	1.210 g/cc	1210 kg/m ³
P-wave sound speed	c_L	2.747e5 cm/s	2.747e3 m/s
elastic modulus	E	8.0e10 dyne/cm ²	8.0e9 Pa
Poisson ratio	ν	0.22	same
bulk modulus	K	4.762e10 dyne/cm ²	4.762e9 Pa
shear modulus	G	3.279e10 dyne/cm ²	3.279e9 Pa
yield stress	σ_y	6.400e8 dyne/cm ²	6.400e7 Pa
yield evolution (kine/iso) (0/1)	β	1.0 (iso)	same
hardening modulus	H	0.0 dyne/cm ²	same

These material properties were used in Zhang [Zhang et al., 2001],

The material properties of cranial bone used in this study were based on the experimental data reported by McElhaney et al. (1970) and Wood (1971). An equivalent Young's modulus of 8 GPa and a yield strain of 0.8% were assumed for the cranium bone, which was described by elastic-plastic constitutive equations. Besides the brain and skull, all other tissues of the head were assumed to be linearly elastic, homogeneous, and isotropic. Material properties used for each anatomic component of the WSUBIM used in the study are listed in Table 1 [skull density 1,210 kg/m³, Young's modulus 8.0 GPa, and Poisson's ratio 0.22].

¹⁰Internal reference: ../material/bone.tex.

D.2.2. Disc

We model disc with the following material properties:¹¹

Table D-3. Vertebral discs material properties.

property		CGS	MKS
density	ρ_0	1.000 g/cc	1000 kg/m ³
P-wave sound speed	c_L	3.025e4 cm/s	3.025e2 m/s
elastic modulus	E	4.27e8 dyne/cm ²	4.27e7 Pa
Poisson ratio	ν	0.40	same
bulk modulus	K	7.117e8 dyne/cm ²	7.117e7 Pa
shear modulus	G	1.525e8 dyne/cm ²	1.525e7 Pa

The Young's modulus is based on Yang [Yang et al., 2016] Figure 2, for 30-year-old at 20 s⁻¹ strain rate.

D.2.3. Vasculature

Vasculature is currently modeled as identical to cerebral spinal fluid.

¹¹Internal reference: .../material/disc.tex.

D.2.4. Airway/Sinus

We model the airway/sinus with the following material properties:¹²

Table D-4. Airway and sinus material properties.

property		CGS	MKS
density	ρ_0	1.218e-3 g/cc	1.218 kg/m ³
P-wave sound speed	c_L	4.432e5 cm/s	4.432e3 m/s
elastic modulus	E	1.398e7 dyne/cm ²	1.398e6 Pa
Poisson ratio	ν	0.49	same
bulk modulus	K	2.330e8 dyne/cm ²	2.330e7 Pa
shear modulus	G	4.691e6 dyne/cm ²	4.691e5 Pa

D.2.5. Membrane

We model the membranes with the following material properties:¹³

Table D-5. Membrane (falx and tentorium) material properties.

property		CGS	MKS
density	ρ_0	1.133 g/cc	1133 kg/m ³
P-wave sound speed	c_L	3.247e4 cm/s	3.247e2 m/s
elastic modulus	E	3.15e8 dyne/cm ²	3.15e7 Pa
Poisson ratio	ν	0.45	same
bulk modulus	K	1.050e9 dyne/cm ²	1.050e8 Pa
shear modulus	G	1.086e8 dyne/cm ²	1.086e7 Pa

¹²Internal reference: ../material/airway.tex.

¹³Internal reference: ../material/membrane.tex.

D.2.6. Cerebral Spinal Fluid (CSF)

We model CSF with the following material properties:¹⁴

Table D-6. Cerebral spinal fluid (CSF) material properties.

property		CGS	MKS
density	ρ_0	1.004 g/cc	1004 kg/m ³
P-wave sound speed	c_L	1.537e5 cm/s	1.400e3 m/s
bulk modulus	K	2.371e10 dyne/cm ²	2.371e9 Pa
shear modulus	G	5.000e3 dyne/cm ²	5.000e2 Pa
elastic modulus	E	1.50e4 dyne/cm ²	1.50e3 Pa
Poisson ratio	ν	0.49999989	same

Here we match the CSF bulk modulus K to the bulk modulus of white and gray matter. We set the CSF shear modulus G to 500 Pa. These two parameters are based on rationale set forth in Zhang [Zhang et al., 2001], at page 24,

Cerebrospinal [fluid] (CSF) is known to be a Newtonian biological fluid that fills the subarachnoid space and ventricular system. Its density and viscosity are close to that of water (Ommaya, 1968; Goldsmith, 1971). Ideally, fluid elements should be used to model the CSF, while solid elements are more suitable for the rest of the brain. However, the true coupling of the Newtonian fluid formulation with the Lagrangian solid formulation has not [been] implemented as [of] yet. In the current model, the CSF was represented by a layer of brick elements of solid material having a density of 1,004 kg/m³. A bulk modulus of 2.19 GPa [the same bulk modulus used to model white and gray matter] and a very [low] shear modulus of 500 Pa were assumed for the CSF to accommodate the pressure gradient and relative movement of the brain occurring between the brain and the dura mater.

¹⁴Internal reference: ../material/csf.tex.

D.2.7. White Matter

We model white matter with the following material properties:¹⁵

Table D-7. White matter material properties.

property		CGS	MKS
density	ρ_0	1.040 g/cc	1040 kg/m ³
bulk modulus	K	2.371e10 dyne/cm ²	2.371e9 Pa
shear input	$A_1 \leftrightarrow G_0$	4.10e5 dyne/cm ²	4.10e4 Pa
shear input	P_1	0.0	0.0
shear input	B_1	0.0	0.0
shear input	Q_1	0.0	0.0
shear input	C_1	0.0	0.0
shear input	R_1	0.0	0.0
cut off strain	e_c	0.05	0.05
prony shear ∞	\hat{G}_∞	1.9024e-1 Pa/Pa	—
prony shear 1	\hat{G}_1	8.0976e-1 Pa/Pa	—
prony shear 2...10	$\hat{G}_{i=2\dots 10}$	1.0000e-4 Pa/Pa	—
shear relax time 1	$\tau_{1,\text{old}}$	1.4286e-3 s	—
	$\beta_{1,\text{old}}$	700 s ⁻¹	—
shear relax time 1	$\tau_{1,\text{new}}$	2.50e-2 s	—
	$\beta_{1,\text{new}}$	40 s ⁻¹	—
shear relax time 2	τ_2	100	—
shear relax time 3	τ_3	150	—
shear relax time 4	τ_4	200	—
shear relax time 5	τ_5	250	—
shear relax time 6	τ_6	300	—
shear relax time 7	τ_7	350	—
shear relax time 8	τ_8	400	—
shear relax time 9	τ_9	450	—
shear relax time 10	τ_{10}	500	—
wlf coef c1	c_1	1.0	—
wlf coef c2	c_2	1.0	—
wlf tref		298 K	—
max Poisson's ratio	ν_{max}	0.49	—

¹⁵Internal reference: ../material/whitematter.tex.

D.2.8. Gray Matter

We model gray matter with the following material properties.¹⁶

Table D-8. Gray matter material properties.

property		GCS	MKS
density	ρ_0	1.040 g/cc	1040 kg/m ³
bulk modulus	K	2.371e10 dyne/cm ²	2.371e9 Pa
shear input	$A_1 \leftrightarrow G_0$	3.40e5 dyne/cm ²	3.40e4 Pa
shear input	P_1	0.0	0.0
shear input	B_1	0.0	0.0
shear input	Q_1	0.0	0.0
shear input	C_1	0.0	0.0
shear input	R_1	0.0	0.0
cut off strain	e_c	0.05	0.05
prony shear ∞	\hat{G}_∞	1.8824e-1 Pa/Pa	—
prony shear 1	\hat{G}_1	8.1176e-1 Pa/Pa	—
prony shear 2...10	$\hat{G}_{i=2\dots 10}$	1.0000e-4 Pa/Pa	—
shear relax time 1	$\tau_{1,\text{old}}$	1.4286e-3 s	—
	$\beta_{1,\text{old}}$	700 s ⁻¹	—
shear relax time 1	$\tau_{1,\text{new}}$	2.50e-2 s	—
	$\beta_{1,\text{new}}$	40 s ⁻¹	—
shear relax time 2	τ_2	100	—
shear relax time 3	τ_3	150	—
shear relax time 4	τ_4	200	—
shear relax time 5	τ_5	250	—
shear relax time 6	τ_6	300	—
shear relax time 7	τ_7	350	—
shear relax time 8	τ_8	400	—
shear relax time 9	τ_9	450	—
shear relax time 10	τ_{10}	500	—
wlf coef c1	c_1	1.0	—
wlf coef c2	c_2	1.0	—
wlf tref		298 K	—
max Poisson's ratio	ν_{max}	0.49	—

¹⁶Internal reference: ../material/graymatter.tex.

D.2.9. Muscle

We model muscle with the following material properties.¹⁷

Table D-9. Muscle material properties.

property		CSG	MKS
density	ρ_0	1.200 g/cc	1200 kg/m ³
P-wave sound speed	c_L	1.885e4 cm/s	1.885e2 m/s
elastic modulus	E	1.67e8 dyne/cm ²	1.67e7 Pa
Poisson ratio	ν	0.42	same
bulk modulus	K	3.479e8 dyne/cm ²	3.479e7 Pa
shear modulus	G	5.880e7 dyne/cm ²	5.880e6 Pa

D.2.10. Skin

Skin is modeled identically as muscle, consistent with Taylor [Taylor et al., 2018].

¹⁷Internal reference: material/doc/muscle.tex.

D.2.11. Helmet Shell

We model the helmet shell with the following material properties:¹⁸

Table D-10. Helmet shell material properties.

property		CGS	MSK
density	ρ_0	1.440 g/cc	1440 kg/m ³
P-wave sound speed	c_L	9.669e5 cm/s	9.669e3 m/s
elastic modulus	E	1.0e12 dyne/cm ²	1.0e11 Pa
Poisson ratio	ν	0.3	same
yield stress	Y_0	8.2e9 dyne/cm ²	8.9e8 Pa
hardening modulus	H	1.36e10 dyne/cm ²	1.36e9 Pa
iso/kine hardening	β	0.5	same

Moss and King [Moss and King, 2011] state their Kevlar material properties are “from the literature [5]” where

- “[5]” = M. Aare and S. Kleiven, Evaluation of head response to ballistic helmet impacts using the finite element method, *Intl J Impact Egr* 34(3):596-608, 2007.
- Chang-Chang composite failure model, material type 22 in LS-DYNA, with material properties from [7,8], and failure criteria for matrix cracking, compression failure, and fiber breakage.
- Total helmet mass = 1.5 kg.
- Isotropic formulation, since impact is likely essentially through isotropic crown of helmet.
- Page 11: “Material properties for the Kevlar helmet shell were obtained from the literature [5]. Although the helmet shell is anisotropic, our simulations showed no substantial difference in the impact response between isotropic and anisotropic representations of the helmet. Consequently, all the simulations discussed in this report treat the Kevlar shell as an isotropic material, which simplified subsequent analyses of the effects of impact-induced plastic deformation of the Kevlar.”

¹⁸Internal reference: ../material/shell.tex.

D.2.12. Foam

The foam used in the helmeted simulations and associated lab testing is a formulation developed by Team Wendy. Because our Team Wendy collaborators labeled the foam response experiments as “Team Wendy Proprietary,” we do not explicitly expose the material’s mechanical performance characteristics, beyond our material models contained in our SSM input decks.¹⁹ Note that the constitutive model utilized to capture the deviatoric behavior of the foam is an orthotropic crush model. The orthotropic crush model was initially developed to model Aluminum Honeycomb, but was able to be adapted for the current purposes. The crush model includes several discrete response regimes, including: (1) a linear elastic region, (2) linear elastic-plastic pore crushing region, (3) non-linear plastic crush up region. These regions allowed the numerical model to capture the full spectrum of the foam’s mechanical response to various loading states.

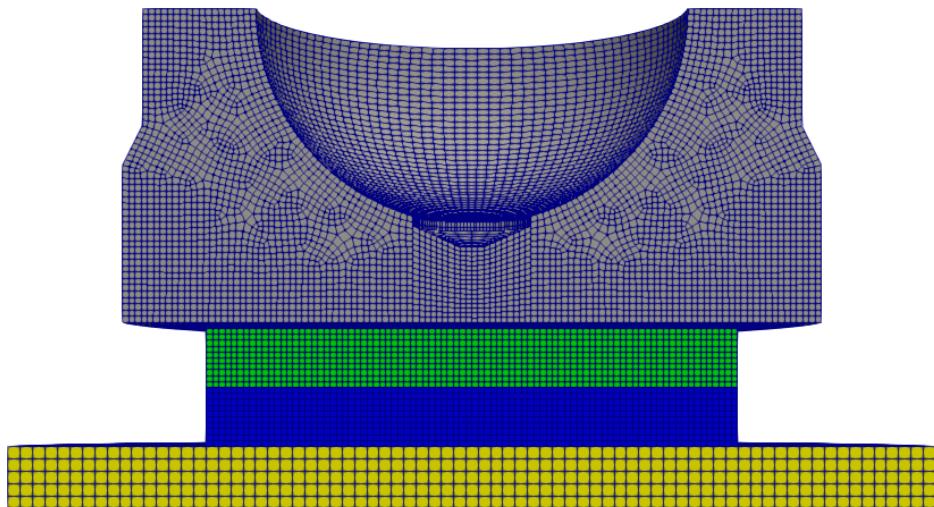


Figure D-4. Half-symmetry view of model used for foam compression material validation.

¹⁹Internal reference: .../material/foam.tex.



Figure D-5. Team Wendy helmet pad used in compression tests.



Figure D-6. Test configuration for compression of the Team Wendy helmet pad.

D.2.13. Steel

We model the steel hemispherical impact with the following material properties:

Table D-11. Steel material properties.

property		CGS	MKS
density	ρ_0	8.0 g/cc	8000 kg/m ³
elastic modulus	E	1.930e12 dyne/cm ²	1.930e11 Pa
Poisson ratio	ν	0.29	same

- Moss and King [Moss and King, 2011] specified the boundary conditions for the anvil, “The base of the steel anvil is constrained so that it cannot move.”
- However, Moss and King did not specify the material properties they used for the anvil.

APPENDIX E. Methods

E.1. Quasi-Rigid Body Dynamics

In this section, we describe the methodology used to calculate quasi-rigid body rotations from deformable bodies undergoing large rotations and small local deformations.¹

E.1.1. Formulation

Cross-Product Operator

One special form of a skew-symmetry operator is a **cross-product operator**, defined as

$$[\mathbf{a}]_{\times} \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \quad (\text{E.1})$$

Thus $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b}$.

Rotation and Stretch of a Deformable Body

Result E.1.1 The angular velocity $\boldsymbol{\omega}$ of a body \tilde{B} that undergoes finite rotations and infinitesimal deformations may be approximated as a least squares projection of the body's configuration column space onto the body's velocity space as

$$\boldsymbol{\omega} = - \left(\mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{A}^T \cdot \{ \mathbf{v}_P - \mathbf{v}_Q \}, \text{ where} \quad (\text{E.2})$$

$$\mathbf{A} = [\mathbf{x}_P - \mathbf{x}_Q]_{\times}. \quad (\text{E.3})$$

Proof.

Let body \tilde{B} undergo finite rotations and infinitesimal deformations in reference frame F . Using the velocity of two points Q and P , with a particle moving on rigid body B ,

$$\mathbf{v}_{P/F} = \mathbf{v}_{Q/F} + \mathbf{v}_{P/Q} + \boldsymbol{\omega}_{B/F} \times \mathbf{r}_{QP} \quad (\text{E.4})$$

$$\mathbf{v}_P = \mathbf{v}_Q + \mathbf{v}_{P/Q} + \boldsymbol{\omega}_B \times \mathbf{r}_{QP} \text{ dropping } F \text{ where implied.} \quad (\text{E.5})$$

¹Reference: ../method/n_point_rotation.tex

Note the term $\mathbf{v}_{P/Q}$ is the *relative velocity* between points P and Q . Assuming infinitesimal deformations, it is reasonable to assume the time derivatives of these quantities are also small, thus $\mathbf{v}_{P/Q} \approx \mathbf{0}$. Thus for body \tilde{B}

$$\mathbf{v}_P - \mathbf{v}_Q \approx \boldsymbol{\omega}_B \times \mathbf{r}_{QP}, \quad (\text{E.6})$$

$$= -\mathbf{r}_{QP} \times \boldsymbol{\omega} \quad (\text{E.7})$$

$$= -(\varphi(\mathbf{X}_P, t) - \varphi(\mathbf{X}_Q, t)) \times \boldsymbol{\omega} \quad (\text{E.8})$$

$$= -(\mathbf{X}_P + \mathbf{u}(\mathbf{X}_P, t) - (\mathbf{X}_Q + \mathbf{u}(\mathbf{X}_Q, t))) \times \boldsymbol{\omega} \quad (\text{E.9})$$

if displacements \mathbf{u} are used; or,

$$= -(\mathbf{x}_P - \mathbf{x}_Q) \times \boldsymbol{\omega} \quad \text{if current placements } \mathbf{x} \text{ are used,} \quad (\text{E.10})$$

$$= -[\mathbf{x}_P - \mathbf{x}_Q]_{\times} \boldsymbol{\omega}, \quad (\text{E.11})$$

where the skew-symmetric matrix, defined in Eq. (E.1), is used. Explicitly, this is

$$[\mathbf{x}_P - \mathbf{x}_Q]_{\times} = \begin{bmatrix} 0 & -(x_{P3} - x_{Q3}) & (x_{P2} - x_{Q2}) \\ (x_{P3} - x_{Q3}) & 0 & -(x_{P1} - x_{Q1}) \\ -(x_{P2} - x_{Q2}) & (x_{P1} - x_{Q1}) & 0 \end{bmatrix}, \quad (\text{E.12})$$

which is solvable through a least-squares approach. ■

E.1.2. Unit Test Validation

$$\ddot{\theta} + \frac{3g}{2L} \sin \theta = 0, \quad (\text{E.13})$$

$$\theta_0 = \frac{\pi}{2}, \quad (\text{E.14})$$

$$\dot{\theta}_0 = 0. \quad (\text{E.15})$$

$$\theta(t) = 2 \sin^{-1} \left[\frac{1}{\sqrt{2}} \operatorname{sn} \left(\sqrt{\frac{3g}{2L}} t, \frac{1}{\sqrt{2}} \right) \right] \quad (\text{E.16})$$

The deformable body pendulum simulation used the following inputs: $\rho_0 = 7.2\text{e}3 \text{ kg/m}^3$, $E = 2.09\text{e}9 \text{ N/m}^2$, $\nu = 0.3$, $\mathbf{r}^{OP} = (1.00, -0.01, -0.01) \text{ m}$, $\mathbf{r}^{OQ} = (1.00, 0.01, -0.01) \text{ m}$, $\mathbf{r}^{OR} = (1.00, -0.01, 0.01) \text{ m}$, $g = 9.81 \text{ m/s}^2$, $L = 1.00 \text{ m}$, and $m = 2.88 \text{ kg}$. The mesh, shown in Figure E-4, had 845 nodal points and 372 hexagonal elements.

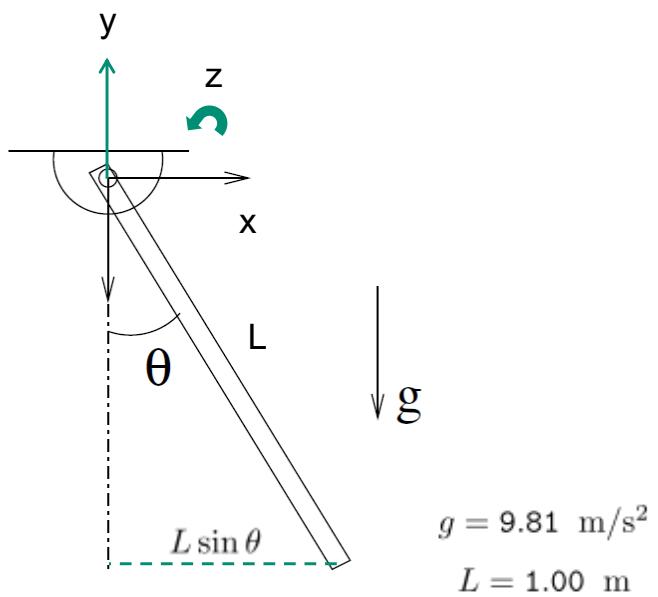


Figure 1. Geometry of the pendulum.

Figure E-1. Reproduction of Figure 1 from [Jog and Motamarri, 2009].

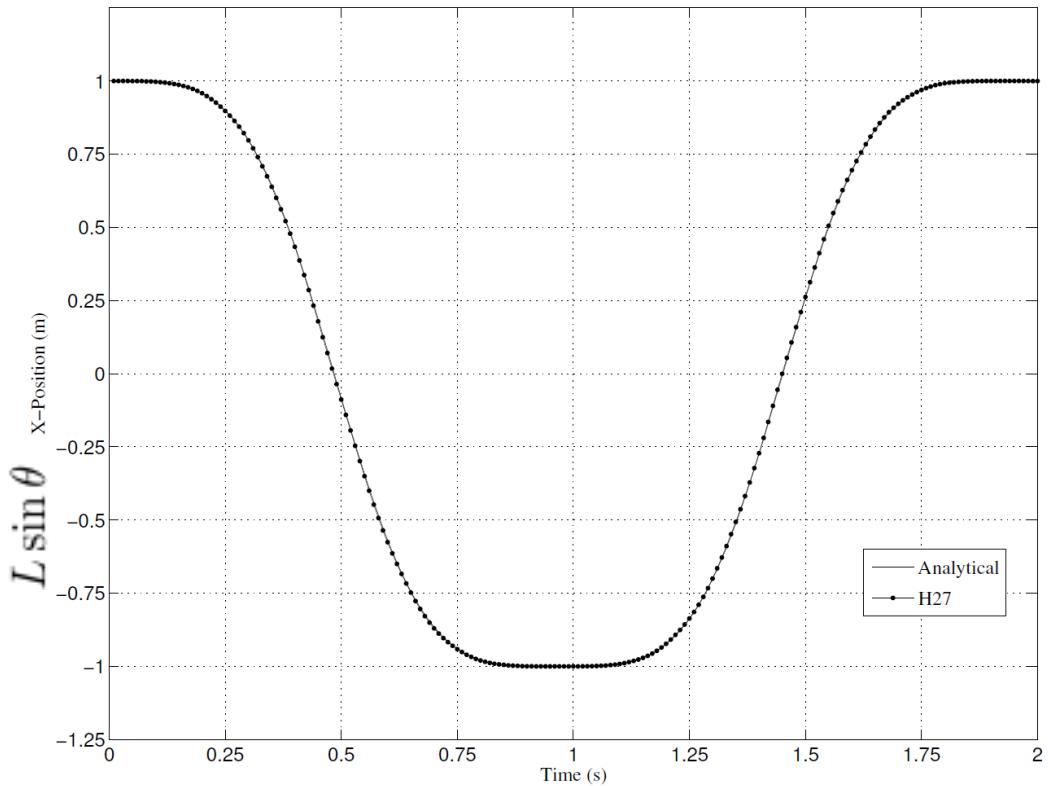


Figure 2. Comparison of the numerically obtained tip displacement of the pendulum with the analytical solution under [Case I](#).

Figure E-2. Reproduction of Figure 2 from [\[Jog and Motamarri, 2009\]](#).

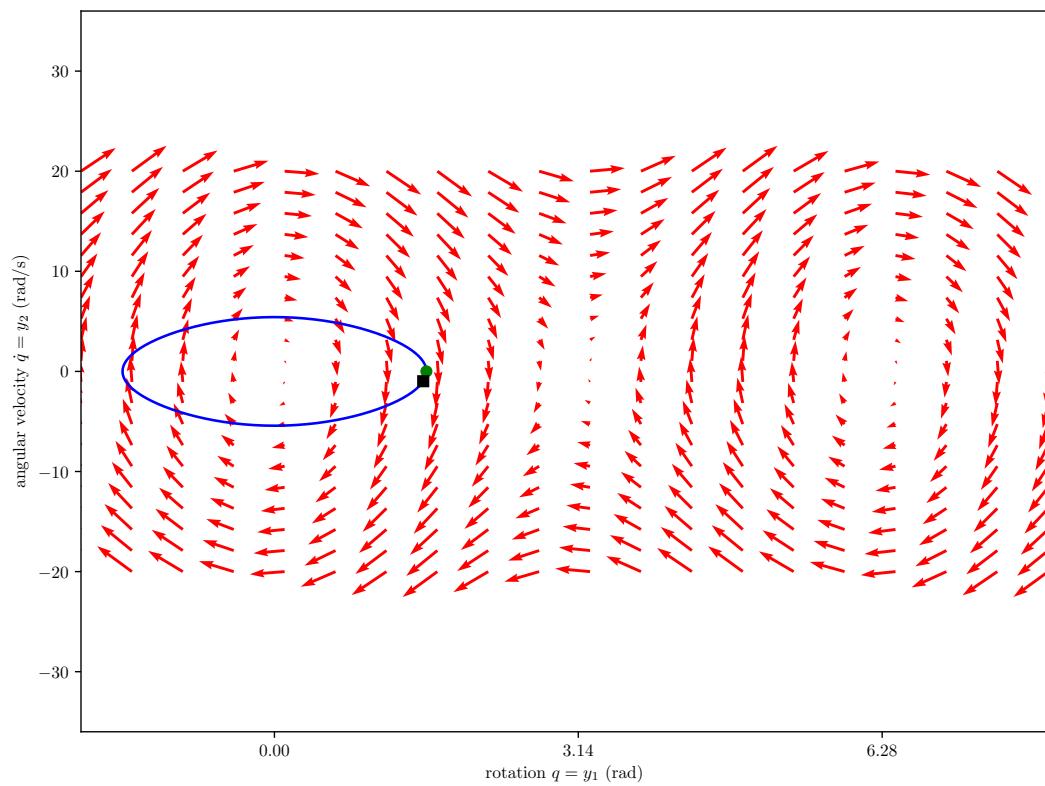


Figure E-3. Rigid body phase diagram reference standard.

origin O [0, 0, 0]

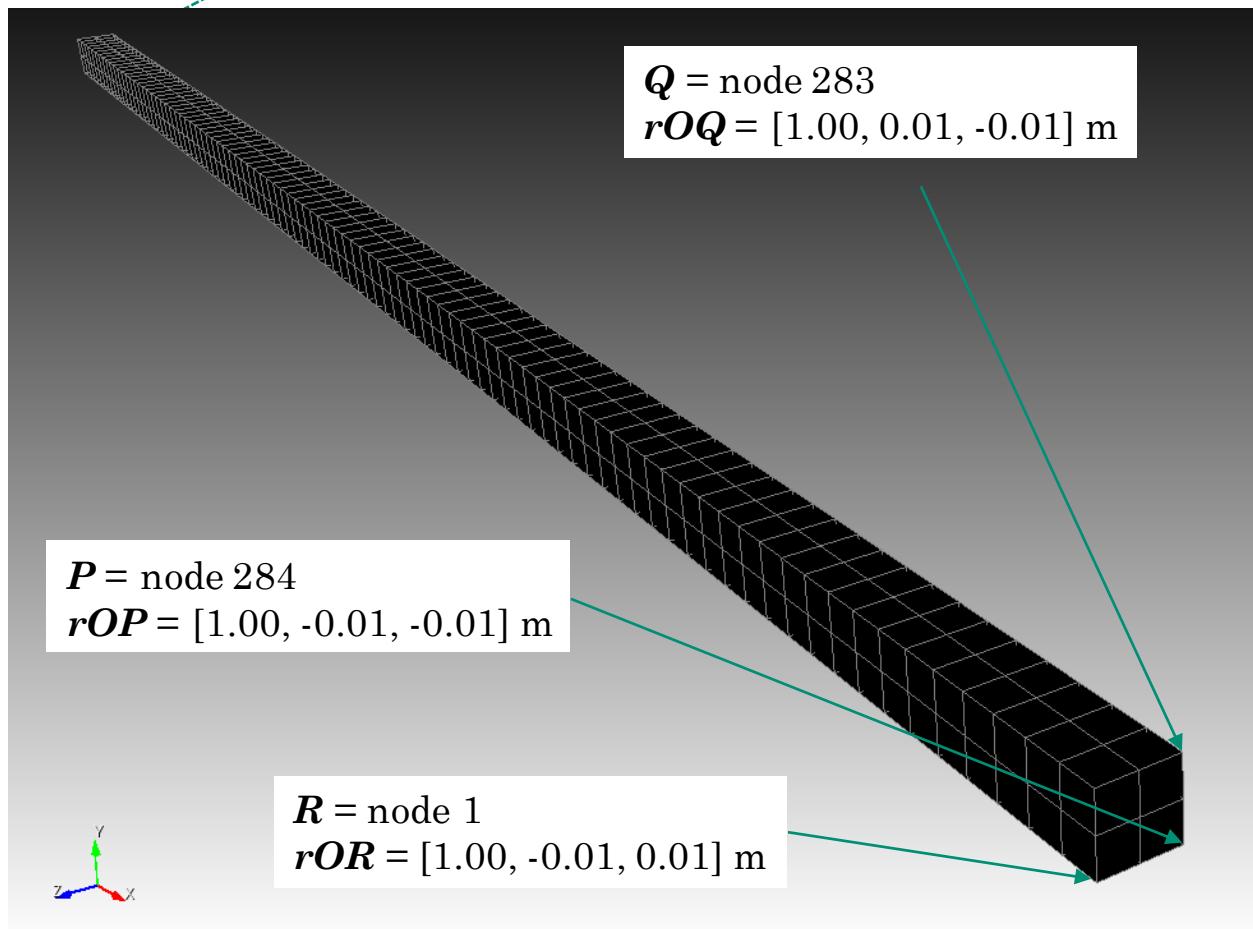


Figure E-4. Finite element mesh of deformable pendulum, used for unit testing of the three point angular velocity algorithm.

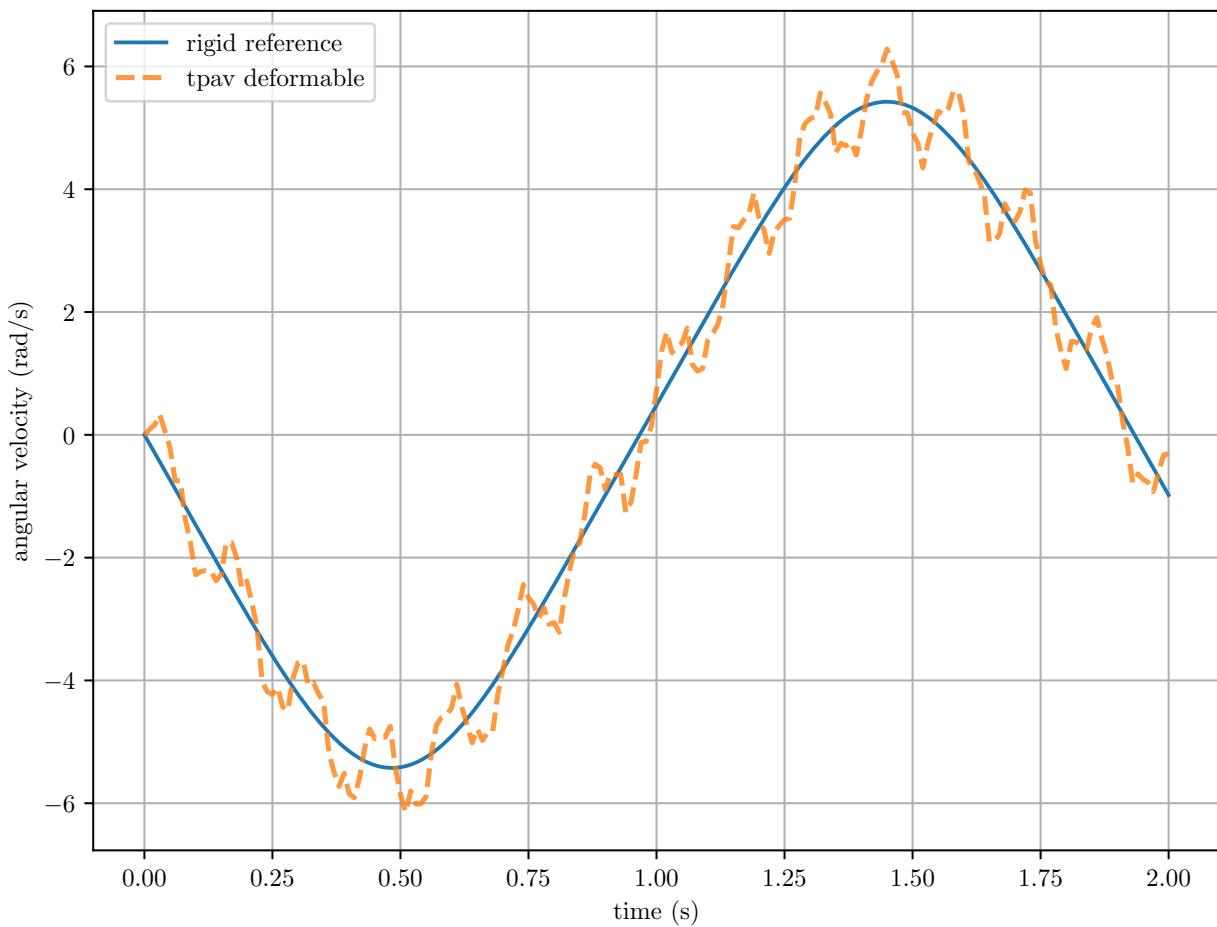


Figure E-5. Comparison of rigid body angular velocity reference to deformable body three point angular velocity algorithm.

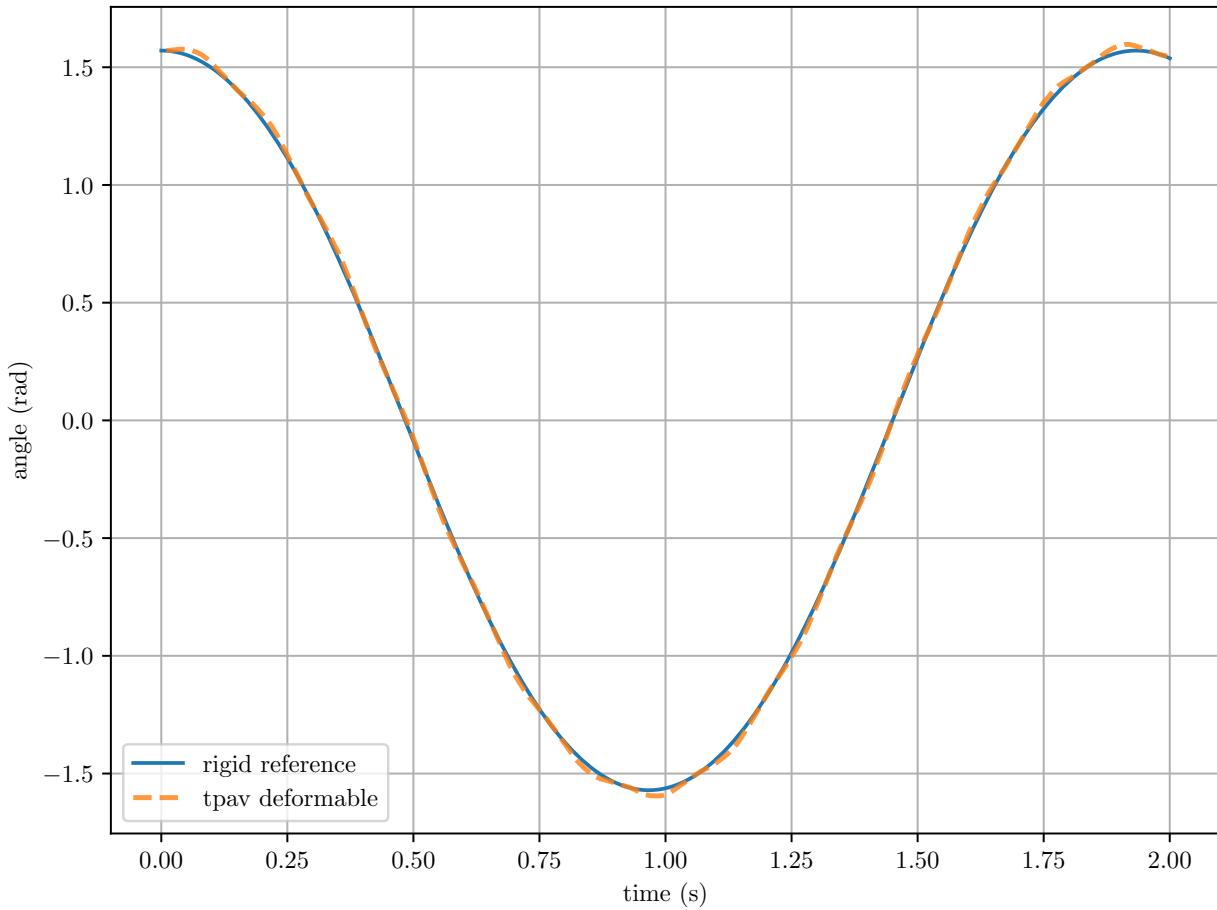


Figure E-6. Comparison of rigid body angular position reference to deformable body three point angular velocity algorithm.

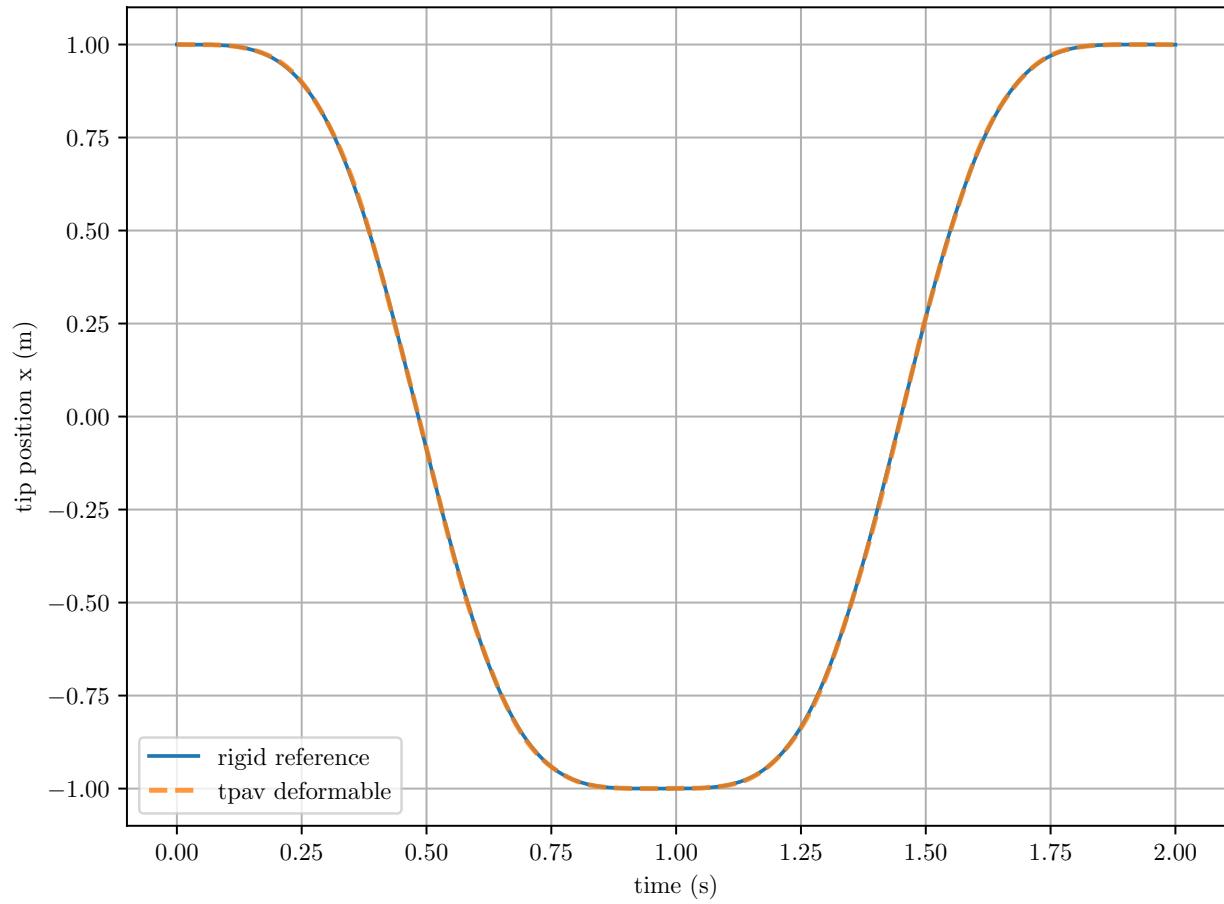


Figure E-7. Comparison of rigid body tip x position reference to deformable body three point angular velocity algorithm.

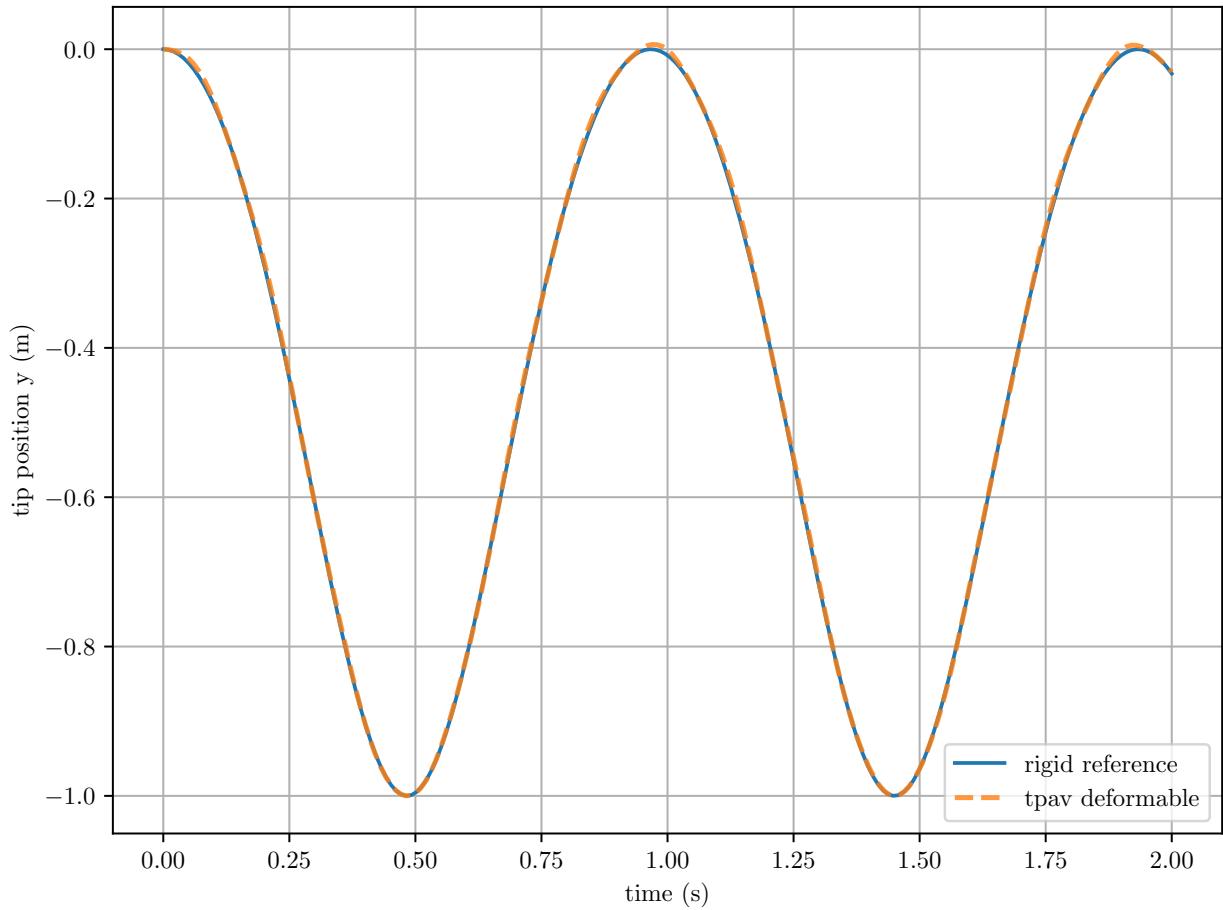


Figure E-8. Comparison of rigid body tip y position reference to deformable body three point angular velocity algorithm.

E.2. Similarity

The purpose of this section is to provide details of our signal similarity implementation, used as an objective measure of framework validation, discussed in Section 1.2.

The cross-correlation is a measure of the *similarity in time* of two time series signals. It is often referred to as the *dot product* of two time series because the cross-correlation operation is a dot production operation.

E.2.1. Vector Dot Product

Consider two vectors \mathbf{a} and \mathbf{b} in three-dimensional space, measured from the same origin and with the same coordinate system, with components,

$$\mathbf{a} = [a_1, a_2, a_3], \quad (\text{E.17})$$

$$\mathbf{b} = [b_1, b_2, b_3]. \quad (\text{E.18})$$

The dot product of the two vectors is simply,

$$\mathbf{a} \cdot \mathbf{b} = a_i b_i = a_1 b_1 + a_2 b_2 + a_3 b_3 \quad (\text{E.19})$$

$$\text{or generally} = \sum_{i=1}^n a_i b_i. \quad (\text{E.20})$$

Next, consider $\mathbf{a}(t)$ and $\mathbf{b}(t)$ be two time-varying signals, occurring over time interval $[t_0, t_f]$. Let the amplitudes of $\mathbf{a}(t)$ and $\mathbf{b}(t)$ be measured from zero to t_f at n equally distributed points such that the signals are sampled at n uniform discrete times in the interval. The two vectors are written explicitly as

$$\mathbf{a} = [a_0, a_1, a_2, \dots, a_{n-3}, a_{n-2}, a_{n-1}], \quad (\text{E.21})$$

$$\mathbf{b} = [b_0, b_1, b_2, \dots, b_{n-3}, b_{n-2}, b_{n-1}]. \quad (\text{E.22})$$

Remark E.2.5. To account for the *initial condition* of the time series, we decrement the counter, $1 \dots n$, to become $0 \dots n - 1$. In both of these cases, the number of sample points is identical, but the indexing is different by one.

Remark E.2.6. Notice that the two signals $\mathbf{a}(t)$ and $\mathbf{b}(t)$ are *synchronized* in time such that

$$\mathbf{t} = [t_0, t_1, t_2, \dots, t_{n-3}, t_{n-2}, t_{n-1}]. \quad (\text{E.23})$$

Notice that $\Delta t = t_1 - t_0 = t_2 - t_1 = \dots = t_{n-1} - t_{n-2}$ and that the signals are sampled at a frequency of $\frac{1}{\Delta t}$ Hz.

E.2.2. Series Dot Product

The dot product of these two signals is

$$\mathbf{a} \cdot \mathbf{b} = a_i b_i = a_0 b_0 + a_1 b_1 + a_2 b_2 + \cdots + a_{n-1} b_{n-1} \quad (\text{E.24})$$

$$\text{or generally } = \sum_{i=0}^{n-1} a_i b_i. \quad (\text{E.25})$$

For comparison of two time series, the dot product will be referred to as the **cross-correlation score**.

Example 2.

This example is a reproduction of the cross-correlation example on the Anomali site.² Let vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} be defined as follows,

$$\mathbf{a} = [1, 2, -2, 4, 2, 3, 1, 0] \quad (\text{E.26})$$

$$\mathbf{b} = [2, 3, -2, 3, 2, 4, 1, -1] \quad (\text{E.27})$$

$$\mathbf{c} = [-2, 0, 4, 0, 1, 1, 0, -2] \quad (\text{E.28})$$

Then the cross-correlation score between \mathbf{a} and \mathbf{b} is found to be high,

$$\mathbf{a} \cdot \mathbf{b} = 41; \quad (\text{E.29})$$

whereas, the cross-correlation score between \mathbf{a} and \mathbf{c} is found to be low,

$$\mathbf{a} \cdot \mathbf{c} = -5. \quad (\text{E.30})$$

...

²See <https://anomaly.io/understand-auto-cross-correlation-normalized-shift/>

E.2.3. Limitation of Cross-Correlation

A significant limitation of the cross-correlation method is its dependence on *amplitude*:

- Large positive magnitudes increase the cross-correlation.
- Large negative magnitudes decrease the cross-correlation.
- A cross-correlation score shouldn't get better (positive) or worse (negative) if the amplitude of the signal is increased or decreased.
- Shape similarity is obfuscated by amplitude dependence. For example, the three cross-correlation scores generated from comparison of three signals, $A \sin(t)$, $B \sin(t)$, and $C \sin(t)$, all of the same shape, $\sin(t)$, but with unique amplitudes A, B, C , will report that two out of the three signals are better correlated by virtue of their closer amplitudes. In a sense, this result is satisfactory, since amplitude distinguishes one signal from another. However, this result is incomplete, since the amplitude dependence obfuscates the fact that all three signals have the same basic shape.
- Shape similarity obfuscation is easily seen with shape *self*-similarity, that is, using the *same* signal for both vectors in the cross-correlation. In this case

$$\mathbf{a} \cdot \mathbf{a} > \mathbf{a} \cdot \frac{\mathbf{a}}{2}, \quad (\text{E.31})$$

which is a sensible outcome, but not helpful to distinguish shape similarity.

To remedy the shortcoming of cross-correlation caused by amplitude dependence, **vector normalization** is employed, as explained in a following section. Prior to this, however, we make a brief introduction to the p-norm.

E.2.4. p-norm

For any number $p \in \mathbb{R}$ and $p \geq 1$, the **p-norm** is defined as

$$\| \mathbf{a} \|_p \triangleq \left(\sum_{i=0}^{n-1} |a_i|^p \right)^{\frac{1}{p}}. \quad (\text{E.32})$$

Example 3.

The L2-norm, also called the **Euclidean norm**, is defined as

$$\| \mathbf{a} \|_2 \triangleq \left(\sum_{i=0}^{n-1} |a_i|^2 \right)^{\frac{1}{2}}, \quad (\text{E.33})$$

and has the physical interpretation as the length of the vector on which the L2-norm operates.

The L1-norm, also called the **Manhattan taxicab norm**, is defined as

$$\| \mathbf{a} \|_1 \triangleq \left(\sum_{i=0}^{n-1} | a_i | \right), \quad (\text{E.34})$$

and has the physical interpretation as the cumulative length (mileage) accumulated for each segment a_i .

The **absolute value** is the L1-norm on a one-dimensional vector space,

$$\| \mathbf{a} \|_1 \triangleq \left(\sum_{i=0}^0 | a_i | \right) = | a_0 |, \quad (\text{E.35})$$

...

E.2.5. Normalized Cross-Correlation

When vector normalization is used with cross-correlation, the resulting method is referred to as *normalized cross-correlation*. Normalized cross-correlation is the same as cross-correlation with one exception: The vectors used in the dot product are normalized by their respective L2-norm.

The normalized cross-correlation for the three-dimensional vectors \mathbf{a} and \mathbf{b} is defined as

$$\frac{\mathbf{a}}{\| \mathbf{a} \|_2} \cdot \frac{\mathbf{b}}{\| \mathbf{b} \|_2} \triangleq \frac{a_i}{\sqrt{\sum_{i=1}^n a_i a_i}} \cdot \frac{b_i}{\sqrt{\sum_{i=1}^n b_i b_i}} \quad (\text{E.36})$$

$$= \frac{a_1 b_1 + a_2 b_2 + a_3 b_3}{\left(\sqrt{a_1^2 + a_2^2 + a_3^2} \right) \left(\sqrt{b_1^2 + b_2^2 + b_3^2} \right)}. \quad (\text{E.37})$$

The foregoing result generalizes to time series $\mathbf{a}(t)$ and $\mathbf{b}(t)$ as

$$\frac{\mathbf{a}}{\| \mathbf{a} \|} \cdot \frac{\mathbf{b}}{\| \mathbf{b} \|} \triangleq \frac{\sum_{i=0}^{n-1} a_i b_i}{\left(\sqrt{\sum_{i=0}^{n-1} a_i^2} \right) \left(\sqrt{\sum_{i=0}^{n-1} b_i^2} \right)} \quad (\text{E.38})$$

Consider again the vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} from Example 1.

Example 4.

Consider again the vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} from Example 1. The *normalized* cross-correlation score between \mathbf{a} and \mathbf{b} is found to be high (relative to a maximum of 1.0),

$$\frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|} = \frac{41}{\sqrt{39} \sqrt{48}} = 0.947; \quad (\text{E.39})$$

whereas, the *normalized* cross-correlation score between \mathbf{a} and \mathbf{c} is found to be low (relative to no correlation for 0.0),

$$\frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \frac{\mathbf{c}}{\|\mathbf{c}\|} = \frac{-5}{\sqrt{39} \sqrt{26}} = -0.157. \quad (\text{E.40})$$

...

The **benefits** of *normalized* cross-correlation are two-fold: Amplitude independence and unit normal scoring. Amplitude independence is seen as

$$\mathbf{a} \cdot \mathbf{a} = \mathbf{a} \cdot \frac{\mathbf{a}}{2}. \quad (\text{E.41})$$

The normalized cross-correlation score now ranges between -1 and +1.

- +1 is perfect correlation,
- 0 is no correlation, and
- -1 is anti-correlation (correlated, but in opposite phase).

E.2.6. Normalized Cross-Correlation with Time Shift

The motivation for **normalization** was to remove amplitude effects from the cross-correlation score. These effects are seen along the y -axis. A complementary effect, called **phase effect**, can exist along the x -axis.

Specifically, two time series may be highly correlated, or even identical, but if out of phase, will cause the cross-correlation and the normalized cross-correlation to report a score of zero.

To remedy this outcome, a second strategy can be formulated. In previous sections, the dot product and the normalized dot product were used for cross-correlation and normalized cross-correlation, respectively. Now, we will shift one of the two signals incrementally in time to both the left and the right, to create what is referred to as the **sliding dot product** and **sliding normalized dot product**, respectively.

Conceptually, the sliding dot product strategy computes the dot product between two time series for all combination of a_i and b_i to determine if a **phase lead** or **phase lag** results in a better correlation score.

The result of the sliding dot product is an expansion of inner products, called **cross-correlation coefficients**, which will be labeled as vector \mathbf{c} , with dimension $2n - 1$, generated by the sliding mechanism. The sliding dot product is illustrated in matrix form, where time series \mathbf{b} slides forward and backward incrementally in time, relative to time series \mathbf{a} . This operation takes the explicit form,

$$\begin{array}{c}
 \text{initial condition, len = 1} \\
 \hline
 \text{slide-forward} \\
 \text{operation} \\
 \vdots \\
 \hline
 \text{len = } n - 1 \\
 \text{slide-backward} \\
 \text{operation} \\
 \vdots \\
 \hline
 \text{len = } n - 1
 \end{array}
 \left\{ \begin{array}{c} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_{-1} \\ c_{-2} \\ \vdots \\ c_{-(n-1)} \end{array} \right\}_{(2n-1 \times 1)} = \\
 \left[\begin{array}{ccccccc} b_0 & b_1 & b_2 & \cdots & b_{n-3} & b_{n-2} & b_{n-1} \\ 0 & b_0 & b_1 & b_2 & \cdots & b_{n-3} & b_{n-2} \\ 0 & 0 & b_0 & b_1 & b_2 & \cdots & b_{n-3} \\ & & \ddots & & & & \vdots \\ & & & \ddots & & & \vdots \\ & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & b_0 \\ b_1 & b_2 & \cdots & b_{n-3} & b_{n-2} & b_{n-1} & 0 \\ b_2 & \cdots & b_{n-3} & b_{n-2} & b_{n-1} & 0 & 0 \\ \vdots & & \ddots & & & & \\ \vdots & & & \ddots & & & \\ b_{n-1} & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]_{(2n-1 \times n)} \left\{ \begin{array}{c} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-3} \\ a_{n-2} \\ a_{n-1} \end{array} \right\}_{(n \times 1)} \quad (\text{E.42})$$

The foregoing matrix form can be reordered row-wise, from smallest to largest value of j , and coded in index form

$$c_j = \sum_{i=0}^{n-1} b_{i-j} a_i, \quad \text{with } j = -(n-1) \dots (n-1). \quad (\text{E.43})$$

The following table provides five samples points for j verify the formula in (E.43).

Example 5.

Table E-1. Five-point verification of sliding dot product form (E.43).

j	c
$-(n - 1)$	$\sum_{i=0}^{n-1} b_{i+(n-1)} a_i = b_{n-1} a_0$
...	...
-1	$\sum_{i=0}^{n-1} b_{i+1} a_i = b_1 a_0 + b_2 a_1 + \dots + b_{n-1} a_{n-2}$
0	$\sum_{i=0}^{n-1} b_i a_i = b_0 a_0 + b_1 a_1 + \dots + b_{n-1} a_{n-1}$
1	$\sum_{i=0}^{n-1} b_{i-1} a_i = b_0 a_1 + b_1 a_2 + \dots + b_{n-2} a_{n-1}$
...	...
$(n - 1)$	$\sum_{i=0}^{n-1} b_{i-(n-1)} a_i = b_0 a_{n-1}$

This example is a reproduction of the normalized cross-correlation with time shift example from the Anomaly site.³ Let vectors \mathbf{a} and \mathbf{b} be defined as follows,

$$\mathbf{a} = [0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4] \quad (\text{E.44})$$

$$\mathbf{b} = [1, 2, 3, 3, 0, 1, 2, 3, 4, 0, 1, 1, 4, 4, 0, 1, 2, 3, 4, 0] \quad (\text{E.45})$$

where slight spacing has been interjected between every fifth element, to help illustrate the pattern embedded in the series. Since the dimension, n , of the signal is 20, there are $2n - 1 = 39$ correlation coefficients, c_i , that can be calculated from the normalized cross-correlation with time shift operation. Below are 9 of the 39 correlation coefficients, showing the initial condition, the first four slide-forward operations, and the first four slide-backward operations.

Δt	-4	-3	-2	-1	0	1	2	3	4
c_i	0.862	0.021	-0.547	-0.423	0.000	0.867	0.127	-0.466	-0.393

The results indicate that sliding the \mathbf{b} signal forward by one time increment results in the largest normalized cross-correlation with time shift correlation coefficient ($c_1 = 0.867$ at $\Delta t = 1$). This correlation can be seen through visual inspection of the two curves at zero time shift, and then shifting signal \mathbf{b} forward by one step.

...

³See <https://anomaly.io/understand-auto-cross-correlation-normalized-shift/>

E.2.7. From Deterministic to Stochastic

In this section, we apply some of the notions of expectation, variance, and covariance that we noted in the first sections with the notions of cross-correlation. Up until this point, the signals used in the cross-correlation discussion were tacitly normalized to the origin at $\mathbf{0}$. So, expectations and deviations were tacitly taken with respect to zero for all times in signals a_i and b_i .

Now, we note the form for correlation applied to stochastic processes (the former will more or less be called a deterministic process, by contrast). In the stochastic process, we will normalize data with respect to an expected value (e.g., mean, in the case of equal likelihood probabilities) as an inhomogeneous datum (different from zero). The mean *could* be zero as a trivial special case, but in general the mean is non-zero.

Stochastic Vector Dot Product

Generally, we substitute $(a_i - \bar{a})$ for (a_i) in the previous definition of vector dot product, appearing in Section E.2.1. The value \bar{a} is the expected value of \mathbf{a} , and in the special case of equal likelihood, \bar{a} is the average value μ_a .

Then, the stochastic vector dot product of three-dimensional signals \mathbf{a} and \mathbf{b} is defined as,

$$\mathbf{a} \cdot \mathbf{b} = (a_i - \bar{a}) (b_i - \bar{b}) \quad (\text{E.46})$$

$$= (a_1 - \bar{a}) (b_1 - \bar{b}) + (a_2 - \bar{a}) (b_2 - \bar{b}) + (a_3 - \bar{a}) (b_3 - \bar{b}) \quad (\text{E.47})$$

$$\text{or generally } = \sum_{i=1}^n (a_i - \bar{a}) (b_i - \bar{b}). \quad (\text{E.48})$$

Stochastic Series Dot Product

Extending this to n -dimensional signals $\mathbf{a}(t)$ and $\mathbf{b}(t)$, the stochastic series dot product becomes,

$$\mathbf{a} \cdot \mathbf{b} = (a_i - \bar{a}) (b_i - \bar{b}) \quad (\text{E.49})$$

$$= (a_0 - \bar{a}) (b_0 - \bar{b}) + (a_1 - \bar{a}) (b_1 - \bar{b}) + (a_2 - \bar{a}) (b_2 - \bar{b}) + \cdots + (a_{n-1} - \bar{a}) (b_{n-1} - \bar{b}) \quad (\text{E.50})$$

$$\text{or generally } = \sum_{i=0}^{n-1} (a_i - \bar{a}) (b_i - \bar{b}). \quad (\text{E.51})$$

Stochastic Series Normalized Dot Product

The normalization of the stochastic series dot product arises by comparison of the expected value to its variance. Noting, with the special case of equal likelihood,

$$\sigma^2 = \frac{1}{n} \sum_{i=0}^{n-1} (a_i - \bar{a})^2, \quad (\text{E.52})$$

then,

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (a_i - \bar{a})^2}. \quad (\text{E.53})$$

This allows the stochastic series normalized dot product to be written as

$$\frac{\mathbf{a}}{\|\mathbf{a}\|_2} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|_2} \stackrel{\Delta}{=} \frac{\frac{1}{n} \sum_{i=0}^{n-1} (a_i - \bar{a})(b_i - \bar{b})}{\left(\sqrt{\sum_{i=0}^{n-1} (a_i - \bar{a})^2} \right) \left(\sqrt{\sum_{i=0}^{n-1} (b_i - \bar{b})^2} \right)}. \quad (\text{E.54})$$

Some statistical references call this the **zero-normalized cross-correlation**.

Remark E.2.7. The factor of $\frac{1}{n}$ in (E.54), which arises from the product of two square root variance terms in the denominator, indicates the error *rate*. The error rate, defined in Section E.3.3, reports error on a per-sample basis.

E.3. Error and Error Rate

E.3.1. Error

The **error** $\mathbf{e}(t)$ between two signals $\mathbf{a}(t)$ and $\mathbf{b}(t)$ is defined as the difference between the two signals,

$$\mathbf{e}(t) \triangleq |\mathbf{a}(t) - \mathbf{b}(t)|. \quad (\text{E.55})$$

Note that $\mathbf{e} \geq \mathbf{0} \ \forall t$. If \mathbf{b} exactly follows \mathbf{a} at every time step, then the error is zero, as expected.

Example 6.

Let vectors \mathbf{a} and \mathbf{b} be defined as follows,

$$\mathbf{a} = [20, 20, 20, 20] \quad (\text{E.56})$$

$$\mathbf{b} = [18, 18, 18, 18] \quad (\text{E.57})$$

Then the error \mathbf{e} is found to be,

$$\mathbf{e} = [2, 2, 2, 2] \quad (\text{E.58})$$

In this example, assuming \mathbf{a} is the ground truth, \mathbf{b} underestimates \mathbf{a} by 2 at every sample. This is an example of **systemic error**, where there is a consistent offset between the two signals, and an offset adjustment (the offset value of 2 in this case) would eliminate all error between the two signals.

...

E.3.2. Accumulated Error

Next, we would like to have a scalar value to describe the error over the entire sample, which is measured at each sample point and has dimension n .

We define the **L1-norm accumulated error**, $\|\mathbf{e}\|_1$, as the sum of the errors of each of the measurement,

$$\|\mathbf{e}\|_1 \triangleq \sum_{i=0}^{n-1} |e_i| = \sum_{i=0}^{n-1} |a_i - b_i|. \quad (\text{E.59})$$

This is the L1-norm, applied to each error measurement, $e_i = a_i - b_i$. A serious problem with accumulated error is that the error increases the more times the signals are measured.

Example 7.

- In the preceding example, if the signals were measured at only one sample, the accumulated error would be 2.
- Similarly, if two samples were taken, the accumulated error would be 4.
- For three samples, the accumulated error would be 6.
- For the example as stated above, with four samples, the accumulated error is 8.

This pattern shows the limitation of accumulated error: that it is dependent on sample size.

...

We define the **L2-norm accumulated error**, $\| \mathbf{e} \|_2$, as the sum of the errors of each of

the measurement,

$$\| \mathbf{e} \|_2 \triangleq \left(\sum_{i=0}^{n-1} |e_i|^2 \right)^{\frac{1}{2}} = \left(\sum_{i=0}^{n-1} |a_i - b_i|^2 \right)^{\frac{1}{2}}. \quad (\text{E.60})$$

This is the L2-norm, applied to the error measurement, $e_i = a_i - b_i$. The systemic error seen above for the L1-norm, that the error increases with increasing number of measurements, also occurs for the L2-norm. To remedy this effect, normalization by the sample count, called **error rate**, will be used.

E.3.3. Error Rate

The concept of **error rate** is to normalize the accumulated error for any L_p -norm type by the dimension $n = \text{len}(\mathbf{a}) = \text{len}(\mathbf{b})$ of the signals. The resulting metric then reports error on a *per-measurement basis*, which eliminates the accumulation of error. Specifically, the **L1-norm error rate** is defined as

$$\| \bar{\mathbf{e}} \|_1 \triangleq \frac{1}{n} \sum_{i=0}^{n-1} |e_i| = \frac{1}{n} \sum_{i=0}^{n-1} |a_i - b_i|, \quad (\text{E.61})$$

and the **L2-norm error rate** is defined as

$$\| \bar{\mathbf{e}} \|_2 \triangleq \frac{1}{n} \left(\sum_{i=0}^{n-1} |e_i|^2 \right)^{\frac{1}{2}} = \frac{1}{n} \left(\sum_{i=0}^{n-1} |a_i - b_i|^2 \right)^{\frac{1}{2}}. \quad (\text{E.62})$$

Notice that the error rate units are [amplitude/sample], hence the rate of the errors in the amplitudes.

Example 8.

Revising the example with vectors \mathbf{a} and \mathbf{b} ,

$$\mathbf{a} = [20, 20, 20, 20] \quad (\text{E.63})$$

$$\mathbf{b} = [18, 18, 18, 18] \quad (\text{E.64})$$

The **L1-norm error rate** $\| \bar{\mathbf{e}} \|_1$ is found to be,

$$\| \bar{\mathbf{e}} \|_1 = \frac{1}{4} \cdot (2 + 2 + 2 + 2) = 2. \quad (\text{E.65})$$

The **L2-norm error rate** $\| \bar{\mathbf{e}} \|_2$ is found to be,

$$\| \bar{\mathbf{e}} \|_2 = \frac{1}{4} \cdot \sqrt{(4 + 4 + 4 + 4)} = 1. \quad (\text{E.66})$$

...

E.4. Strains

The purpose of this section is to provide justification for comparison of Almansi-Euler strain measures from experiments to Green-Lagrange strain measures from simulations. Comparisons of these two measures are justified if the magnitudes are sufficiently small. For the strain magnitudes found in the current simulations, we feel justified. However, for future work wherein the skull rotation kinematics become more severe, care will need to be taken to harmonize the strain measures.

E.4.1. Seth-Hill Strain Family

Seth and Hill showed that the Green-Lagrange strain tensor \mathbf{E} and the Euler-Almansi strain tensor \mathbf{e} are special cases of the so-called Seth-Hill family of strain measures, defined as

$$\mathbf{E}^{(m)} \triangleq \begin{cases} \frac{1}{m} (\mathbf{U}^m - \mathbf{1}) & \text{for } m \neq 0, \\ \ln \mathbf{U} & \text{for } m = 0; \text{ and,} \end{cases} \quad (\text{E.67})$$

$$\mathbf{e}^{(m)} \triangleq \begin{cases} \frac{1}{m} (\mathbf{1} - \mathbf{v}^{-m}) & \text{for } m \neq 0, \\ \ln \mathbf{v} & \text{for } m = 0. \end{cases} \quad (\text{E.68})$$

The principal stretches $\lambda_\alpha = l_\alpha/L_\alpha$, $0 < \lambda_\alpha < \infty$, allow the strain measure $\mathbf{E}^{(m)}$ to be written as principal strains, as a function of principal stretch, $f(\lambda_\alpha)$,

$$\begin{aligned} \mathbf{E}^{(m)} &= \sum_{\alpha=1}^3 f(\lambda_\alpha) \mathbf{N}_\alpha \otimes \mathbf{N}_\alpha \iff \\ \mathbf{E}_{IJ}^{(m)} &= f(\lambda_1) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + f(\lambda_2) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + f(\lambda_3) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (\text{E.69})$$

where the **stretch function**

$$f(\lambda_\alpha) \triangleq \begin{cases} \frac{1}{m} (\lambda_\alpha^m - 1) & \text{for } m \neq 0, \\ \ln \lambda_\alpha & \text{for } m = 0. \end{cases} \quad (\text{E.70})$$

For integer values⁴ of $m \in [-2, 2]$, five common strain measures result, listed in Table E-2, in their three-dimensional and one-dimensional forms. Similar relationships can be constructed for the spatial tensors using

$$\mathbf{e}^{(m)} = \sum_{\alpha=1}^3 g(\lambda_\alpha) \mathbf{n}_\alpha \otimes \mathbf{n}_\alpha, \quad (\text{E.71})$$

⁴Technically, m can be any real number, not just an integer.

$$g(\lambda_\alpha) \triangleq \begin{cases} \frac{1}{m} (1 - \lambda_\alpha^{-m}) & \text{for } m \neq 0, \\ \ln \lambda_\alpha & \text{for } m = 0. \end{cases} \quad (\text{E.72})$$

Table E-2. Strains obtained from the Seth-Hill family.

m	Name	3D	1D
2	Green-Lagrange	$\mathbf{E}^{(2)} = \frac{1}{2} (\mathbf{U}^2 - \mathbf{1})$	$\mathbf{E}_{\text{LAG}} = \frac{1}{2} \left[\left(\frac{\ell}{L} \right)^2 - 1 \right]$
1	engineering (Biot, nominal)	$\mathbf{E}^{(1)} = \mathbf{U} - \mathbf{1}$	$\mathbf{E}_{\text{ENG}} = \frac{\ell - L}{L}$
0	log (Hencky, natural)	$\mathbf{E}^{(0)} = \ln \mathbf{U}$	$\mathbf{E}_{\text{LOG}} = \ln \left(\frac{\ell}{L} \right)$
-1	true	$\mathbf{E}^{(-1)} = \mathbf{1} - \mathbf{U}^{-1}$	$\mathbf{E}_{\text{TRUE}} = \frac{\ell - L}{\ell}$
-2	Euler-Almansi	$\mathbf{E}^{(-2)} = \frac{1}{2} (\mathbf{1} - \mathbf{U}^{-2})$	$\mathbf{E}_{\text{EUL}} = \frac{1}{2} \left[1 - \left(\frac{L}{\ell} \right)^2 \right]$

Figure E-9 illustrates the one-dimensional strains as a function of stretch ratio $\lambda = \ell/L$. The figure illustrates several results:

- For small stretches, $\ell \approx L$, (a) the stretch ratio is near unity, $\lambda \approx 1$, (b) the strain values are small, $f(\lambda) \approx 0$, and (c) the tangent of the strains with respect to the stretch ratio is near unity, $df/d\lambda \approx 1$.
- For elongations, $\lambda > 1$, the strain monotonically increases since $df/d\lambda > 0$ when $\lambda > 0$.
- For extreme compressions, $\lambda \approx 0$, (a) the Green-Lagrange strain goes to a value of $-\frac{1}{2}$, (b) the engineering (Biot, nominal) strain tensor goes to a value of -1 , and (c) the log, true, and Eulerian strains tend to $-\infty$.
- The engineering (Biot, nominal) strain is a linear function of stretch λ ; all other measures are nonlinear functions of stretch λ .

[Neff et al., 2016] suggested “reasonable requirements”⁵ on $f(\lambda) : \mathbb{R}^+ \mapsto \mathbb{R}$, summarized in Table E-3, wherein a “+” indicates the requirement is satisfied and a “-” indicates the requirement is not satisfied.

Table E-3. Reasonable requirements on the stretch function.

Requirement	$\mathbf{E}^{(2)}$	$\mathbf{E}^{(1)}$	$\mathbf{E}^{(0)}$	$\mathbf{E}^{(-1)}$	$\mathbf{E}^{(-2)}$
f is smooth	+	+	+	+	+
f is monotonically increasing	+	+	+	+	+
$f _{\lambda=1} = 0$	+	+	+	+	+
$f' _{\lambda=1} = 1$	+	+	+	+	+
As $\lambda \rightarrow \infty$, $f \rightarrow +\infty$	+	+	+	-	-
As $\lambda \rightarrow 0^+$, $f \rightarrow -\infty$	-	-	+	+	+
$-f(\lambda) = f\left(\frac{1}{\lambda}\right)$	-	-	+	-	-
$f(\lambda^\alpha) = \alpha f(\lambda)$ for $\alpha \in \mathbb{R}$	-	-	+	-	-

The results in Table E-3 and Figure E-9 illustrate that the log strain $\mathbf{E}^{(0)}$ retains more of the desired qualities than any other strain tensor, in the context finite compression and extension.⁶

For infinitesimal deformation, all tensors converge to the infinitesimal strain tensor $\boldsymbol{\epsilon} = \text{SYM}(\nabla \mathbf{u})$. For finite deformation, however, the Seth-Hill strain measures given by the $f(\lambda)$ function diverge quickly for both large compression and large tension. Figure E-10 illustrates the one-dimensional strains subtracted from the natural logarithmic strain, $\ln \lambda$, as a function of stretch ratio $\lambda = \ell/L$. The log strain is considered as the finite deformation baseline. The results show, for example, that in compression at $\lambda = 0.8$, the Green-Langrange strain tensor underreports the log strain by nearly 5%. Such a result

⁵See page 6.

⁶The Bažant strain, $f(\lambda) = \frac{1}{2}(\lambda - \frac{1}{\lambda})$, not considered here, also satisfies $-f(\lambda) = f\left(\frac{1}{\lambda}\right)$.

illustrates that for finite deformation, (1) strain measures are **not** interchangeable, and (2) it is incomplete to simply say “strain.” Rather, both the strain value *and* strain tensor must be specified.

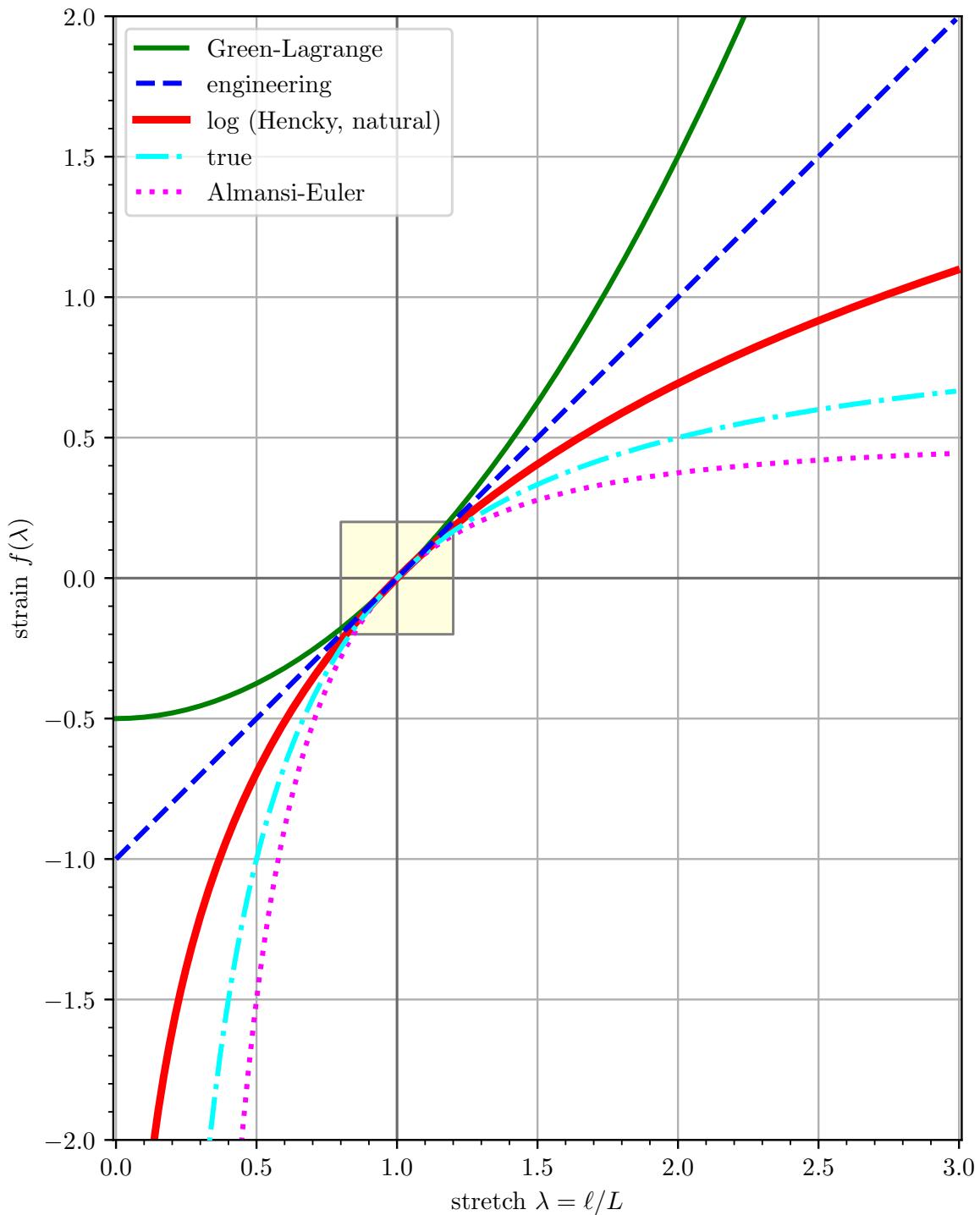


Figure E-9. One dimensional strain as a function of stretch ratio.

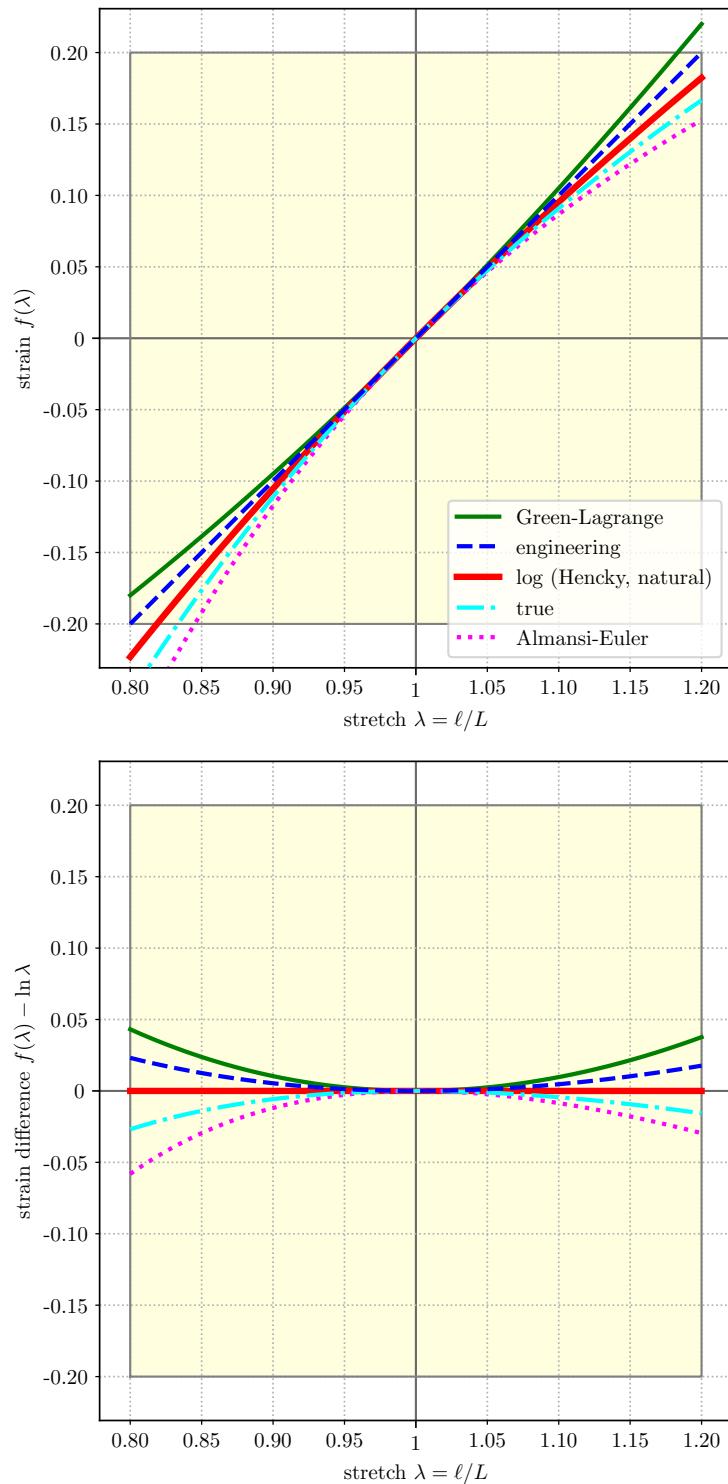


Figure E-10. One dimensional strain difference of strain function minus the natural logarithmic strain as a function of stretch ratio.

E.5. Strain Rates

Per [Pierson, 2018], Solid Mechanics (SSM) operates on the log-strain of the left stretch tensor \mathbf{v} , with solid element variable `log_strain` as,

$$\text{log_strain} = \ln \mathbf{v}. \quad (\text{E.73})$$

With $\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{v}\mathbf{R}$, $\mathbf{D} = \mathbf{R}^T \mathbf{d} \mathbf{R}$, and $\mathbf{d} = \text{SYM}(\nabla \mathbf{v})$, additional kinematic variables available from SSM are, with

$$\text{unrotated_log_strain} = \ln \mathbf{U}, \quad (\text{E.74})$$

$$\text{green_lagrange_strain} = \mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{1}), \text{ and} \quad (\text{E.75})$$

$$\text{unrotated_rate_of_deformation} = \mathbf{D}. \quad (\text{E.76})$$

SSM works in the unrotated current configuration because of its roots with finite deformation hypoelastic rate constitutive equations.⁷

In contrast, our collaborators' experiments utilize the reference configuration, since it is the easiest configuration for experimental measurements. Therefore, this section contains details of properly mapping back kinematic variables of interest from the current configuration to the reference configuration.

In this section, we describe the methodology of calculating Green-Lagrange strain rates, and invariants thereof.⁸

E.5.1. Green-Lagrange Strain Rate

With Rate of Deformation Tensor

The time derivative of the Green-Lagrange strain tensor, $\dot{\mathbf{E}}$, is the pull back of the rate of deformation tensor, \mathbf{d} ,

$$\dot{\mathbf{E}} = \mathbf{F}^T \mathbf{d} \mathbf{F} \iff \dot{\mathbf{E}}_{IJ} = \mathbf{F}_{Ji} \mathbf{d}_{ij} \mathbf{F}_{jJ}, \quad (\text{E.77})$$

⁷See, for example [Simo and Hughes, 2006], in particular Section 7.3, “Ad Hoc Extensions of Phenomenological Plasticity Based on Hypoelastic Relationships,” pages 269–275.

⁸Reference: ../method/edotvm.tex

with $\mathbf{d} = \mathbf{d}^T$, $\dot{\mathbf{E}} = \dot{\mathbf{E}}^T$, and $\mathbf{F} \neq \mathbf{F}^T$. The components of $\dot{\mathbf{E}}$ follow:

$$\begin{aligned}\dot{E}_{11} &= F_{11} (d_{11} F_{11} + d_{12} F_{21} + d_{13} F_{31}) + \\ &F_{21} (d_{12} F_{11} + d_{22} F_{21} + d_{23} F_{31}) + \\ &F_{31} (d_{13} F_{11} + d_{23} F_{21} + d_{33} F_{31})\end{aligned}\quad (\text{E.78})$$

$$\begin{aligned}\dot{E}_{12} &= F_{11} (d_{11} F_{12} + d_{12} F_{22} + d_{13} F_{32}) + \\ &F_{21} (d_{12} F_{12} + d_{22} F_{22} + d_{23} F_{32}) + \\ &F_{31} (d_{13} F_{12} + d_{23} F_{22} + d_{33} F_{32})\end{aligned}\quad (\text{E.79})$$

$$\begin{aligned}\dot{E}_{13} &= F_{11} (d_{11} F_{13} + d_{12} F_{23} + d_{13} F_{33}) + \\ &F_{21} (d_{12} F_{13} + d_{22} F_{23} + d_{23} F_{33}) + \\ &F_{31} (d_{13} F_{13} + d_{23} F_{23} + d_{33} F_{33})\end{aligned}\quad (\text{E.80})$$

$$\begin{aligned}\dot{E}_{22} &= F_{12} (d_{11} F_{12} + d_{12} F_{22} + d_{13} F_{32}) + \\ &F_{22} (d_{12} F_{12} + d_{22} F_{22} + d_{23} F_{32}) + \\ &F_{32} (d_{13} F_{12} + d_{23} F_{22} + d_{33} F_{32})\end{aligned}\quad (\text{E.81})$$

$$\begin{aligned}\dot{E}_{23} &= F_{12} (d_{11} F_{13} + d_{12} F_{23} + d_{13} F_{33}) + \\ &F_{22} (d_{12} F_{13} + d_{22} F_{23} + d_{23} F_{33}) + \\ &F_{32} (d_{13} F_{13} + d_{23} F_{23} + d_{33} F_{33})\end{aligned}\quad (\text{E.82})$$

$$\begin{aligned}\dot{E}_{33} &= F_{13} (d_{11} F_{13} + d_{12} F_{23} + d_{13} F_{33}) + \\ &F_{23} (d_{12} F_{13} + d_{22} F_{23} + d_{23} F_{33}) + \\ &F_{33} (d_{13} F_{13} + d_{23} F_{23} + d_{33} F_{33})\end{aligned}\quad (\text{E.83})$$

With Unrotated Rate of Deformation Tensor

SSM reports \mathbf{D} , not \mathbf{d} . So, the rotations (embedded in the deformation gradient) should not be used for the pull back operation. Instead, just the stretches should be used. The time derivative of the Green-Lagrange strain tensor, $\dot{\mathbf{E}}$, is

$$\dot{\mathbf{E}} = \mathbf{U} \mathbf{D} \mathbf{U} \iff \dot{E}_{IJ} = U_{IK} D_{KL} U_{LJ}, \quad (\text{E.84})$$

with $\mathbf{D} = \mathbf{d}^T$, $\dot{\mathbf{E}} = \dot{\mathbf{E}}^T$, and $\mathbf{U} = \mathbf{U}^T$. The components of $\dot{\mathbf{E}}$ follow:

$$\begin{aligned}\dot{E}_{11} &= U_{11} (D_{11} U_{11} + D_{12} U_{12} + D_{13} U_{13}) + \\ &U_{12} (D_{12} U_{11} + D_{22} U_{12} + D_{23} U_{13}) + \\ &U_{13} (D_{13} U_{11} + D_{23} U_{12} + D_{33} U_{13})\end{aligned}\quad (\text{E.85})$$

$$\begin{aligned}\dot{E}_{12} &= U_{11} (D_{11} U_{12} + D_{12} U_{22} + D_{13} U_{23}) + \\ &U_{12} (D_{12} U_{12} + D_{22} U_{22} + D_{23} U_{23}) + \\ &U_{13} (D_{13} U_{12} + D_{23} U_{22} + D_{33} U_{23})\end{aligned}\quad (\text{E.86})$$

$$\begin{aligned}\dot{E}_{13} &= U_{11} (D_{11} U_{13} + D_{12} U_{23} + D_{13} U_{33}) + \\ &U_{12} (D_{12} U_{13} + D_{22} U_{23} + D_{23} U_{33}) + \\ &U_{13} (D_{13} U_{13} + D_{23} U_{23} + D_{33} U_{33})\end{aligned}\quad (\text{E.87})$$

$$\begin{aligned}\dot{E}_{22} &= U_{12} (D_{11} U_{12} + D_{12} U_{22} + D_{13} U_{23}) + \\ &U_{22} (D_{12} U_{12} + D_{22} U_{22} + D_{23} U_{23}) + \\ &U_{23} (D_{13} U_{12} + D_{23} U_{22} + D_{33} U_{23})\end{aligned}\quad (\text{E.88})$$

$$\begin{aligned}\dot{E}_{23} &= U_{12} (D_{11} U_{13} + D_{12} U_{23} + D_{13} U_{33}) + \\ &U_{22} (D_{12} U_{13} + D_{22} U_{23} + D_{23} U_{33}) + \\ &U_{23} (D_{13} U_{13} + D_{23} U_{23} + D_{33} U_{33})\end{aligned}\quad (\text{E.89})$$

$$\begin{aligned}\dot{E}_{33} &= U_{13} (D_{11} U_{13} + D_{12} U_{23} + D_{13} U_{33}) + \\ &U_{23} (D_{12} U_{13} + D_{22} U_{23} + D_{23} U_{33}) + \\ &U_{33} (D_{13} U_{13} + D_{23} U_{23} + D_{33} U_{33})\end{aligned}\quad (\text{E.90})$$

E.5.2. The Mises Invariant

The J_2 invariant of $\dot{\mathbf{E}}$ is

$$J_2(\dot{\mathbf{E}}) = \frac{1}{2} \text{dev}(\dot{\mathbf{E}}) : \text{dev}(\dot{\mathbf{E}}) \quad (\text{E.91})$$

$$= \frac{(\dot{E}_{11} - \dot{E}_{22})^2 + (\dot{E}_{22} - \dot{E}_{33})^2 + (\dot{E}_{33} - \dot{E}_{11})^2}{6} + \dot{E}_{12}^2 + \dot{E}_{23}^2 + \dot{E}_{31}^2. \quad (\text{E.92})$$

Then, the Mises Green-Lagrange strain rate is

$$\dot{\mathbf{E}}_{\text{VM}} = \sqrt{\frac{2}{3} \text{dev}(\dot{\mathbf{E}}) : \text{dev}(\dot{\mathbf{E}})} = \sqrt{\frac{4}{3} J_2(\dot{\mathbf{E}})}. \quad (\text{E.93})$$

Similarly, the **Mises Green-Lagrange strain** is

$$\mathbf{E}_{\text{VM}} = \sqrt{\frac{2}{3} \text{dev}(\mathbf{E}) : \text{dev}(\mathbf{E})} = \sqrt{\frac{4}{3} J_2(\mathbf{E})}. \quad (\text{E.94})$$

E.5.3. Implementation

We include a few implementation notes here mainly as a future memory aid to the authors rather than an explanation for readers.

1. Implement $\dot{\mathbf{E}}_{IJ}$ components as SSM function
`strain_rate_from_unrotated_rate_of_deformation.i`,
checked into `casco_sim/include/`, similar to how `C2B2/sim/include/` is implemented.
2. Implement J_2 invariant as SSM function
`j2_invariant.i`,
checked into `casco_sim/include/`.
3. Update `casco_sim/shear_simple/shear_simple.i` to include the $\dot{\mathbf{E}}_{IJ}$ and J_2 SSM user variables. Test implementation results against closed form solution (see Chad's simple shear notes from `main.pdf` and supporting *Mathematica* notebook files, `casco_doc/writing/report/Green_Lagrange_strain_rate.nb`.)

E.6. Toward Eigenvalues

SSM has eigenvalues of the log strain tensor, $\ln \mathbf{v}$. To calculate the eigenvalues of the Green-Lagrange strain tensor \mathbf{E} , without resorting to eigenvalue calculations, consider the following: obtain the eigenvalues of directly from the stretches. To do this, begin with the spectral representation of the deformation gradient \mathbf{F} , the right stretch \mathbf{U} , the rotation \mathbf{R} , the left stretch \mathbf{v} , the right Cauchy-Green strain tensor \mathbf{C} , and the left Cauchy-Green strain tensor \mathbf{b} , respectively,

$$\mathbf{F} = \sum_{\alpha=1}^3 \lambda_{\alpha} \mathbf{n}_{\alpha} \otimes \mathbf{N}_{\alpha} \quad \mathbf{U} = \sum_{\alpha=1}^3 \lambda_{\alpha} \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha} \quad (\text{E.95})$$

$$\mathbf{R} = \sum_{\alpha=1}^3 \mathbf{n}_{\alpha} \otimes \mathbf{N}_{\alpha} \quad \mathbf{v} = \sum_{\alpha=1}^3 \lambda_{\alpha} \mathbf{n}_{\alpha} \otimes \mathbf{n}_{\alpha} \quad (\text{E.96})$$

$$\mathbf{C} = \sum_{\alpha=1}^3 \lambda_{\alpha}^2 \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha} \quad \mathbf{b} = \sum_{\alpha=1}^3 \lambda_{\alpha}^2 \mathbf{n}_{\alpha} \otimes \mathbf{n}_{\alpha} \quad (\text{E.97})$$

The Green-Lagrange stain tensor \mathbf{E} and the Euler-Almansi strain tensor \mathbf{e} , in principal stretches and stretch directions, are

$$\mathbf{E} = \sum_{\alpha=1}^3 \frac{1}{2} (\lambda_{\alpha}^2 - 1) \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha}, \quad \mathbf{e} = \sum_{\alpha=1}^3 \frac{1}{2} (1 - \lambda_{\alpha}^{-2}) \mathbf{n}_{\alpha} \otimes \mathbf{n}_{\alpha}. \quad (\text{E.98})$$

The generalization of the Seth-Hill material stain tensor $\mathbf{E}^{(m)}$ and spatial strain tensor $\mathbf{e}^{(m)}$, in principal stretches and stretch directions, are

$$\mathbf{E}^{(m)} = \sum_{\alpha=1}^3 \frac{1}{m} (\lambda_{\alpha}^m - 1) \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha}, \quad \mathbf{e}^{(m)} = \sum_{\alpha=1}^3 \frac{1}{m} (1 - \lambda_{\alpha}^{-m}) \mathbf{n}_{\alpha} \otimes \mathbf{n}_{\alpha}, \quad (\text{E.99})$$

and the relationship between the two strain tensors is given through a rotation \mathbf{R} transformation,

$$\mathbf{e}^{(-m)} = \mathbf{R} \mathbf{E}^{(m)} \mathbf{R}^T, \quad (\text{E.100})$$

obtained through use of (E.99) with $\mathbf{n}_{\alpha} = \mathbf{R} \mathbf{N}_{\alpha}$. In the case when $m \rightarrow \infty$, the material and spatial **logarithmic strain tensors**, also known as the **Hencky** material and spatial strain tensors, \mathbf{H} and \mathbf{h} , are obtained, *e.g.*, [Xiao et al., 1997]⁹, as,

$$\mathbf{H} = \ln \mathbf{U} = \mathbf{E}^{(0)} = \sum_{\alpha=1}^3 \ln \lambda_{\alpha} \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha}, \text{ and} \quad (\text{E.101})$$

$$\mathbf{h} = \ln \mathbf{v} = \mathbf{e}^{(0)} = \sum_{\alpha=1}^3 \ln \lambda_{\alpha} \mathbf{n}_{\alpha} \otimes \mathbf{n}_{\alpha}. \quad (\text{E.102})$$

E.6.1. Maximum Principal Strain using \mathbf{E}

Since

$$\mathbf{E} = \frac{1}{2} (\mathbf{C} - \mathbf{1}) \text{ and } \mathbf{C} = \sum_{\alpha=1}^3 \lambda_{\alpha}^2 \mathbf{N}_{\alpha} \otimes \mathbf{N}_{\alpha}, \quad (\text{E.103})$$

then

$$\text{eig}_{\alpha}(\mathbf{E}) = \frac{1}{2} [\lambda_{\alpha}^2 - 1] \quad (\text{E.104})$$

$$= \frac{1}{2} [\{\exp(\ln \lambda_{\alpha})\}^2 - 1]. \quad (\text{E.105})$$

E.6.2. Implementation

The SSM implementations, checked into the `casco_sim/include/` folder as files,

- `e_eigenvalues-variables.i`

⁹See page 54, Eq. (2.2).

- `e_eigenvalues-functions.i`

are as follows:

$$\begin{aligned} \text{min_principal_green_lagrange_strain} \\ = 0.5 * (\text{pow}(\text{exp}(\text{min_principal_log_strain}), 2) - 1.0) \end{aligned} \quad (\text{E.106})$$

$$\begin{aligned} \text{intermediate_principal_green_lagrange_strain} \\ = 0.5 * (\text{pow}(\text{exp}(\text{intermediate_principal_log_strain}), 2) - 1.0) \end{aligned} \quad (\text{E.107})$$

$$\begin{aligned} \text{max_principal_green_lagrange_strain} \\ = 0.5 * (\text{pow}(\text{exp}(\text{max_principal_log_strain}), 2) - 1.0) \end{aligned} \quad (\text{E.108})$$

APPENDIX F. Code

The code details pertain to input files for SSM, as well as Python implementations for post-processing [Hovey, 2020]. These details are memorialized in turn below.

F.1. SSM Input Data

F.1.1. Green Lagrange Strain Rate and Mises Invariant

e_dot_vm-functions.i

```
1 # include this towards the top after begin sierra
2
3 begin function e_dot_1_1_function
4 type is analytic
5 expression variable: u = element_sym_tensor right_stretch
6 expression variable: d = element_tensor rate_of_deformation
7 evaluate expression = "(d_xx*u_xx*u_xx) + (2*d_xy*u_xx*u_xy) +          \#
8                      (d_yy*u_xy*u_xy) + (2*d_xz*u_xx*u_zx) +          \#
9                      (2*d_yz*u_xy*u_yz) + (d_zz*u_zx*u_zx)"
```

```
10 end function e_dot_1_1_function
11
12 begin function e_dot_2_2_function
13 type is analytic
14 expression variable: u = element_sym_tensor right_stretch
15 expression variable: d = element_tensor rate_of_deformation
16 evaluate expression = "(d_xx*u_xy*u_xy) + (2*d_xy*u_xy*u_yy) +          \#
17                      (d_yy*u_yy*u_xy) + (2*d_xz*u_xy*u_yz) +          \#
18                      (2*d_yz*u_yy*u_yz) + (d_zz*u_yz*u_yz)"
```

```
19 end function e_dot_2_2_function
20
21 begin function e_dot_3_3_function
22 type is analytic
23 expression variable: u = element_sym_tensor right_stretch
24 expression variable: d = element_tensor rate_of_deformation
25 evaluate expression = "(d_xx*u_zx*u_zx) + (2*d_xy*u_zx*u_yz) +          \#
26                      (d_yy*u_yz*u_yz) + (2*d_xz*u_zx*u_zz) +          \#
27                      (2*d_yz*u_yz*u_zz) + (d_zz*u_zz*u_zz)"
```

```
28 end function e_dot_3_3_function
29
30 begin function e_dot_1_2_function
31 type is analytic
32 expression variable: u = element_sym_tensor right_stretch
```

```

33 expression variable: d = element_tensor rate_of_deformation
34 evaluate expression = "(d_xy*u_xy*u_xy) + (d_xx*u_xx*u_xy) +      \#
35             (d_xy*u_yy*u_xx) + (d_xz*u_yz*u_xx) +      \#
36             (d_yy*u_yy*u_xy) + (d_yz*u_yz*u_xy) +      \#
37             (d_xz*u_xy*u_zx) + (d_yz*u_yy*u_zx) +      \#
38             (d_zz*u_yz*u_zx)"                                \#
39 end function e_dot_1_2_function
40
41 begin function e_dot_1_3_function
42 type is analytic
43 expression variable: u = element_sym_tensor right_stretch
44 expression variable: d = element_tensor rate_of_deformation
45 evaluate expression = "(d_xz*u_zx*u_zx) + (d_xx*u_zx*u_xx) +      \#
46             (d_xy*u_yz*u_xx) + (d_xz*u_zz*u_xx) +      \#
47             (d_xy*u_xy*u_zx) + (d_yy*u_yz*u_xy) +      \#
48             (d_yz*u_zz*u_xy) + (d_yz*u_yz*u_zx) +      \#
49             (d_zz*u_zz*u_zx)"                                \#
50 end function e_dot_1_3_function
51
52 begin function e_dot_2_3_function
53 type is analytic
54 expression variable: u = element_sym_tensor right_stretch
55 expression variable: d = element_tensor rate_of_deformation
56 evaluate expression = "(d_yz*u_yz*u_yz) + (d_xx*u_zx*u_xy) +      \#
57             (d_xy*u_yz*u_xy) + (d_xz*u_zz*u_xy) +      \#
58             (d_xy*u_zx*u_yy) + (d_yy*u_yz*u_yy) +      \#
59             (d_yz*u_zz*u_yy) + (d_xz*u_zx*u_yz) +      \#
60             (d_zz*u_zz*u_yz)"                                \#
61 end function e_dot_2_3_function
62
63 begin function j_2_e_dot_function
64 type is analytic
65 expression variable: e_dot_1_1 = element e_dot_1_1
66 expression variable: e_dot_2_2 = element e_dot_2_2
67 expression variable: e_dot_3_3 = element e_dot_3_3
68 expression variable: e_dot_1_2 = element e_dot_1_2
69 expression variable: e_dot_1_3 = element e_dot_1_3
70 expression variable: e_dot_2_3 = element e_dot_2_3
71 evaluate expression = "((pow((e_dot_1_1 - e_dot_2_2),2) +      \#
72             pow((e_dot_2_2 - e_dot_3_3),2) +      \#
73             pow((e_dot_3_3 - e_dot_1_1),2))/6) +      \#
74             pow((e_dot_1_2),2) +      \#
75             pow((e_dot_2_3),2) +      \#
76             pow((e_dot_1_3),2)"                                \#
77 end function j_2_e_dot_function
78
79 begin function e_dot_vm_function
80 type is analytic
81 expression variable: j_2_e_dot = element j_2_e_dot
82 evaluate expression = "sqrt((4/3)*j_2_e_dot)"
83 end function e_dot_vm_function

```

e_dot_vm-variables.i

```
1 begin user variable e_dot_1_1
2 initial value = 0.0
3 type = element real length = 1
4 end
5
6 begin user variable e_dot_2_2
7 initial value = 0.0
8 type = element real length = 1
9 end
10
11 begin user variable e_dot_3_3
12 initial value = 0.0
13 type = element real length = 1
14 end
15
16 begin user variable e_dot_1_2
17 initial value = 0.0
18 type = element real length = 1
19 end
20
21 begin user variable e_dot_1_3
22 initial value = 0.0
23 type = element real length = 1
24 end
25
26 begin user variable e_dot_2_3
27 initial value = 0.0
28 type = element real length = 1
29 end
30
31 begin user variable j_2_e_dot
32 initial value = 0.0
33 type = element real length = 1
34 end
35
36 begin user variable e_dot_vm
37 initial value = 0.0
38 type = element real length = 1
39 end
40
41 begin user output
42 compute element e_dot_1_1 as function e_dot_1_1_function
43 compute element e_dot_2_2 as function e_dot_2_2_function
44 compute element e_dot_3_3 as function e_dot_3_3_function
45 compute element e_dot_1_2 as function e_dot_1_2_function
46 compute element e_dot_1_3 as function e_dot_1_3_function
47 compute element e_dot_2_3 as function e_dot_2_3_function
48 compute element j_2_e_dot as function j_2_e_dot_function
49 compute element e_dot_vm as function e_dot_vm_function
50 compute at every step
```

```
51 end user output
```

F.1.2. Green Lagrange Strain Eigenvalues

e_eigenvalues-functions.i

```
1 #include this towards the top after begin sierra
2
3 begin function min_principal_green_lagrange_strain_function
4 type is analytic
5 expression variable: min_eig = element_real min_principal_log_strain
6 evaluate expression = "0.5*(pow(exp(min_eig),2) - 1.0)"
7 end function min_principal_green_lagrange_strain_function
8
9 begin function intermediate_principal_green_lagrange_strain_function
10 type is analytic
11 expression variable: intermediate_eig = element_real intermediate_principal_log_strain
12 evaluate expression = "0.5*(pow(exp(intermediate_eig),2) - 1.0)"
13 end function intermediate_principal_green_lagrange_strain_function
14
15 begin function max_principal_green_lagrange_strain_function
16 type is analytic
17 expression variable: max_eig = element_real max_principal_log_strain
18 evaluate expression = "0.5*(pow(exp(max_eig),2) - 1.0)"
19 end function max_principal_green_lagrange_strain_function
```

e_eigenvalues-variables.i

```
1 begin user variable min_principal_green_lagrange_strain
2 initial value = 0.0
3 type = element real length = 1
4 end
5
6 begin user variable intermediate_principal_green_lagrange_strain
7 initial value = 0.0
8 type = element real length = 1
9 end
10
11 begin user variable max_principal_green_lagrange_strain
12 initial value = 0.0
13 type = element real length = 1
14 end
15
16 begin user output
17 compute element min_principal_green_lagrange_strain \#
18     as function min_principal_green_lagrange_strain_function
19 compute element intermediate_principal_green_lagrange_strain \#
```

```

20      as function intermediate_principal_green_lagrange_strain_function
21  compute element max_principal_green_lagrange_strain \#
22      as function max_principal_green_lagrange_strain_function
23  compute at every step
24 end user output

```

F.1.3. Green Lagrange Mises Invariant

e_vm-functions.i

```

1  #include this towards the top after begin sierra
2
3 begin function j_2_e_function
4 type is analytic
5 expression variable: e = element_sym_tensor green_lagrange_strain
6 evaluate expression = "((pow((e_xx - e_yy),2) + \#
7           pow((e_yy - e_zz),2) + \#
8           pow((e_zz - e_xx),2))/6) + \#
9           pow((e_xy),2) + \#
10          pow((e_yz),2) + \#
11          pow((e_zx),2)" :
12 end function j_2_e_function
13
14 begin function e_vm_function
15 type is analytic
16 expression variable: j_2_e = element j_2_e
17 evaluate expression = "sqrt((4/3)*j_2_e)"
18 end function e_vm_function

```

e_vm-variables.i

```

1 begin user variable j_2_e
2 initial value = 0.0
3 type = element real length = 1
4 end
5
6 begin user variable e_vm
7 initial value = 0.0
8 type = element real length = 1
9 end
10
11 begin user output
12 compute element j_2_e as function j_2_e_function
13 compute element e_vm as function e_vm_function
14 compute at every step
15 end user output

```

F.1.4. First Piola-Kirchhoff Stress

pk1-functions.i

```
1 # include this towards the top after begin sierra
2
3 begin function pk1_1_1_function
4 type is analytic
5 expression variable: sigma = element_sym_tensor cauchy_stress
6 expression variable: f = element_tensor deformation_gradient
7 evaluate expression = "((-f_yz*f_zy) + (f_yy*f_zz))*sigma_xx"
8 end function pk1_1_1_function
9
10 begin function pk1_1_2_function
11 type is analytic
12 expression variable: sigma = element_sym_tensor cauchy_stress
13 expression variable: f = element_tensor deformation_gradient
14 evaluate expression = "((f_yz*f_zx) - (f_yx*f_zz))*sigma_xy"
15 end function pk1_1_2_function
16
17 begin function pk1_1_3_function
18 type is analytic
19 expression variable: sigma = element_sym_tensor cauchy_stress
20 expression variable: f = element_tensor deformation_gradient
21 evaluate expression = "((-f_yy*f_zx) + (f_yx*f_zy))*sigma_zx"
22 end function pk1_1_3_function
23
24 begin function pk1_2_1_function
25 type is analytic
26 expression variable: sigma = element_sym_tensor cauchy_stress
27 expression variable: f = element_tensor deformation_gradient
28 evaluate expression = "((f_xz*f_zy) - (f_xy*f_zz))*sigma_xy"
29 end function pk1_2_1_function
30
31 begin function pk1_2_2_function
32 type is analytic
33 expression variable: sigma = element_sym_tensor cauchy_stress
34 expression variable: f = element_tensor deformation_gradient
35 evaluate expression = "((-f_xz*f_zx) + (f_xx*f_zz))*sigma_yy"
36 end function pk1_2_2_function
37
38 begin function pk1_2_3_function
39 type is analytic
40 expression variable: sigma = element_sym_tensor cauchy_stress
41 expression variable: f = element_tensor deformation_gradient
42 evaluate expression = "((f_xy*f_zx) - (f_xx*f_zy))*sigma_yz"
43 end function pk1_2_3_function
44
45 begin function pk1_3_1_function
46 type is analytic
47 expression variable: sigma = element_sym_tensor cauchy_stress
48 expression variable: f = element_tensor deformation_gradient
```

```

49 evaluate expression = "((-f_xz*f_yy) + (f_xy*f_yz))*sigma_zx"
50 end function pk1_3_1_function
51
52 begin function pk1_3_2_function
53 type is analytic
54 expression variable: sigma = element_sym_tensor cauchy_stress
55 expression variable: f = element_tensor deformation_gradient
56 evaluate expression = "((f_xz*f_yx) - (f_xx*f_yz))*sigma_yz"
57 end function pk1_3_2_function
58
59 begin function pk1_3_3_function
60 type is analytic
61 expression variable: sigma = element_sym_tensor cauchy_stress
62 expression variable: f = element_tensor deformation_gradient
63 evaluate expression = "((-f_xy*f_yx) + (f_xx*f_yy))*sigma_zz"
64 end function pk1_3_3_function

```

pk1-variables.i

```

1 begin user variable pk1_1_1
2 initial value = 0.0
3 type = element real length = 1
4 end
5
6 begin user variable pk1_1_2
7 initial value = 0.0
8 type = element real length = 1
9 end
10
11 begin user variable pk1_1_3
12 initial value = 0.0
13 type = element real length = 1
14 end
15
16 begin user variable pk1_2_1
17 initial value = 0.0
18 type = element real length = 1
19 end
20
21 begin user variable pk1_2_2
22 initial value = 0.0
23 type = element real length = 1
24 end
25
26 begin user variable pk1_2_3
27 initial value = 0.0
28 type = element real length = 1
29 end
30
31 begin user variable pk1_3_1
32 initial value = 0.0

```

```

33 type = element real length = 1
34 end
35
36 begin user variable pk1_3_2
37 initial value = 0.0
38 type = element real length = 1
39 end
40
41 begin user variable pk1_3_3
42 initial value = 0.0
43 type = element real length = 1
44 end
45
46
47 begin user output
48 compute element pk1_1_1 as function pk1_1_1_function
49 compute element pk1_1_2 as function pk1_1_2_function
50 compute element pk1_1_3 as function pk1_1_3_function
51 compute element pk1_2_1 as function pk1_2_1_function
52 compute element pk1_2_2 as function pk1_2_2_function
53 compute element pk1_2_3 as function pk1_2_3_function
54 compute element pk1_3_1 as function pk1_3_1_function
55 compute element pk1_3_2 as function pk1_3_2_function
56 compute element pk1_3_3 as function pk1_3_3_function
57 compute at every step
58 end user output

```

F.1.5. Translational Case (Bob-063f)

```

1 begin sierra simulation_name
2
3 # -----
4 # units: grams, centimeters, seconds
5 # -----
6
7 # -----
8 # include files
9 # -----
10
11 {include("../include/e_vm-functions.i")}
12 {include("../include/e_dot_vm-functions.i")}
13
14 # -----
15 # direction vectors
16 # -----
17
18 define point origin with coordinates 0.0 0.0 0.0
19
20 define direction x_positive with vector 1.0 0.0 0.0
21 define direction y_positive with vector 0.0 1.0 0.0
22 define direction z_positive with vector 0.0 0.0 1.0

```

```

23
24     define axis x_axis with point origin direction x_positive
25     define axis y_axis with point origin direction y_positive
26     define axis z_axis with point origin direction z_positive
27
28     # -----
29     # user functions
30     # -----
31
32     begin function gravity_accel
33         type is constant
34         begin values
35             1.0
36         end values
37     end function gravity_accel
38
39     begin function crush_soft
40         type is piecewise linear
41         abscissa = compaction
42         ordinate = stress
43         x scale  = 1.0
44         x offset = 0.0
45         y scale  = 1.0
46         y offset = 0.0
47         begin values
48             0.0977 10.05e5    # (s, dyne/cm2)
49             0.8      13.41e5 # (s, dyne/cm2)
50         end values
51     end function crush_soft
52
53     begin function crush_hard
54         type is piecewise linear
55         abscissa = compaction
56         ordinate = stress
57         x scale  = 1.0
58         x offset = 0.0
59         y scale  = 1.0
60         y offset = 0.0
61         begin values
62             0.0 17.4293e5    # (s, dyne/cm2)
63             1.0 32.4293e5 # (s, dyne/cm2)
64         end values
65     end function crush_hard
66
67     begin function constant_velocity
68         type is piecewise linear
69         abscissa = time_value
70         ordinate = velocity_value
71         x scale  = 1.0
72         x offset = 0.0
73         y scale  = 1.0
74         y offset = 0.0
75         begin values
76             0.0 20.0 # (s, cm/s)

```

```

77      1.0 20.0 # (s, cm/s)
78  end values
79 end function constant_velocity
80
81 begin function load_unload
82   type = piecewise linear
83   abscissa = time_value
84   ordinate = displacement_value
85   x scale = 1.0
86   x offset = 0.0
87   y scale = 1.0
88   y offset = 0.0
89   begin values
90     0.0 0.0
91     1.0 -1.0
92     2.0 0.0
93   end values
94 end function load_unload
95
96 begin function ramp
97   type = piecewise linear
98   abscissa = time_value
99   ordinate = displacement_value
100  x scale = 1.0
101  x offset = 0.0
102  y scale = 1.0
103  y offset = 0.0
104  begin values
105    0.0 0.0
106    1.0 -1.0 # negative to create compression
107  end values
108 end function ramp
109
110 # -----
111 # materials: listed in alphabetical order
112 # -----
113
114 # -----
115 begin material al6061t6
116 # -----
117
118 # asm.matweb.com, ma6061t6
119
120 density = 2.77 # g/cc = 0.098 lb/in^3
121
122 begin parameters for model elastic_plastic
123
124   youngs modulus = 68.9e10 # dyne/cm^2 = 10e6 psi = 68.9 GPa
125   poissons ratio = 0.33 # cm/cm = unitless
126   yield stress = 276.0e7 # dyne/cm^2 = 40e3 psi = 276 MPa
127   hardening modulus = 68.9e7 # dyne/cm^2 = 10e3 psi = 68 MPa
128   beta = 1.0 # unitless
129
130 end parameters for model elastic_plastic

```

```

131
132 end material al6061t6
133
134 # -----
135 begin material bone
136 # -----
137
138 # reference: Morse 2014, C. Franck Lacrosse paper
139 # bone: linear elastic, E = 6.5 GPa, nu = 0.45; rho = ?
140 # materials properties QC 2019-06-25 CBH with Materials for Casco document
141
142 density = 1.210 # g/cc
143
144 begin parameters for model elastic
145   youngs modulus = 8.0e10 # dyne/cm^2 = 8.0 GPa
146   poissons ratio = 0.22 # cm/cm = unitless
147 end parameters for model elastic
148
149 begin parameters for model elastic_plastic
150   youngs modulus = 8.0e10 # dyne/cm^2 = 8.0 GPa
151   poissons ratio = 0.22 # cm/cm = unitless
152   yield stress = 6.40e8 # dyne/cm^2 = 64 MPa
153   beta = 1.0 # 1.0 is isotropic hardening, 0.0 is kinematic
154   hardening modulus = 0.0 # dyne/cm^2 (perfect plasticity)
155 end parameters for model elastic_plastic
156
157 begin parameters for model johnson_cook
158   youngs modulus = 8.0e10 # dyne/cm^2 = 8.0 GPa
159   poissons ratio = 0.22 # cm/cm = unitless
160   yield stress = 9.5e8 # dyne/cm^2 = 95 MPa
161   hardening constant = 0.0 # unitless
162   hardening exponent = 0.0 # unitless
163   rate constant = 0.0 # unitless
164   reference rate = 0.0 # unitless
165   edot_ref = 1.0 # unitless
166   d1 = 0.008 # unitless
167   d2 = 0.0 # unitless
168   d3 = 0.0 # unitless
169   d4 = 0.0 # unitless
170   d5 = 0.0 # unitless
171 end parameters for model johnson_cook
172
173 end material bone
174
175 # -----
176 begin material csf # cerebral spinal fluid
177 # -----
178
179 # materials properties QC 2019-06-25 CBH with Materials for Casco document
180
181 density = 1.004 # g/cc
182
183 begin parameters for model elastic
184   # 2019-07-16 starting with Bob-035, specific (K, G) because less

```

```

185      # sensitive than specifying nu ~ = 0.499
186      #
187      # youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
188      # poissons ratio = 0.4987252 # cm/cm = unitless
189      #
190      shear modulus = 5.000e3 # dyne/cm^2 = 500 Pa, matches Zhang 2001
191      bulk modulus = 2.371e10 # dyne/cm^2 (match K for gray and white matter)
192  end parameters for model elastic
193
194  end material csf
195
196  # -----
197  begin material disc # see /cielo/sims/b1341_10.0 for rho, cs, possion, yield
198  # -----
199
200      # materials properties QC 2019-06-25 CBH with Materials for Casco document
201
202      density = 1.000 # g/cc
203
204      begin parameters for model elastic
205          youngs modulus = 4.27e8 # dyne/cm^2 = 42.7 MPa
206                      # Yang 2016, Fig 2, 30-yo at 20%/s strain rate
207          poissons ratio = 0.40 # cm/cm = unitless
208  end parameters for model elastic
209
210  end material disc
211
212  # -----
213  begin material graymatter
214  # -----
215
216      # reference: Morse 2014, C. Franck Lacrosse paper
217      # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3
218      # materials properties QC 2019-06-25 CBH with Materials for Casco document
219      # for placeholder elastic and Swanson viscoelastic
220
221      density = 1.040 # g/cc
222
223      # placeholder prior to Swanson
224      # begin parameters for model elastic
225      #   youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
226      #   poissons ratio = 0.45 # cm/cm = unitless
227      # end parameters for model elastic
228
229      begin parameters for model viscoelastic_swanson
230          bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
231      #   shear modulus = 3.400e5 # dyne/cm^2
232          A1 = 3.400e5 # dyne/cm^2
233          P1 = 0
234          B1 = 0
235          Q1 = 0
236          C1 = 0
237          R1 = 0
238          cut off strain = 0.05

```

```

239 prony shear infinity = 1.8824e-1 # Pa/Pa
240 prony shear 1 = 8.1176e-1 # Pa/Pa
241 prony shear 2 = 1.0e-4
242 prony shear 3 = 1.0e-4
243 prony shear 4 = 1.0e-4
244 prony shear 5 = 1.0e-4
245 prony shear 6 = 1.0e-4
246 prony shear 7 = 1.0e-4
247 prony shear 8 = 1.0e-4
248 prony shear 9 = 1.0e-4
249 prony shear 10 = 1.0e-4
250 # shear relax time 1 = 2.50e-2 # seconds, formerly 1.4286e-3 s
251 shear relax time 1 = 1.4286e-3 # seconds, investigate quick relaxation
252 shear relax time 2 = 100
253 shear relax time 3 = 150
254 shear relax time 4 = 200
255 shear relax time 5 = 250
256 shear relax time 6 = 300
257 shear relax time 7 = 350
258 shear relax time 8 = 400
259 shear relax time 9 = 450
260 shear relax time 10 = 500
261 wlf coef c1 = 1.0
262 wlf coef c2 = 1.0
263 wlf tref = 298
264 MAX POISSONS RATIO = 0.49
265 end parameters for model viscoelastic_swanson
266
267 end material graymatter
268
269 # -----
270 begin material helmetpad # formerly pads in Lud
271 # -----
272
273 # materials properties QC 2019-06-25 CBH with Materials for Casco document
274
275 density = 0.140 # g/cc
276
277 begin parameters for model elastic_plastic
278   youngs modulus = 7.00e7 # dyne/cm^2 = 7.00e6 Pa
279   poissons ratio = 0.20 # cm/cm = unitless
280   yield stress = 2.5e6 # dyne/cm^2 = 2.5e5 Pa
281   hardening modulus = 7.0e6 # dyne/cm^2 = 7.0e5 Pa
282   beta = 0.5 # unitless
283 end parameters for model elastic_plastic
284
285 end material helmetpad
286
287 # -----
288 begin material helmetpadhard # from cyl-1mm-non-0305-hard-038.i
289 # -----
290
291 # reference: Morse 2014, C. Franck Lacrosse paper
292 # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3

```

```

293 # materials properties QC 2019-06-25 CBH with Materials for Casco document
294 # for placeholder elastic and Swanson viscoelastic
295
296 # density = 0.059 # g/cc, 14.84 g / 250.518 cc = 0.059
297 density = 1.103e-1 # g/cc # 2019-08-09 match Sushant measured mass
298
299 begin parameters for model elastic
300   youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
301   poissons ratio = 0.45 # cm/cm = unitless
302 end parameters for model elastic
303
304 begin parameters for model elastic_plastic # note, parameters from test data at 20/s
305   youngs modulus = 4.66e7 # dyne/cm^2 = 4.66 MPa
306   poissons ratio = 0.1 # cm/cm = unitless
307   yield stress = 0.21e7 # dyne/cm^2 = 0.21 MPa
308   beta = 1.0 # unitless
309   # hardening modulus = 0.025e7 # dyne/cm^2 = 0.025 MPa
310   hardening modulus = 0.025e3 # dyne/cm^2 = 0.0000025 MPa, want close to flat
311 end parameters for model elastic_plastic
312
313 begin parameters for model viscoelastic_swanson
314   bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
315   # shear modulus = 3.400e5 # dyne/cm^2
316   A1 = 3.400e5 # dyne/cm^2
317   P1 = 0
318   B1 = 0
319   Q1 = 0
320   C1 = 0
321   R1 = 0
322   cut off strain = 0.05
323   prony shear infinity = 1.8824e-1 # Pa/Pa
324   prony shear 1 = 8.1176e-1 # Pa/Pa
325   prony shear 2 = 1.0e-4
326   prony shear 3 = 1.0e-4
327   prony shear 4 = 1.0e-4
328   prony shear 5 = 1.0e-4
329   prony shear 6 = 1.0e-4
330   prony shear 7 = 1.0e-4
331   prony shear 8 = 1.0e-4
332   prony shear 9 = 1.0e-4
333   prony shear 10 = 1.0e-4
334   shear relax time 1 = 2.50e-2 # seconds, formerly 1.4286e-3 s
335   shear relax time 2 = 100
336   shear relax time 3 = 150
337   shear relax time 4 = 200
338   shear relax time 5 = 250
339   shear relax time 6 = 300
340   shear relax time 7 = 350
341   shear relax time 8 = 400
342   shear relax time 9 = 450
343   shear relax time 10 = 500
344   wlf coef c1 = 1.0
345   wlf coef c2 = 1.0
346   wlf tref = 298

```

```

347     MAX POISSONS RATIO      = 0.49
348 end parameters for model viscoelastic_swanson
349
350 begin parameters for model orthotropic_crush
351     youngs modulus = 1.00e+07
352     poissons ratio = 0.00
353
354     ex = 1.00e+07
355     ey = 1.00e+07
356     ez = 1.00e+07
357     gxy = 0.50e+07
358     gyx = 0.50e+07
359     gzx = 0.50e+07
360
361     crush xx = crush_hard
362     crush yy = crush_hard
363     crush zz = crush_hard
364     crush xy = crush_hard
365     crush yz = crush_hard
366     crush zx = crush_hard
367
368     vmin = 0.70
369
370     yield stress = 28.967650904744864e+07
371
372 end parameters for model orthotropic_crush
373
374 end material helmetpadhard
375
376 # -----
377 begin material helmetpadsoft # from cyl-1mm-non-0305-soft-039.i
378 # -----
379
380 # reference: Morse 2014, C. Franck Lacrosse paper
381 # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3
382 # materials properties QC 2019-06-25 CBH with Materials for Casco document
383 # for placeholder elastic and Swanson viscoelastic
384
385 # density = 0.056 # g/cc, 14.15 g / 250.518 cc = 0.059
386 density = 1.047e-1 # g/cc # 2019-08-09 match Sushant measured mass
387
388 begin parameters for model elastic
389     youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
390     poissons ratio = 0.45 # cm/cm = unitless
391 end parameters for model elastic
392
393 begin parameters for model elastic_plastic # note, parameters from test data at 20/s
394     youngs modulus = 0.85e7 # dyne/cm^2 = 0.85 MPa
395     poissons ratio = 0.1 # cm/cm = unitless
396     yield stress = 0.048e7 # dyne/cm^2 = 0.048 MPa
397     beta = 1.0 # unitless
398     hardening modulus = 0.091e7 # dyne/cm^2 = 0.091 MPa
399 end parameters for model elastic_plastic
400

```

```

401 begin parameters for model viscoelastic_swanson
402     bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
403     # shear modulus = 3.400e5 # dyne/cm^2
404     A1 = 3.400e5 # dyne/cm^2
405     P1 = 0
406     B1 = 0
407     Q1 = 0
408     C1 = 0
409     R1 = 0
410     cut off strain = 0.05
411     prony shear infinity = 1.8824e-1 # Pa/Pa
412     prony shear 1 = 8.1176e-1 # Pa/Pa
413     prony shear 2 = 1.0e-4
414     prony shear 3 = 1.0e-4
415     prony shear 4 = 1.0e-4
416     prony shear 5 = 1.0e-4
417     prony shear 6 = 1.0e-4
418     prony shear 7 = 1.0e-4
419     prony shear 8 = 1.0e-4
420     prony shear 9 = 1.0e-4
421     prony shear 10 = 1.0e-4
422     shear relax time 1 = 2.50e-2 # seconds, formerly 1.4286e-3 s
423     shear relax time 2 = 100
424     shear relax time 3 = 150
425     shear relax time 4 = 200
426     shear relax time 5 = 250
427     shear relax time 6 = 300
428     shear relax time 7 = 350
429     shear relax time 8 = 400
430     shear relax time 9 = 450
431     shear relax time 10 = 500
432     wlf coef c1 = 1.0
433     wlf coef c2 = 1.0
434     wlf tref = 298
435     MAX POISSONS RATIO = 0.49
436 end parameters for model viscoelastic_swanson
437
438 begin parameters for model orthotropic_crush
439     youngs modulus = 6.0e+06
440     poissons ratio = 0.00
441
442     ex = 6.000e+06
443     ey = 6.000e+06
444     ez = 6.000e+06
445     gxy = 3.00e+06
446     gyx = 3.00e+06
447     gzx = 3.00e+06
448
449     crush xx = crush_soft
450     crush yy = crush_soft
451     crush zz = crush_soft
452     crush xy = crush_soft
453     crush yz = crush_soft
454     crush zx = crush_soft

```

```

455
456     vmin = 0.7
457
458     yield stress = 1.00e+08
459
460 end parameters for model orthotropic_crush
461
462 end material helmetpadsoft
463
464 # -----
465 begin material helmetshell # formerly helmet in Lud
466 # -----
467
468 # materials properties QC 2019-06-25 CBH with Materials for Casco document
469 # density          = 1.440 # g/cc
470 density          = 9.250e-1 # g/cc # 2019-08-09 match Sushant measured mass
471
472 begin parameters for model elastic_plastic
473     youngs modulus = 1.0e12 # dyne/cm^2 = 1.0e11 Pa
474     poissons ratio = 0.30 # cm/cm = unitless
475     yield stress   = 8.2e9 # dyne/cm^2 = 8.2e8 Pa
476     hardening modulus = 1.36e10 # dyne/cm^2 = 1.36e9 Pa
477     beta = 0.5          # unitless
478 end parameters for model elastic_plastic
479
480 end material helmetshell
481
482 # -----
483 begin material hemi
484 # -----
485
486 # 304 SS per Sushant
487
488 density = 8.00 # g/cc
489
490 begin parameters for model elastic
491
492     youngs modulus = 193.0e10 # dyne/cm^2 = 10e6 psi = 193.0 GPa
493     poissons ratio = 0.29    # cm/cm      = unitless
494
495 end parameters for model elastic
496
497 end material hemi
498
499 # -----
500 begin material magnesium
501 # -----
502
503 # MEP pad, cf Malave email 2018-11-20-1232
504 # DOT mass = 3226.4 g
505 # travel arm assy (arm, mount, clamp) = 1773.6 g
506 # asm.matweb.com, magnesium alloys, general
507 # materials properties QC 2019-06-25 CBH with Materials for Casco document
508

```

```

509  # density = 1.800 # g/cc
510  density = 2.080 # g/cc, increase to make sim mass match exp mass
511
512  begin parameters for model elastic
513
514      youngs modulus      =  4.050e11 # dyne/cm^2 = 4.50e10 Pa
515      poissons ratio      =  0.35      # cm/cm      = unitless
516
517  end parameters for model elastic
518
519  end material magnesium
520
521  # -----
522  begin material membrane # formerly ft for falx-tentorium in Lud
523  # -----
524
525  # materials properties QC 2019-06-25 CBH with Materials for Casco document
526
527  density = 1.133 # g/cc
528
529  begin parameters for model elastic
530      youngs modulus      =  3.15e8 # dyne/cm^2 = 3.15e7 Pa
531      poissons ratio      =  0.45      # cm/cm      = unitless
532  end parameters for model elastic
533
534  end material membrane
535
536  # -----
537  begin material meppad
538  # -----
539
540  # MEP pad, cf
541  # Malave email 2018-11-20-1232
542  # Fawzi email 2018-10-31-1036
543  # Cadex created MEP from cast polyurethane called Triathane from
544  # Crosslink Technology Inc Hardness 60 Shore A (not MEP
545  # made from neoprene rubber), tensile modulus at 100% elongation
546  # is 397 psi = 2.737e7 dyne/cm^2, assume same in tension and compression
547  # materials properties QC 2019-06-25 CBH with Materials for Casco document
548
549  # density = 1.030 # g/cc, should yield pad mass 467.65 g
550  density = 1.010 # g/cc, should yield pad mass 467.65 g
551  # update to make sim mass match exp mass
552
553  begin parameters for model elastic
554
555      # youngs modulus      =  2.4633e7 # dyne/cm^2 # s1
556      # youngs modulus      =  2.737e7 # dyne/cm^2 # s0 s3 s4
557      # youngs modulus      =  3.0107e7 # dyne/cm^2 # s2 s5
558      # youngs modulus      =  3.2844e7 # dyne/cm^2 # s6
559      # youngs modulus      =  3.5581e7 # dyne/cm^2 # s7
560      # youngs modulus      =  3.8318e7 # dyne/cm^2 # s8
561      youngs modulus      =  5.4740e7 # dyne/cm^2 # s9
562      # poissons ratio      =  0.36      # cm/cm      = unitless # s3

```

```

563      # poissons ratio      =  0.40    # cm/cm      = unitless # s0 s1 s2
564      # poissons ratio      =  0.44    # cm/cm      = unitless # s4 s5
565      # poissons ratio      =  0.48    # cm/cm      = unitless # s6
566      poissons ratio      =  0.49    # cm/cm      = unitless # s7 s8 s9
567
568  end parameters for model elastic
569
570 end material meppad
571
572 # -----
573 begin material mepplate
574 # -----
575
576 # MEP plate, cf Malave email 2018-11-20-1232
577 # Note: MEP (pad + plate) = 974.85 g (combined)
578 # materials properties QC 2019-06-25 CBH with Materials for Casco document
579
580 # density = 2.700 # g/cc, should yield plate mass 507.195 g
581 density = 2.540 # g/cc, should yield plate mass 507.195 g
582                         # update to make sim mass match exp mass
583
584 begin parameters for model elastic
585
586     youngs modulus      =  6.89e11 # dyne/cm^2 = 10e6 psi = 6.89e10 Pa
587     poissons ratio      =  0.33    # cm/cm      = unitless
588
589 end parameters for model elastic
590
591 end material mepplate
592
593 # -----
594 begin material muscle # exactly copy of skin for now 2019-02-18
595                         # needs migration to match CTH Swanson model usage
596 # -----
597
598 # materials properties QC 2019-06-25 CBH with Materials for Casco document
599
600 density = 1.200 # g/cc
601
602 begin parameters for model elastic
603     youngs modulus      =  1.67e8 # dyne/cm^2 = 1.67e7 Pa
604     poissons ratio      =  0.42    # cm/cm      = unitless
605 end parameters for model elastic
606
607 end material muscle
608
609 # -----
610 begin material sinus
611 # -----
612
613 density = 1.218e-3 # g/cc
614
615 begin parameters for model elastic
616     youngs modulus      =  1398.e4 # dyne/cm^2 = 1.398 MPa

```

```

617     poissos ratio = 0.49    # cm/cm = unitless
618 end parameters for model elastic
619
620 end material sinus
621
622 # -----
623 begin material skin
624 # -----
625
626 # reference: Morse 2014, C. Franck Lacrosse paper
627 # skin: neo-Hookean, K=34.7 MPa, mu = 5880 kPa; rho = 1040 kg/m^3
628 # materials properties QC 2019-06-25 CBH with Materials for Casco document
629
630 density = 1.200 # g/cc
631
632 begin parameters for model elastic
633     youngs modulus = 1.67e8 # dyne/cm^2 = 1.67e7 Pa
634     poissos ratio = 0.42    # cm/cm = unitless
635 end parameters for model elastic
636
637 end material skin
638
639 # -----
640 begin material whitematter
641 # -----
642
643 # reference: Morse 2014, C. Franck Lacrosse paper
644 # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3
645 # materials properties QC 2019-06-25 CBH with Materials for Casco document
646 # for placeholder elastic and Swanson viscoelastic
647
648 density = 1.040 # g/cc
649
650 # placeholder prior to Swanson
651 # begin parameters for model elastic
652 #     youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
653 #     poissos ratio = 0.45    # cm/cm = unitless
654 # end parameters for model elastic
655
656 begin parameters for model viscoelastic_swanson
657     bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
658 #     shear modulus = 4.100e5 # dyne/cm^2
659     A1          = 4.100e5 # dyne/cm^2
660     P1          = 0
661     B1          = 0
662     Q1          = 0
663     C1          = 0
664     R1          = 0
665     cut off strain = 0.05
666     prony shear infinity = 1.9024e-1 # Pa/Pa
667     prony shear 1 = 8.0976e-1 # Pa/Pa
668     prony shear 2 = 1.0e-4
669     prony shear 3 = 1.0e-4
670     prony shear 4 = 1.0e-4

```

```

671      prony shear 5      = 1.0e-4
672      prony shear 6      = 1.0e-4
673      prony shear 7      = 1.0e-4
674      prony shear 8      = 1.0e-4
675      prony shear 9      = 1.0e-4
676      prony shear 10     = 1.0e-4
677      # shear relax time 1 = 2.50e-2 # seconds, formerly 1.4286e-3 s
678      shear relax time 1 = 1.4286e-3 # seconds, investigate quick relaxation
679      shear relax time 2 = 100
680      shear relax time 3 = 150
681      shear relax time 4 = 200
682      shear relax time 5 = 250
683      shear relax time 6 = 300
684      shear relax time 7 = 350
685      shear relax time 8 = 400
686      shear relax time 9 = 450
687      shear relax time 10 = 500
688      wlf coef c1       = 1.0
689      wlf coef c2       = 1.0
690      wlf tref           = 298
691      MAX POISSONS RATIO = 0.49
692  end parameters for model viscoelastic_swanson
693
694  end material whitematter
695
696  # -----
697  # mesh
698  # -----
699
700  begin finite element model crush
701
702      database name = ../../geometry/data/bob-1mm-5kg-helmet-hemi/bob-1mm-5kg-helmet2-hemi.g
703      # database name = /projects/sibl/geometry/data/bob-1mm-5kg-helmet-hemi/bob-1mm-5kg-helmet2-hemi.g
704      # database name = /projects/sibl/geometry/data/bob-1mm-5kg-helmet-hemi/bob-1mm-5kg-helmet-hemi.g
705      # database name = /projects/sibl/geometry/data/bob-2mm-5kg-helmet-mep/bob-2mm-5kg-helmet-mep.g
706      # database name = /projects/sibl/geometry/data/bob-2mm-5kg-non-mep/bob-2mm-5kg-non-mep.g
707      # database name = /projects/sibl/geometry/data/bob-2mm-non-mep/bob-2mm-non-mep.g
708      # database name = /projects/sibl/geometry/data/bob-1mm-non-mep/bob-1mm-non-mep.g
709      # database name = /projects/sibl/geometry/data/bob-non-mep/bob-2mm-non-mep.g
710      # database name = /projects/sibl/geometry/data/dot-sphere-mep/dot_sphere_mep.g
711      # database name = dot_sphere_mep.g
712      # database name = sphere.g
713      # database name = quarter_sphere_block_target.g
714      # database name = quarter_sphere.g
715      # database name = foam_cube_001.g
716      # database name = /projects/sibl/casco/geo/foam_cube_001.g
717  database type = exodusII
718
719  # -----
720  # Bob material blocks
721  # -----
722  # with bob-2mm-5kg-non-0305-mep-031, block 5 (larynx) no longer exists
723  # begin parameters for block block_1 block_2 block_3 block_4 \#
724  #   block_5 block_6 block_7 block_8 block_9 block_10

```

```

725      #
726      #   material = magnesium
727      #   model = elastic
728      #
729      # end parameters for block block_1 block_2 block_3 block_4 \#
730      #   block_5 block_6 block_7 block_8 block_9 block_10
731
732      # -----
733      begin parameters for block block_1 # bone
734
735          material = bone
736          # model = elastic
737          model = elastic_plastic
738
739      end parameters for block block_1
740
741      # -----
742      begin parameters for block block_2 # disc
743
744          material = disc
745          model = elastic
746
747      end parameters for block block_2
748
749      # -----
750      begin parameters for block block_3 # vasculature
751
752          material = csf # temporarily model as csf
753          model = elastic
754
755      end parameters for block block_3
756
757      # -----
758      begin parameters for block block_4 # airway.sinus
759
760          material = sinus
761          model = elastic
762
763      end parameters for block block_4
764
765      # -----
766      # with bob-2mm-5kg-non-0305-mep-031, block 5 (larynx) no longer exists
767      # begin parameters for block block_5 # larynx
768
769      #   material = sinus # temporarily model as sinus
770      #   model = elastic
771
772      # end parameters for block block_5
773
774      # -----
775      # begin parameters for block block_6 # membrane
776      begin parameters for block block_5 # membrane
777
778          material = membrane

```

```

779     model = elastic
780
781 # end parameters for block block_6
782 end parameters for block block_5
783
784 # -----
785 # begin parameters for block block_7 # CSF
786 begin parameters for block block_6 # CSF
787
788     material = csf
789     model = elastic
790
791 # end parameters for block block_7
792 end parameters for block block_6
793
794 # -----
795 # begin parameters for block block_8 # white matter
796 begin parameters for block block_7 # white matter
797
798     material = whitematter
799     model = viscoelastic_swanson
800
801 # end parameters for block block_8
802 end parameters for block block_7
803
804 # -----
805 # begin parameters for block block_9 # gray matter
806 begin parameters for block block_8 # gray matter
807
808     material = graymatter
809     model = viscoelastic_swanson
810
811 # end parameters for block block_9
812 end parameters for block block_8
813
814 # -----
815 # begin parameters for block block_10 # muscle
816 begin parameters for block block_9 # muscle
817
818     material = muscle
819     model = elastic
820
821 # end parameters for block block_10
822 end parameters for block block_9
823
824 # -----
825 # begin parameters for block block_11 # skin
826 begin parameters for block block_10 # skin
827
828     material = skin
829     model = elastic
830
831 # end parameters for block block_11
832 end parameters for block block_10

```

```

833
834     # -----
835     # Hemispherical Target
836     # -----
837     begin parameters for block block_12 # hemi
838
839         material = hemi
840         model = elastic
841
842     end parameters for block block_12
843
844     # -----
845     # Helmet Shell - Kevlar
846     # -----
847     begin parameters for block block_20
848
849         material = helmetshell
850         model = elastic_plastic
851
852     end parameters for block block_20
853
854     # -----
855     # Helmet Foam - Hard
856     # -----
857     begin parameters for block block_21 block_22 block_23 block_24 block_25
858
859         material = helmetpadhard
860         # model = elastic_plastic
861         model = orthotropic_crush
862
863     end parameters for block block_21 block_22 block_23 block_24 block_25
864
865     # -----
866     # Helmet Foam - Soft
867     # -----
868     begin parameters for block block_211 block_221 block_231 block_241 block_251
869
870         material = helmetpadsoft
871         # model = elastic_plastic
872         model = orthotropic_crush
873
874     end parameters for block block_211 block_221 block_231 block_241 block_251
875
876
877
878     end finite element model crush
879
880     # -----
881     # procedures
882     # -----
883
884     begin presto procedure presto_procedure
885
886     # -----

```

```

887     # time control
888     # -----
889
890     begin time control
891
892         begin time stepping block phase_1
893
894             start time = 0.000 # second
895
896             begin parameters for presto region presto_region
897                 time step scale factor = 1.0 # unitless
898             end parameters for presto region presto_region
899
900         end time stepping block phase_1
901
902         # termination time = 0.0001 # second
903         # termination time = 0.001 # second
904         # termination time = 0.006 # second
905         # termination time = 0.008 # second
906         # termination time = 0.010 # second
907         # termination time = 0.012 # second
908         # termination time = 0.020 # second
909         # termination time = 0.035 # second
910         # termination time = 0.040 # second
911         # termination time = 0.060 # second
912         # termination time = 0.100 # second
913
914     end time control
915
916     # -----
917     # presto region
918     # -----
919
920     begin presto region presto_region
921
922         use finite element model crush
923
924         # -----
925         # to locate c.g. of Bob
926         # -----
927     begin mass properties
928         block = block_1 block_2 block_3 block_4 block_5 block_6 block_7 \#
929             block_8 block_9 block_10
930         structure name = assembly_bob
931     end mass properties
932
933     # -----
934     # to tally the hard foam mass
935     # -----
936     begin mass properties
937         block = block_21 block_22 block_23 block_24 block_25
938         structure name = assembly_hardfoam
939     end mass properties
940

```

```

941 # -----
942 # to tally the soft foam mass
943 #
944 begin mass properties
945   block = block_211 block_221 block_231 block_241 block_251
946   structure name = assembly_softfoam
947 end mass properties
948
949 # -----
950 # to tally the hard/soft foam pad assembly mass
951 #
952 begin mass properties
953   block = block_21  block_22  block_23  block_24  block_25 \
954           block_211 block_221 block_231 block_241 block_251
955   structure name = assembly_allfoam
956 end mass properties
957
958 # -----
959 # begin prescribed velocity
960 #  node set = nodelist_1
961 #  direction = z_axis
962 #  function = constant_velocity
963 #  scale factor = -1.0
964 # end prescribed velocity
965
966 # -----
967 # bottom of plate complete fixity
968 #
969 begin fixed displacement
970   node set = nodelist_1
971   components = x y z
972 end fixed displacement
973
974 # -----
975 # symmetry boundary condition
976 #
977 # begin fixed displacement
978 #  node set = nodelist_2
979 #  components = y
980 # end fixed displacement
981
982 # ----- (page 543/968 SM 4.48)
983 begin initial velocity
984
985   # node set commands
986   block = block_1 block_2 block_3 block_4 \
987           block_5 block_6 block_7 block_8 block_9 block_10 \
988           block_20 block_21 block_22 block_23 block_24 block_25 \
989           block_211 block_221 block_231 block_241 block_251
990
991   # direction specification commands
992   # component = z # vertical axis direction
993   direction = z_positive
994

```

```

995     # magnitude specification commands
996     magnitude = 304.8 # cm/s = 10 ft/s
997     # magnitude = 1.0 # cm/s
998
999     end initial velocity
1000
1001     # -----
1002     begin gravity
1003         function = gravity_accel
1004         direction = z_positive
1005         gravitational constant = 981.0 # cm/s^2
1006         # gravitational constant = 0.0 # cm/s^2
1007     end gravity
1008
1009     # -----
1010     begin contact definition contact_definition
1011         # User manual page 623/968
1012         # DASH search algorithm activates augmented Lagrange enforcement, and
1013         # ACME search algorithm activates kinematic enforcement
1014         # see Section 8.9
1015         search = dash
1016         skin all blocks = on
1017
1018         begin tied model tied_model
1019         end tied model tied_model
1020
1021         begin constant friction model friction_model
1022             friction coefficient = 0.2
1023         end constant friction model friction_model
1024
1025         begin interaction defaults
1026             general contact = on
1027             self contact = off
1028             friction model = friction_model
1029         end interaction defaults
1030
1031         initial overlap removal = on
1032
1033         begin remove initial overlap
1034             overlap normal tolerance = 0.1
1035             overlap tangential tolerance = 0.1
1036             overlap iterations = 100
1037             debug iteration plot = off
1038         end remove initial overlap
1039
1040         #
1041         begin constant friction model friction_model
1042             friction coefficient = 0.2
1043         end constant friction model friction_model
1044
1045
1046         begin interaction foam_to_helmet
1047             surfaces      = block_20 block_21 block_22 block_23 block_24 block_25
1048             friction model = tied_model

```

```

1049         end interaction foam_to_helmet
1050
1051
1052     #
1053     end contact definition contact_definition
1054
1055     # -----
1056
1057     {include("../include/e_vm-variables.i")}
1058     {include("../include/e_dot_vm-variables.i")}
1059
1060     # -----
1061
1062     # -----
1063     begin element death dead_ele
1064         INCLUDE ALL BLOCKS
1065         death on inversion = on
1066         death on ill defined contact = on
1067         death steps = 5
1068         force valid acme connectivity
1069     end element death dead_ele
1070
1071     # -----
1072
1073     begin results output field_exodus
1074
1075         database name = output_field.e
1076         database type = exodusII
1077         # at time 0.0 increment = 0.00002 # seconds
1078         # at time 0.0 increment = 0.0001 # seconds
1079         at time 0.0 increment = 0.0002 # seconds
1080         # at time 0.0 increment = 0.010 # seconds
1081
1082         # -----
1083         # kinematics
1084
1085         nodal variables = coordinates as x
1086         nodal variables = displacement as displvec
1087         nodal variables = velocity as v
1088
1089         # -----
1090         # stress
1091
1092         element variables = von_mises # real
1093         element variables = hydrostatic_stress # real
1094         element variables = fluid_pressure # real
1095
1096         # -----
1097         # strain
1098
1099         element variables = min_principal_log_strain # real
1100         element variables = max_principal_log_strain # real
1101         element variables = max_shear_log_strain # real
1102         element variables = min_principal_green_lagrange_strain # real

```

```

1103     element variables = max_principal_green_lagrange_strain # real
1104
1105 # -----
1106 # strain rate
1107 # -----
1108     element variables = min_principal_rate_of_deformation # real
1109     element variables = max_principal_rate_of_deformation # real
1110     element variables = min_principal_green_lagrange_strain_rate # real
1111     element variables = max_principal_green_lagrange_strain_rate # real
1112
1113 # -----
1114 # misc
1115 # -----
1116     element variables = overlap_volume_ratio
1117     element variables = death_status as dead_or_alive
1118
1119 # -----
1120 # invariants
1121 # -----
1122     element variables = e_vm      # real, von Mises of GL strain
1123     element variables = e_dot_vm # real, von Mises of GL strain rate
1124
1125 # -----
1126 # energy
1127 # -----
1128     global variables = timestep      as ts
1129     global variables = contact_energy as ce
1130     global variables = external_energy as ee
1131     global variables = internal_energy as ie
1132     global variables = kinetic_energy as ke
1133     global variables = hourglass_energy as hge # check this is near zero
1134     global variables = strain_energy   as se
1135     global variables = momentum       as mo
1136
1137 end results output field_exodus
1138
1139 # -----
1140 begin heartbeat output hscth_file
1141
1142     stream name = history.csv
1143     format = SpyHis
1144     at time 0.0 increment = 0.00002    # seconds, 50,000 Hz acquisition
1145 # at time 0.0 increment = 3.00003e-5 # seconds, 33,333 Hz acquisition
1146 # at time 0.0 increment = 0.0001      # seconds, 10,000 Hz acquisition
1147
1148 # -----
1149 # helmet
1150 # -----
1151     node coordinates  nearest location 33.61001, 21.349597, 26.401136 as rhel
1152     node displacement  nearest location 33.61001, 21.349597, 26.401136 as uhel
1153     node velocity      nearest location 33.61001, 21.349597, 26.401136 as vhel
1154     #
1155 # -----
1156 # Bob head superior-to-inferior every cm in z direction

```

```

1157  #
1158  # -----#
1159  #
1160  node displacement(z)    nearest location 34.0, 21.0, 23.0 as u23
1161  node velocity(z)       nearest location 34.0, 21.0, 23.0 as v23
1162  # node acceleration(z) nearest location 34.0, 21.0, 23.0 as a23
1163  #
1164  node displacement(z)    nearest location 34.0, 21.0, 22.0 as u22
1165  node velocity(z)       nearest location 34.0, 21.0, 22.0 as v22
1166  # node acceleration(z) nearest location 34.0, 21.0, 22.0 as a22
1167  #
1168  node displacement(z)    nearest location 34.0, 21.0, 21.0 as u21
1169  node velocity(z)       nearest location 34.0, 21.0, 21.0 as v21
1170  # node acceleration(z) nearest location 34.0, 21.0, 21.0 as a21
1171  #
1172  node displacement(z)    nearest location 34.0, 21.0, 20.0 as u20
1173  node velocity(z)       nearest location 34.0, 21.0, 20.0 as v20
1174  # node acceleration(z) nearest location 34.0, 21.0, 20.0 as a20
1175  #
1176  node displacement(z)    nearest location 34.0, 21.0, 19.0 as u19
1177  node velocity(z)       nearest location 34.0, 21.0, 19.0 as v19
1178  # node acceleration(z) nearest location 34.0, 21.0, 19.0 as a19
1179  #
1180  node displacement(z)    nearest location 34.0, 21.0, 18.0 as u18
1181  node velocity(z)       nearest location 34.0, 21.0, 18.0 as v18
1182  # node acceleration(z) nearest location 34.0, 21.0, 18.0 as a18
1183  #
1184  node displacement(z)    nearest location 34.0, 21.0, 17.0 as u17
1185  node velocity(z)       nearest location 34.0, 21.0, 17.0 as v17
1186  # node acceleration(z) nearest location 34.0, 21.0, 17.0 as a17
1187  #
1188  node displacement(z)    nearest location 34.0, 21.0, 16.0 as u16
1189  node velocity(z)       nearest location 34.0, 21.0, 16.0 as v16
1190  # node acceleration(z) nearest location 34.0, 21.0, 16.0 as a16
1191  #
1192  node displacement(z)    nearest location 34.0, 21.0, 15.0 as u15
1193  node velocity(z)       nearest location 34.0, 21.0, 15.0 as v15
1194  # node acceleration(z) nearest location 34.0, 21.0, 15.0 as a15
1195  #
1196  node displacement(z)    nearest location 34.0, 21.0, 14.0 as u14
1197  node velocity(z)       nearest location 34.0, 21.0, 14.0 as v14
1198  # node acceleration(z) nearest location 34.0, 21.0, 14.0 as a14
1199  #
1200  node displacement(z)    nearest location 34.0, 21.0, 13.0 as u13
1201  node velocity(z)       nearest location 34.0, 21.0, 13.0 as v13
1202  # node acceleration(z) nearest location 34.0, 21.0, 13.0 as a13
1203  #
1204  node displacement(z)    nearest location 34.0, 21.0, 12.0 as u12
1205  node velocity(z)       nearest location 34.0, 21.0, 12.0 as v12
1206  # node acceleration(z) nearest location 34.0, 21.0, 12.0 as a12
1207  #
1208  node displacement(z)    nearest location 34.0, 21.0, 11.0 as u11
1209  node velocity(z)       nearest location 34.0, 21.0, 11.0 as v11
1210  # node acceleration(z) nearest location 34.0, 21.0, 11.0 as a11

```

```

1211      #
1212      node displacement(z)    nearest location 34.0, 21.0, 10.0 as u10
1213      node velocity(z)      nearest location 34.0, 21.0, 10.0 as v10
1214      # node acceleration(z)  nearest location 34.0, 21.0, 10.0 as a10
1215      #
1216      #
1217      # Bob c.g. from assembly_bob begin
1218      #
1219      # Bob-1mm center of gravity
1220      node coordinates      nearest location 33.6315, 21.4892, 12.3570 as rcg
1221      node displacement     nearest location 33.6315, 21.4892, 12.3570 as ucg
1222      node velocity        nearest location 33.6315, 21.4892, 12.3570 as vcg
1223      # node acceleration   nearest location 33.6315, 21.4892, 12.3570 as acg
1224      #
1225      # Bob-2mm center of gravity
1226      # node coordinates     nearest location 33.6329, 21.4885, 12.3519 as rcg
1227      # node displacement    nearest location 33.6329, 21.4885, 12.3519 as ucg
1228      # node velocity       nearest location 33.6329, 21.4885, 12.3519 as vcg
1229      # node acceleration   nearest location 33.6329, 21.4885, 12.3519 as acg
1230      #
1231      # Bob c.g. from assembly_bob end
1232      #
1233      #
1234      #
1235      # Bob bone tracers angular velocity begin
1236      #
1237      # Bob-1mm
1238      # from origin O to point P, superior, posterior, midline skull
1239      node coordinates      nearest location 33.6315, 25.8, 21.3 as rOP
1240      node displacement     nearest location 33.6315, 25.8, 21.3 as uOP
1241      node velocity        nearest location 33.6315, 25.8, 21.3 as vOP
1242      #
1243      # from origin O to point Q, inferior, posterior, midline skull
1244      node coordinates      nearest location 33.6315, 29.5, 10.2 as rOQ
1245      node displacement     nearest location 33.6315, 29.5, 10.2 as uOQ
1246      node velocity        nearest location 33.6315, 29.5, 10.2 as vOQ
1247      #
1248      # from origin O to point R, superior, anterior, midline skull
1249      node coordinates      nearest location 33.6315, 14.6, 19.7 as rOR
1250      node displacement     nearest location 33.6315, 14.6, 19.7 as uOR
1251      node velocity        nearest location 33.6315, 14.6, 19.7 as vOR
1252      #
1253      #
1254      # Bob bone tracers angular velocity end
1255      #
1256
1257      end heartbeat output hscth_file
1258
1259      # See SSM User Guide v4.50, Section 9.6.1 Restart Options, page 777/988
1260      # Here is the most basic restart: restart data is written at the
1261      # last step of analysis or if SSM detects an internal error, such as
1262      # element inversion.
1263      #
1264      begin restart data restart_data

```

```

1265     database name = g.rtout  # the restart file
1266     at wall time 10h increment = 10h  # submit_script will run 20 h 20 m
1267     restart = auto
1268     end restart data restart_data
1269
1270     end presto region presto_region
1271
1272     end presto procedure presto_procedure
1273
1274     # -----
1275     # end
1276     # ---
1277
1278 end sierra simulation_name

```

submit_script

```

1 #!/bin/bash
2 sierra -T 30:20:00 --account FY180100 -j 160 --job-name bob_063f --pre
3   --run --post adagio -i bob-1mm-5kg-helmet2-0305-hemi-063f.i

```

F.1.6. Translational/Rotational Case (Bob-066b)

bob-1mm-5kg-helmet2-0305-hemi-066b.i

```

1 begin sierra simulation_name
2
3     # -----
4     # units: grams, centimeters, seconds
5     # ---
6
7     # -----
8     # include files
9     # -----
10
11 {include("../include/e_vm-functions.i")}
12 {include("../include/e_dot_vm-functions.i")}
13
14 # -----
15 # direction vectors
16 # -----
17
18 define point origin with coordinates 0.0 0.0 0.0
19 define point cg with coordinates 33.6315 21.4892 12.3570
20
21 define direction x_positive with vector 1.0 0.0 0.0
22 define direction y_positive with vector 0.0 1.0 0.0

```

```

23  define direction z_positive with vector 0.0 0.0 1.0
24
25  define axis x_axis with point origin direction x_positive
26  define axis y_axis with point origin direction y_positive
27  define axis z_axis with point origin direction z_positive
28
29  define axis cg_rotation_axis with point cg direction x_positive
30
31  # -----
32  # user functions
33  # -----
34
35  begin function gravity_accel
36      type is constant
37      begin values
38          1.0
39      end values
40  end function gravity_accel
41
42  begin function crush_soft
43      type is piecewise linear
44      abscissa = compaction
45      ordinate = stress
46      x scale = 1.0
47      x offset = 0.0
48      y scale = 1.0
49      y offset = 0.0
50      begin values
51          0.0977 10.05e5    # (s, dyne/cm2)
52          0.8    13.41e5 # (s, dyne/cm2)
53      end values
54  end function crush_soft
55
56  begin function crush_hard
57      type is piecewise linear
58      abscissa = compaction
59      ordinate = stress
60      x scale = 1.0
61      x offset = 0.0
62      y scale = 1.0
63      y offset = 0.0
64      begin values
65          0.0 17.4293e5    # (s, dyne/cm2)
66          1.0 32.4293e5 # (s, dyne/cm2)
67      end values
68  end function crush_hard
69
70  begin function constant_velocity
71      type is piecewise linear
72      abscissa = time_value
73      ordinate = velocity_value
74      x scale = 1.0
75      x offset = 0.0
76      y scale = 1.0

```

```

77     y offset = 0.0
78     begin values
79         0.0 0.0 # (s, cm/s)
80         0.001 20.0 # (s, cm/s)
81         1.0 20.0 # (s, cm/s)
82     end values
83 end function constant_velocity
84
85 begin function load_unload
86     type = piecewise linear
87     abscissa = time_value
88     ordinate = displacement_value
89     x scale = 1.0
90     x offset = 0.0
91     y scale = 1.0
92     y offset = 0.0
93     begin values
94         0.0 0.0
95         1.0 -1.0
96         2.0 0.0
97     end values
98 end function load_unload
99
100 begin function ramp
101     type = piecewise linear
102     abscissa = time_value
103     ordinate = displacement_value
104     x scale = 1.0
105     x offset = 0.0
106     y scale = 1.0
107     y offset = 0.0
108     begin values
109         0.0 0.0
110         1.0 -1.0 # negative to create compression
111     end values
112 end function ramp
113
114 {include("skull_rotate.txt")}
115
116 {include("Vely.txt")}
117
118 {include("VelZ.txt")}
119
120 # -----
121 # materials: listed in alphabetical order
122 # -----
123
124 # -----
125 begin material al6061t6
126 # -----
127
128 # asm.matweb.com, ma6061t6
129
130 density = 2.77 # g/cc = 0.098 lb/in^3

```

```

131
132 begin parameters for model elastic_plastic
133
134     youngs modulus      = 68.9e10 # dyne/cm^2 = 10e6 psi = 68.9 GPa
135     poissons ratio      = 0.33   # cm/cm      = unitless
136     yield stress        = 276.0e7 # dyne/cm^2 = 40e3 psi = 276 MPa
137     hardening modulus   = 68.9e7 # dyne/cm^2 = 10e3 psi = 68 MPa
138     beta                = 1.0    # unitless
139
140 end parameters for model elastic_plastic
141
142 end material al6061t6
143
144 # -----
145 begin material bone
146 # -----
147
148 # reference: Morse 2014, C. Franck Lacrosse paper
149 # bone: linear elastic, E = 6.5 GPa, nu = 0.45; rho = ?
150 # materials properties QC 2019-06-25 CBH with Materials for Casco document
151
152 density = 1.210 # g/cc
153
154 begin parameters for model elastic
155     youngs modulus      = 8.0e10 # dyne/cm^2 = 8.0 GPa
156     poissons ratio      = 0.22   # cm/cm      = unitless
157 end parameters for model elastic
158
159 begin parameters for model elastic_plastic
160     youngs modulus      = 8.0e10 # dyne/cm^2 = 8.0 GPa
161     poissons ratio      = 0.22   # cm/cm      = unitless
162     yield stress        = 6.40e8 # dyne/cm^2 = 64 MPa
163     beta                = 1.0    # 1.0 is isotropic hardening, 0.0 is kinematic
164     hardening modulus   = 0.0    # dyne/cm^2 (perfect plasticity)
165 end parameters for model elastic_plastic
166
167 begin parameters for model johnson_cook
168     youngs modulus      = 8.0e10 # dyne/cm^2 = 8.0 GPa
169     poissons ratio      = 0.22   # cm/cm      = unitless
170     yield stress        = 9.5e8  # dyne/cm^2 = 95 MPa
171     hardening constant  = 0.0    # unitless
172     hardening exponent  = 0.0    # unitless
173     rate constant       = 0.0    # unitless
174     reference rate     = 0.0    # unitless
175     edot_ref            = 1.0    # unitless
176     d1                  = 0.008 # unitless
177     d2                  = 0.0    # unitless
178     d3                  = 0.0    # unitless
179     d4                  = 0.0    # unitless
180     d5                  = 0.0    # unitless
181 end parameters for model johnson_cook
182
183 end material bone
184

```

```

185  # -----
186  begin material csf # cerebral spinal fluid
187  # -----
188
189  # materials properties QC 2019-06-25 CBH with Materials for Casco document
190
191  density = 1.004 # g/cc
192
193  begin parameters for model elastic
194      # 2019-07-16 starting with Bob-035, specific (K, G) because less
195      # sensitive than specifying nu ~ 0.499
196      #
197      # youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
198      # poissons ratio = 0.4987252 # cm/cm = unitless
199      #
200      # shear modulus = 5.000e3 # dyne/cm^2 = 500 Pa, matches Zhang 2001
201      # bulk modulus = 2.371e10 # dyne/cm^2 (match K for gray and white matter)
202  end parameters for model elastic
203
204  end material csf
205
206  # -----
207  begin material disc # see /cielo/sims/b1341_10.0 for rho, cs, possion, yield
208  # -----
209
210  # materials properties QC 2019-06-25 CBH with Materials for Casco document
211
212  density = 1.000 # g/cc
213
214  begin parameters for model elastic
215      youngs modulus = 4.27e8 # dyne/cm^2 = 42.7 MPa
216          # Yang 2016, Fig 2, 30-vo at 20%/s strain rate
217      poissons ratio = 0.40 # cm/cm = unitless
218  end parameters for model elastic
219
220  end material disc
221
222  # -----
223  begin material graymatter
224  # -----
225
226  # reference: Morse 2014, C. Franck Lacrosse paper
227  # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3
228  # materials properties QC 2019-06-25 CBH with Materials for Casco document
229  # for placeholder elastic and Swanson viscoelastic
230
231  density = 1.040 # g/cc
232
233  # placeholder prior to Swanson
234  # begin parameters for model elastic
235  #   youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
236  #   poissons ratio = 0.45 # cm/cm = unitless
237  # end parameters for model elastic
238

```

```

239 begin parameters for model viscoelastic_swanson
240     bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
241 #     shear modulus = 3.400e5 # dyne/cm^2
242     A1          = 3.400e5 # dyne/cm^2
243     P1          = 0
244     B1          = 0
245     Q1          = 0
246     C1          = 0
247     R1          = 0
248     cut off strain = 0.05
249     prony shear infinity = 1.8824e-1 # Pa/Pa
250     prony shear 1 = 8.1176e-1 # Pa/Pa
251     prony shear 2 = 1.0e-4
252     prony shear 3 = 1.0e-4
253     prony shear 4 = 1.0e-4
254     prony shear 5 = 1.0e-4
255     prony shear 6 = 1.0e-4
256     prony shear 7 = 1.0e-4
257     prony shear 8 = 1.0e-4
258     prony shear 9 = 1.0e-4
259     prony shear 10 = 1.0e-4
260 # shear relax time 1 = 2.50e-2 # seconds, formerly 1.4286e-3 s
261     shear relax time 1 = 1.4286e-3
262     shear relax time 2 = 100
263     shear relax time 3 = 150
264     shear relax time 4 = 200
265     shear relax time 5 = 250
266     shear relax time 6 = 300
267     shear relax time 7 = 350
268     shear relax time 8 = 400
269     shear relax time 9 = 450
270     shear relax time 10 = 500
271     wlf coef c1 = 1.0
272     wlf coef c2 = 1.0
273     wlf tref = 298
274     MAX POISSONS RATIO = 0.49
275 end parameters for model viscoelastic_swanson
276
277 end material graymatter
278
279 # -----
280 begin material helmetpad # formerly pads in Lud
281 # -----
282
283 # materials properties QC 2019-06-25 CBH with Materials for Casco document
284
285     density = 0.140 # g/cc
286
287 begin parameters for model elastic_plastic
288     youngs modulus = 7.00e7 # dyne/cm^2 = 7.00e6 Pa
289     poissons ratio = 0.20 # cm/cm = unitless
290     yield stress = 2.5e6 # dyne/cm^2 = 2.5e5 Pa
291     hardening modulus = 7.0e6 # dyne/cm^2 = 7.0e5 Pa
292     beta = 0.5          # unitless

```

```

293     end parameters for model elastic_plastic
294
295 end material helmetpad
296
297 # -----
298 begin material helmetpadhard # from cyl-1mm-non-0305-hard-038.i
299 # -----
300
301 # reference: Morse 2014, C. Franck Lacrosse paper
302 # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3
303 # materials properties QC 2019-06-25 CBH with Materials for Casco document
304 # for placeholder elastic and Swanson viscoelastic
305
306 # density = 0.059 # g/cc, 14.84 g / 250.518 cc = 0.059
307 density = 1.103e-1 # g/cc # 2019-08-09 match Sushant measured mass
308
309 begin parameters for model elastic
310     youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
311     poissons ratio = 0.45 # cm/cm = unitless
312 end parameters for model elastic
313
314 begin parameters for model elastic_plastic # note, parameters from test data at 20/s
315     youngs modulus = 4.66e7 # dyne/cm^2 = 4.66 MPa
316     poissons ratio = 0.1 # cm/cm = unitless
317     yield stress = 0.21e7 # dyne/cm^2 = 0.21 MPa
318     beta = 1.0 # unitless
319 #     hardening modulus = 0.025e7 # dyne/cm^2 = 0.025 MPa
320     hardening modulus = 0.025e3 # dyne/cm^2 = 0.0000025 MPa, want close to flat
321 end parameters for model elastic_plastic
322
323 begin parameters for model viscoelastic_swanson
324     bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
325 #     shear modulus = 3.400e5 # dyne/cm^2
326     A1 = 3.400e5 # dyne/cm^2
327     P1 = 0
328     B1 = 0
329     Q1 = 0
330     C1 = 0
331     R1 = 0
332     cut off strain = 0.05
333     prony shear infinity = 1.8824e-1 # Pa/Pa
334     prony shear 1 = 8.1176e-1 # Pa/Pa
335     prony shear 2 = 1.0e-4
336     prony shear 3 = 1.0e-4
337     prony shear 4 = 1.0e-4
338     prony shear 5 = 1.0e-4
339     prony shear 6 = 1.0e-4
340     prony shear 7 = 1.0e-4
341     prony shear 8 = 1.0e-4
342     prony shear 9 = 1.0e-4
343     prony shear 10 = 1.0e-4
344 #     shear relax time 1 = 2.50e-2 # seconds, formerly 1.4286e-3 s
345     shear relax time 1 = 1.4286e-3
346     shear relax time 2 = 100

```

```

347     shear relax time 3      = 150
348     shear relax time 4      = 200
349     shear relax time 5      = 250
350     shear relax time 6      = 300
351     shear relax time 7      = 350
352     shear relax time 8      = 400
353     shear relax time 9      = 450
354     shear relax time 10     = 500
355     wlf coef c1           = 1.0
356     wlf coef c2           = 1.0
357     wlf tref                = 298
358     MAX POISSONS RATIO     = 0.49
359 end parameters for model viscoelastic_swanson
360
361 begin parameters for model orthotropic_crush
362     youngs modulus = 1.00e+07
363     poisssons ratio = 0.00
364
365     ex = 1.00e+07
366     ey = 1.00e+07
367     ez = 1.00e+07
368     gxy = 0.50e+07
369     gyz = 0.50e+07
370     gzx = 0.50e+07
371
372     crush xx = crush_hard
373     crush yy = crush_hard
374     crush zz = crush_hard
375     crush xy = crush_hard
376     crush yz = crush_hard
377     crush zx = crush_hard
378
379     vmin = 0.70
380
381     yield stress = 28.967650904744864e+07
382
383 end parameters for model orthotropic_crush
384
385 end material helmetpadhard
386
387 # -----
388 begin material helmetpadsoft # from cyl-1mm-non-0305-soft-039.i
389 # -----
390
391 # reference: Morse 2014, C. Franck Lacrosse paper
392 # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3
393 # materials properties QC 2019-06-25 CBH with Materials for Casco document
394 # for placeholder elastic and Swanson viscoelastic
395
396 # density = 0.056 # g/cc, 14.15 g / 250.518 cc = 0.059
397 density = 1.047e-1 # g/cc # 2019-08-09 match Sushant measured mass
398
399 begin parameters for model elastic
400     youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa

```

```

401     poissos ratio = 0.45    # cm/cm = unitless
402 end parameters for model elastic
403
404 begin parameters for model elastic_plastic # note, parameters from test data at 20/s
405     youngs modulus = 0.85e7    # dyne/cm^2 = 0.85 MPa
406     poissos ratio = 0.1        # cm/cm = unitless
407     yield stress = 0.048e7    # dyne/cm^2 = 0.048 MPa
408     beta = 1.0                # unitless
409     hardening modulus = 0.091e7 # dyne/cm^2 = 0.091 MPa
410 end parameters for model elastic_plastic
411
412 begin parameters for model viscoelastic_swanson
413     bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
414     # shear modulus = 3.400e5 # dyne/cm^2
415     A1 = 3.400e5 # dyne/cm^2
416     P1 = 0
417     B1 = 0
418     Q1 = 0
419     C1 = 0
420     R1 = 0
421     cut off strain = 0.05
422     prony shear infinity = 1.8824e-1 # Pa/Pa
423     prony shear 1 = 8.1176e-1 # Pa/Pa
424     prony shear 2 = 1.0e-4
425     prony shear 3 = 1.0e-4
426     prony shear 4 = 1.0e-4
427     prony shear 5 = 1.0e-4
428     prony shear 6 = 1.0e-4
429     prony shear 7 = 1.0e-4
430     prony shear 8 = 1.0e-4
431     prony shear 9 = 1.0e-4
432     prony shear 10 = 1.0e-4
433     # shear relax time 1 = 2.50e-2 # seconds, formerly 1.4286e-3 s
434     shear relax time 1 = 1.4286e-3
435     shear relax time 2 = 100
436     shear relax time 3 = 150
437     shear relax time 4 = 200
438     shear relax time 5 = 250
439     shear relax time 6 = 300
440     shear relax time 7 = 350
441     shear relax time 8 = 400
442     shear relax time 9 = 450
443     shear relax time 10 = 500
444     wlf coef c1 = 1.0
445     wlf coef c2 = 1.0
446     wlf tref = 298
447     MAX POISSONS RATIO = 0.49
448 end parameters for model viscoelastic_swanson
449
450 begin parameters for model orthotropic_crush
451     youngs modulus = 6.0e+06
452     poissos ratio = 0.00
453
454     ex = 6.000e+06

```

```

455     ey = 6.000e+06
456     ez = 6.000e+06
457     gxy = 3.00e+06
458     gyz = 3.00e+06
459     gzx = 3.00e+06
460
461     crush xx = crush_soft
462     crush yy = crush_soft
463     crush zz = crush_soft
464     crush xy = crush_soft
465     crush yz = crush_soft
466     crush zx = crush_soft
467
468     vmin = 0.7
469
470     yield stress = 1.00e+08
471
472 end parameters for model orthotropic_crush
473
474 end material helmetpadsoft
475
476 # -----
477 begin material helmetshell # formerly helmet in Lud
478 # -----
479
480 # materials properties QC 2019-06-25 CBH with Materials for Casco document
481 # density = 1.440 # g/cc
482 density = 9.250e-1 # g/cc # 2019-08-09 match Sushant measured mass
483
484 begin parameters for model elastic_plastic
485     youngs modulus = 1.0e12 # dyne/cm^2 = 1.0e11 Pa
486     poissons ratio = 0.30 # cm/cm = unitless
487     yield stress = 8.2e9 # dyne/cm^2 = 8.2e8 Pa
488     hardening modulus = 1.36e10 # dyne/cm^2 = 1.36e9 Pa
489     beta = 0.5 # unitless
490 end parameters for model elastic_plastic
491
492 end material helmetshell
493
494 # -----
495 begin material hemi
496 # -----
497
498 # 304 SS per Sushant
499
500 density = 8.00 # g/cc
501
502 begin parameters for model elastic
503
504     youngs modulus = 193.0e10 # dyne/cm^2 = 10e6 psi = 193.0 GPa
505     poissons ratio = 0.29 # cm/cm = unitless
506
507 end parameters for model elastic
508

```

```

509 end material hemi
510
511 # -----
512 begin material magnesium
513 # -----
514
515 # MEP pad, cf Malave email 2018-11-20-1232
516 # DOT mass = 3226.4 g
517 # travel arm assy (arm, mount, clamp) = 1773.6 g
518 # asm.matweb.com, magnesium alloys, general
519 # materials properties QC 2019-06-25 CBH with Materials for Casco document
520
521 # density = 1.800 # g/cc
522 density = 2.080 # g/cc, increase to make sim mass match exp mass
523
524 begin parameters for model elastic
525
526     youngs modulus      = 4.050e11 # dyne/cm^2 = 4.50e10 Pa
527     poissons ratio       = 0.35      # cm/cm      = unitless
528
529 end parameters for model elastic
530
531 end material magnesium
532
533 # -----
534 begin material membrane # formerly ft for falx-tentorium in Lud
535 # -----
536
537 # materials properties QC 2019-06-25 CBH with Materials for Casco document
538
539 density = 1.133 # g/cc
540
541 begin parameters for model elastic
542     youngs modulus      = 3.15e8 # dyne/cm^2 = 3.15e7 Pa
543     poissons ratio       = 0.45      # cm/cm      = unitless
544 end parameters for model elastic
545
546 end material membrane
547
548 # -----
549 begin material meppad
550 # -----
551
552 # MEP pad, cf
553 # Malave email 2018-11-20-1232
554 # Fawzi email 2018-10-31-1036
555 # Cadex created MEP from cast polyurethane called Triathane from
556 # Crosslink Technology Inc Hardness 60 Shore A (not MEP
557 # made from neoprene rubber), tensile modulus at 100% elongation
558 # is 397 psi = 2.737e7 dyne/cm^2, assume same in tension and compression
559 # materials properties QC 2019-06-25 CBH with Materials for Casco document
560
561 # density = 1.030 # g/cc, should yield pad mass 467.65 g
562 density = 1.010 # g/cc, should yield pad mass 467.65 g

```

```

563          # update to make sim mass match exp mass
564
565 begin parameters for model elastic
566
567     # youngs modulus      =  2.4633e7 # dyne/cm^2 # s1
568     # youngs modulus      =  2.737e7 # dyne/cm^2 # s0 s3 s4
569     # youngs modulus      =  3.0107e7 # dyne/cm^2 # s2 s5
570     # youngs modulus      =  3.2844e7 # dyne/cm^2 # s6
571     # youngs modulus      =  3.5581e7 # dyne/cm^2 # s7
572     # youngs modulus      =  3.8318e7 # dyne/cm^2 # s8
573     youngs modulus      =  5.4740e7 # dyne/cm^2 # s9
574     # poissons ratio      =  0.36   # cm/cm      = unitless # s3
575     # poissons ratio      =  0.40   # cm/cm      = unitless # s0 s1 s2
576     # poissons ratio      =  0.44   # cm/cm      = unitless # s4 s5
577     # poissons ratio      =  0.48   # cm/cm      = unitless # s6
578     poissons ratio      =  0.49   # cm/cm      = unitless # s7 s8 s9
579
580 end parameters for model elastic
581
582 end material meppad
583
584 # -----
585 begin material mepplate
586 # -----
587
588 # MEP plate, cf Malave email 2018-11-20-1232
589 # Note: MEP (pad + plate) = 974.85 g (combined)
590 # materials properties QC 2019-06-25 CBH with Materials for Casco document
591
592 # density = 2.700 # g/cc, should yield plate mass 507.195 g
593 density = 2.540 # g/cc, should yield plate mass 507.195 g
594          # update to make sim mass match exp mass
595
596 begin parameters for model elastic
597
598     youngs modulus      =  6.89e11 # dyne/cm^2 = 10e6 psi = 6.89e10 Pa
599     poissons ratio      =  0.33   # cm/cm      = unitless
600
601 end parameters for model elastic
602
603 end material mepplate
604
605 # -----
606 begin material muscle # exactly copy of skin for now 2019-02-18
607          # needs migration to match CTH Swanson model usage
608 # -----
609
610 # materials properties QC 2019-06-25 CBH with Materials for Casco document
611
612 density = 1.200 # g/cc
613
614 begin parameters for model elastic
615     youngs modulus      =  1.67e8 # dyne/cm^2 = 1.67e7 Pa
616     poissons ratio      =  0.42   # cm/cm      = unitless

```

```

617     end parameters for model elastic
618
619 end material muscle
620
621 # -----
622 begin material sinus
623 # -----
624
625 density = 1.218e-3 # g/cc
626
627 begin parameters for model elastic
628     youngs modulus = 1398.e4 # dyne/cm^2 = 1.398 MPa
629     poissons ratio = 0.49 # cm/cm = unitless
630 end parameters for model elastic
631
632 end material sinus
633
634 # -----
635 begin material skin
636 # -----
637
638 # reference: Morse 2014, C. Franck Lacrosse paper
639 # skin: neo-Hookean, K=34.7 MPa, mu = 5880 kPa; rho = 1040 kg/m^3
640 # materials properties QC 2019-06-25 CBH with Materials for Casco document
641
642 density = 1.200 # g/cc
643
644 begin parameters for model elastic
645     youngs modulus = 1.67e8 # dyne/cm^2 = 1.67e7 Pa
646     poissons ratio = 0.42 # cm/cm = unitless
647 end parameters for model elastic
648
649 end material skin
650
651 # -----
652 begin material whitematter
653 # -----
654
655 # reference: Morse 2014, C. Franck Lacrosse paper
656 # general brain: neo-Hookean, K=2190 MPa, mu = 22.53 kPa; rho = 1040 kg/m^3
657 # materials properties QC 2019-06-25 CBH with Materials for Casco document
658 # for placeholder elastic and Swanson viscoelastic
659
660 density = 1.040 # g/cc
661
662 # placeholder prior to Swanson
663 # begin parameters for model elastic
664 #     youngs modulus = 1.50e8 # dyne/cm^2 = 15.0 MPa
665 #     poissons ratio = 0.45 # cm/cm = unitless
666 # end parameters for model elastic
667
668 begin parameters for model viscoelastic_swanson
669     bulk modulus = 2.371e10 # dyne/cm^2 = 2.371e9 Pa
670 #     shear modulus = 4.100e5 # dyne/cm^2

```

```

671      A1          =  4.100e5  # dyne/cm^2
672      P1          =  0
673      B1          =  0
674      Q1          =  0
675      C1          =  0
676      R1          =  0
677      cut off strain      =  0.05
678      prony shear infinity = 1.9024e-1  # Pa/Pa
679      prony shear 1        = 8.0976e-1  # Pa/Pa
680      prony shear 2        = 1.0e-4
681      prony shear 3        = 1.0e-4
682      prony shear 4        = 1.0e-4
683      prony shear 5        = 1.0e-4
684      prony shear 6        = 1.0e-4
685      prony shear 7        = 1.0e-4
686      prony shear 8        = 1.0e-4
687      prony shear 9        = 1.0e-4
688      prony shear 10       = 1.0e-4
689      # shear relax time 1 = 2.50e-2  # seconds, formerly 1.4286e-3 s
690      shear relax time 1  = 1.4286e-3
691      shear relax time 2  = 100
692      shear relax time 3  = 150
693      shear relax time 4  = 200
694      shear relax time 5  = 250
695      shear relax time 6  = 300
696      shear relax time 7  = 350
697      shear relax time 8  = 400
698      shear relax time 9  = 450
699      shear relax time 10 = 500
700      wlf coef c1        = 1.0
701      wlf coef c2        = 1.0
702      wlf tref            = 298
703      MAX POISSONS RATIO = 0.49
704      end parameters for model viscoelastic_swanson
705
706      end material whitematter
707
708      begin rigid body rigidSkull
709          reference location = 33.6315 21.4892 12.3570
710          # magnitude = 1000.0
711          # direction = y_positive
712          # angular velocity = 1570.0 # 90 deg rotation at t=0.001
713          # angular velocity = 1000.0
714          # cylindrical axis = cg_rotation_axis
715      end rigid body rigidSkull
716
717      begin solid section rigidSection
718          rigid body = rigidSkull
719      end solid section rigidSection
720
721
722      # -----
723      # mesh
724

```

```

725 # -----
726 # mesh
727 # ----
728
729 begin finite element model crush
730
731     database name = ../geometry/bob-1mm-5kg-helmet2-hemi.g
732 # database name = ../../geometry/data/bob-1mm-5kg-helmet-hemi/bob-1mm-5kg-helmet2-hemi.g
733 # database name = /projects/sibl/geometry/data/bob-1mm-5kg-helmet-hemi/bob-1mm-5kg-helmet2-hemi.g
734 # database name = /projects/sibl/geometry/data/bob-1mm-5kg-helmet-hemi/bob-1mm-5kg-helmet-hemi.g
735 # database name = /projects/sibl/geometry/data/bob-2mm-5kg-helmet-mep/bob-2mm-5kg-helmet-mep.g
736 # database name = /projects/sibl/geometry/data/bob-2mm-5kg-non-mep/bob-2mm-5kg-non-mep.g
737 # database name = /projects/sibl/geometry/data/bob-2mm-non-mep/bob-2mm-non-mep.g
738 # database name = /projects/sibl/geometry/data/bob-1mm-non-mep/bob-1mm-non-mep.g
739 # database name = /projects/sibl/geometry/data/bob-non-mep/bob-2mm-non-mep.g
740 # database name = /projects/sibl/geometry/data/dot-sphere-mep/dot_sphere_mep.g
741 # database name = dot_sphere_mep.g
742 # database name = sphere.g
743 # database name = quarter_sphere_block_target.g
744 # database name = quarter_sphere.g
745 # database name = foam_cube_001.g
746 # database name = /projects/sibl/casco/geo/foam_cube_001.g
747 database type = exodusII
748
749 omit block block_12 block_20 block_21 block_22 block_23 block_24 \#
750             block_25 block_211 block_221 block_231 block_241 block_251
751
752 # -----
753 # Bob material blocks
754 # -----
755 # with bob-2mm-5kg-non-0305-mep-031, block 5 (larynx) no longer exists
756 # begin parameters for block block_1 block_2 block_3 block_4 \#
757 #   block_5 block_6 block_7 block_8 block_9 block_10
758 #
759 #   material = magnesium
760 #   model = elastic
761 #
762 # end parameters for block block_1 block_2 block_3 block_4 \#
763 #   block_5 block_6 block_7 block_8 block_9 block_10
764
765 # -----
766 begin parameters for block block_1 # bone
767
768     material = bone
769 #   model = elastic
770     model = elastic_plastic
771     section = rigidSection
772
773 end parameters for block block_1
774
775 # -----
776 begin parameters for block block_2 # disc
777
778     material = disc

```

```

779     model = elastic
780     section = rigidSection
781
782 end parameters for block block_2
783
784 # -----
785 begin parameters for block block_3 # vasculature
786
787     material = csf # temporarily model as csf
788     model = elastic
789
790 end parameters for block block_3
791
792 # -----
793 begin parameters for block block_4 # airway.sinus
794
795     material = sinus
796     model = elastic
797     section = rigidSection
798
799 end parameters for block block_4
800
801 # -----
802 # with bob-2mm-5kg-non-0305-mep-031, block 5 (larynx) no longer exists
803 # begin parameters for block block_5 # larynx
804
805 #     material = sinus # temporarily model as sinus
806 #     model = elastic
807
808 # end parameters for block block_5
809
810 # -----
811 # begin parameters for block block_6 # membrane
812 begin parameters for block block_5 # membrane
813
814     material = membrane
815     model = elastic
816
817 # end parameters for block block_6
818 end parameters for block block_5
819
820 # -----
821 # begin parameters for block block_7 # CSF
822 begin parameters for block block_6 # CSF
823
824     material = csf
825     model = elastic
826
827 # end parameters for block block_7
828 end parameters for block block_6
829
830 # -----
831 # begin parameters for block block_8 # white matter
832 begin parameters for block block_7 # white matter

```

```

833     material = whitematter
834     model = viscoelastic_swanson
835
836
837 # end parameters for block block_8
838 end parameters for block block_7
839
840 # -----
841 # begin parameters for block block_9 # gray matter
842 begin parameters for block block_8 # gray matter
843
844     material = graymatter
845     model = viscoelastic_swanson
846
847 # end parameters for block block_9
848 end parameters for block block_8
849
850 # -----
851 # begin parameters for block block_10 # muscle
852 begin parameters for block block_9 # muscle
853
854     material = muscle
855     model = elastic
856     section = rigidSection
857
858 # end parameters for block block_10
859 end parameters for block block_9
860
861 # -----
862 # begin parameters for block block_11 # skin
863 begin parameters for block block_10 # skin
864
865     material = skin
866     model = elastic
867     section = rigidSection
868
869 # end parameters for block block_11
870 end parameters for block block_10
871
872 # -----
873 # Hemispherical Target
874 # -----
875 # begin parameters for block block_12 # hemi
876 #
877 #     material = hemi
878 #     model = elastic
879 #
880 #     end parameters for block block_12
881
882 #     # -----
883 #     # Helmet Shell - Kevlar
884 #     # -----
885 #     begin parameters for block block_20
886 #

```

```

887 #      material = helmetshell
888 #      model = elastic_plastic
889 #
890 #      end parameters for block block_20
891 #
892 #      # -----
893 #      # Helmet Foam - Hard
894 #      # -----
895 #      begin parameters for block block_21 block_22 block_23 block_24 block_25
896 #
897 #      material = helmetpadhard
898 #      # model = elastic_plastic
899 #      model = orthotropic_crush
900 #
901 #      end parameters for block block_21 block_22 block_23 block_24 block_25
902 #
903 #      # -----
904 #      # Helmet Foam - Soft
905 #      # -----
906 #      begin parameters for block block_211 block_221 block_231 block_241 block_251
907 #
908 #      material = helmetpadsoft
909 #      # model = elastic_plastic
910 #      model = orthotropic_crush
911 #
912 #      end parameters for block block_211 block_221 block_231 block_241 block_251
913
914
915
916 end finite element model crush
917
918 # -----
919 # procedures
920 # -----
921
922 begin presto procedure presto_procedure
923
924 # -----
925 # time control
926 # -----
927
928 begin time control
929
930 begin time stepping block phase_1
931
932 start time = 0.000 # second
933
934 begin parameters for presto region presto_region
935     time step scale factor = 1.0 # unitless
936 end parameters for presto region presto_region
937
938 end time stepping block phase_1
939
940 # termination time = 0.0001 # second

```

```

941     # termination time = 0.001 # second
942     # termination time = 0.006 # second
943     # termination time = 0.008 # second
944     # termination time = 0.010 # second
945     # termination time = 0.012 # second
946     # termination time = 0.025 # second
947     # termination time = 0.035 # second
948     # termination time = 0.040 # second
949     # termination time = 0.100 # second
950
951 end time control
952
953 # -----
954 # presto region
955 # -----
956
957 begin presto region presto_region
958
959     use finite element model crush
960
961 # -----
962 # to locate c.g. of Bob
963 # -----
964 begin mass properties
965     block = block_1 block_2 block_3 block_4 block_5 block_6 block_7 \#
966             block_8 block_9 block_10
967     structure name = assembly_bob
968 end mass properties
969
970 # -----
971 # to tally the hard foam mass
972 # -----
973 # begin mass properties
974 #     block = block_21 block_22 block_23 block_24 block_25
975 #     structure name = assembly_hardfoam
976 # end mass properties
977
978 # -----
979 # to tally the soft foam mass
980 # -----
981 # begin mass properties
982 #     block = block_211 block_221 block_231 block_241 block_251
983 #     structure name = assembly_softfoam
984 # end mass properties
985
986 # -----
987 # to tally the hard/soft foam pad assembly mass
988 # -----
989 # begin mass properties
990 #     block = block_21 block_22 block_23 block_24 block_25 \#
991 #             block_211 block_221 block_231 block_241 block_251
992 #     structure name = assembly_allfoam
993 # end mass properties
994

```

```

995  # -----
996  begin initial velocity
997
998      # node set commands
999      block = block_3 block_5 block_6 block_7 block_8
1000
1001      # direction specification commands
1002      # component = z # vertical axis direction
1003      direction = z_positive
1004
1005      # magnitude specification commands
1006      magnitude = 304.8 # cm/s = 10 ft/s
1007
1008  end initial velocity
1009
1010 begin prescribed rotational velocity
1011     rigid body = rigidSkull
1012     # block = block_1
1013     # cylindrical axis = cg_rotation_axis
1014     component = x
1015     function = skull_rotate
1016     scale factor = -1.0 # radians/sec
1017 end prescribed rotational velocity
1018
1019 begin prescribed velocity
1020     rigid body = rigidSkull
1021     direction = z_positive
1022     function = VelZ
1023     scale factor = 1.0    #
1024 end prescribed velocity
1025
1026 begin prescribed velocity
1027     rigid body = rigidSkull
1028     direction = y_positive
1029     function = VelY
1030     scale factor = -1.0    #
1031 end prescribed velocity
1032
1033 # begin prescribed velocity
1034 # block = block_1
1035 # node set subroutine = aupst_trans_rot_velocity
1036
1037 # -----
1038 # begin gravity
1039 # function = gravity_accel
1040 # direction = z_positive
1041 # gravitational constant = 981.0 # cm/s^2
1042 # # gravitational constant = 0.0 # cm/s^2
1043 # end gravity
1044
1045 # -----
1046 begin contact definition contact_definition
1047     # User manual page 623/968
1048     # DASH search algorithm activates augmented Lagrange enforcement, and

```

```

1049      # ACME search algorithm activates kinematic enforcement
1050      # see Section 8.9
1051      search = dash
1052      skin all blocks = on
1053
1054      begin tied model tied_model
1055      end tied model tied_model
1056
1057      begin constant friction model friction_model
1058          friction coefficient = 0.2
1059      end constant friction model friction_model
1060
1061      begin interaction defaults
1062          general contact = on
1063          self contact = off
1064          friction model = friction_model
1065      end interaction defaults
1066
1067      initial overlap removal = on
1068
1069      begin remove initial overlap
1070          overlap normal tolerance = 0.1
1071          overlap tangential tolerance = 0.1
1072          overlap iterations = 100
1073          debug iteration plot = off
1074      end remove initial overlap
1075
1076      #
1077      begin constant friction model friction_model
1078          friction coefficient = 0.2
1079      end constant friction model friction_model
1080
1081
1082      # begin interaction foam_to_helmet
1083      #     surfaces          = block_20 block_21 block_22 block_23 block_24 block_25
1084      #     friction model    = tied_model
1085      # end interaction foam_to_helmet
1086
1087
1088      #
1089      end contact definition contact_definition
1090
1091      # -----
1092
1093      {include("../include/e_vm-variables.i")}
1094      {include("../include/e_dot_vm-variables.i")}
1095
1096      # -----
1097
1098      # -----
1099      begin element death dead_ele
1100          INCLUDE ALL BLOCKS
1101          death on inversion = on
1102          death on ill defined contact = on

```

```

1103     death steps = 5
1104     force valid acme connectivity
1105 end element death dead_ele
1106 # -----
1107
1108 # -----
1109 begin results output field_exodus
1110
1111     database name = output_field.e
1112     database type = exodusII
1113 # at time 0.0 increment = 0.00002 # seconds
1114 # at time 0.0 increment = 0.0001 # seconds
1115     at time 0.0 increment = 0.0002 # seconds
1116 # at time 0.0 increment = 0.010 # seconds
1117
1118 # -----
1119 # kinematics
1120 # -----
1121     nodal variables = coordinates as x
1122     nodal variables = displacement as displvec
1123     nodal variables = velocity as v
1124
1125 # -----
1126 # stress
1127 # -----
1128     element variables = von_mises # real
1129     element variables = hydrostatic_stress # real
1130     element variables = fluid_pressure # real
1131
1132 # -----
1133 # strain
1134 # -----
1135     element variables = min_principal_log_strain # real
1136     element variables = max_principal_log_strain # real
1137     element variables = max_shear_log_strain # real
1138     element variables = min_principal_green_lagrange_strain # real
1139     element variables = max_principal_green_lagrange_strain # real
1140 #
1141 # -----
1142 # strain rate
1143 # -----
1144     element variables = min_principal_rate_of_deformation # real
1145     element variables = max_principal_rate_of_deformation # real
1146     element variables = min_principal_green_lagrange_strain_rate # real
1147     element variables = max_principal_green_lagrange_strain_rate # real
1148
1149 # -----
1150 # misc
1151 # -----
1152     element variables = overlap_volume_ratio
1153     element variables = death_status as dead_or_alive
1154
1155 # -----
1156 # invariants

```

```

1157      # -----
1158      element variables = e_vm      # real, von Mises of GL strain
1159      element variables = e_dot_vm # real, von Mises of GL strain rate
1160
1161      # -----
1162      # energy
1163      # -----
1164      global variables = timestep      as ts
1165      global variables = contact_energy as ce
1166      global variables = external_energy as ee
1167      global variables = internal_energy as ie
1168      global variables = kinetic_energy as ke
1169      global variables = hourglass_energy as hge # check this is near zero
1170      global variables = strain_energy   as se
1171      global variables = momentum       as mo
1172
1173  end results output field_exodus
1174
1175  # -----
1176 begin heartbeat output hscth_file
1177
1178      stream name = history.csv
1179      format = SpyHis
1180      at time 0.0 increment = 0.00002    # seconds, 50,000 Hz acquisition
1181      # at time 0.0 increment = 3.00003e-5 # seconds, 33,333 Hz acquisition
1182      # at time 0.0 increment = 0.0001     # seconds, 10,000 Hz acquisition
1183
1184      # -----
1185      # helmet
1186      # -----
1187      node coordinates  nearest location 33.61001, 21.349597, 26.401136 as rhel
1188      node displacement  nearest location 33.61001, 21.349597, 26.401136 as uhel
1189      node velocity      nearest location 33.61001, 21.349597, 26.401136 as vhel
1190      #
1191      # -----
1192      # Bob head superior-to-inferior every cm in z direction
1193      # -----
1194      # add 34.0 22.0 22.5 from CTH
1195      #
1196      node displacement(z)  nearest location 34.0, 21.0, 23.0 as u23
1197      node velocity(z)     nearest location 34.0, 21.0, 23.0 as v23
1198      # node acceleration(z) nearest location 34.0, 21.0, 23.0 as a23
1199      #
1200      node displacement(z)  nearest location 34.0, 21.0, 22.0 as u22
1201      node velocity(z)     nearest location 34.0, 21.0, 22.0 as v22
1202      # node acceleration(z) nearest location 34.0, 21.0, 22.0 as a22
1203      #
1204      node displacement(z)  nearest location 34.0, 21.0, 21.0 as u21
1205      node velocity(z)     nearest location 34.0, 21.0, 21.0 as v21
1206      # node acceleration(z) nearest location 34.0, 21.0, 21.0 as a21
1207      #
1208      node displacement(z)  nearest location 34.0, 21.0, 20.0 as u20
1209      node velocity(z)     nearest location 34.0, 21.0, 20.0 as v20
1210      # node acceleration(z) nearest location 34.0, 21.0, 20.0 as a20

```

```

1211  #
1212  node displacement(z)  nearest location 34.0, 21.0, 19.0 as u19
1213  node velocity(z)    nearest location 34.0, 21.0, 19.0 as v19
1214  # node acceleration(z)  nearest location 34.0, 21.0, 19.0 as a19
1215  #
1216  node displacement(z)  nearest location 34.0, 21.0, 18.0 as u18
1217  node velocity(z)    nearest location 34.0, 21.0, 18.0 as v18
1218  # node acceleration(z)  nearest location 34.0, 21.0, 18.0 as a18
1219  #
1220  node displacement(z)  nearest location 34.0, 21.0, 17.0 as u17
1221  node velocity(z)    nearest location 34.0, 21.0, 17.0 as v17
1222  # node acceleration(z)  nearest location 34.0, 21.0, 17.0 as a17
1223  #
1224  node displacement(z)  nearest location 34.0, 21.0, 16.0 as u16
1225  node velocity(z)    nearest location 34.0, 21.0, 16.0 as v16
1226  # node acceleration(z)  nearest location 34.0, 21.0, 16.0 as a16
1227  #
1228  node displacement(z)  nearest location 34.0, 21.0, 15.0 as u15
1229  node velocity(z)    nearest location 34.0, 21.0, 15.0 as v15
1230  # node acceleration(z)  nearest location 34.0, 21.0, 15.0 as a15
1231  #
1232  node displacement(z)  nearest location 34.0, 21.0, 14.0 as u14
1233  node velocity(z)    nearest location 34.0, 21.0, 14.0 as v14
1234  # node acceleration(z)  nearest location 34.0, 21.0, 14.0 as a14
1235  #
1236  node displacement(z)  nearest location 34.0, 21.0, 13.0 as u13
1237  node velocity(z)    nearest location 34.0, 21.0, 13.0 as v13
1238  # node acceleration(z)  nearest location 34.0, 21.0, 13.0 as a13
1239  #
1240  node displacement(z)  nearest location 34.0, 21.0, 12.0 as u12
1241  node velocity(z)    nearest location 34.0, 21.0, 12.0 as v12
1242  # node acceleration(z)  nearest location 34.0, 21.0, 12.0 as a12
1243  #
1244  node displacement(z)  nearest location 34.0, 21.0, 11.0 as u11
1245  node velocity(z)    nearest location 34.0, 21.0, 11.0 as v11
1246  # node acceleration(z)  nearest location 34.0, 21.0, 11.0 as a11
1247  #
1248  node displacement(z)  nearest location 34.0, 21.0, 10.0 as u10
1249  node velocity(z)    nearest location 34.0, 21.0, 10.0 as v10
1250  # node acceleration(z)  nearest location 34.0, 21.0, 10.0 as a10
1251  #
1252  # -----
1253  # Bob c.g. from assembly_bob begin
1254  # -----
1255  # Bob-1mm center of gravity
1256  # shifting up by 1 cell so it does not coincide with rb ref node
1257  node coordinates  nearest location 33.6315, 21.4892, 13.3570 as rcg
1258  node displacement  nearest location 33.6315, 21.4892, 13.3570 as ucg
1259  node velocity     nearest location 33.6315, 21.4892, 13.3570 as vcg
1260  # node acceleration nearest location 33.6315, 21.4892, 12.3570 as acg
1261  #
1262  # Bob-2mm center of gravity
1263  # node coordinates  nearest location 33.6329, 21.4885, 12.3519 as rcg
1264  # node displacement nearest location 33.6329, 21.4885, 12.3519 as ucg

```

```

1265      # node velocity      nearest location 33.6329, 21.4885, 12.3519 as vcg
1266      # node acceleration  nearest location 33.6329, 21.4885, 12.3519 as acg
1267      #
1268      # Bob-1mm center of gravity - top of head
1269      node coordinates    nearest location 33.6315, 21.4892, 22.7000 as tcg_r  # node 4178148
1270      node displacement   nearest location 33.6315, 21.4892, 22.7000 as tcg_u  # node 4178148
1271      # -----
1272      # Bob c.g. from assembly_bob end
1273      #
1274      #
1275      #
1276      # Bob bone tracers angular velocity begin
1277      #
1278      # Bob-1mm
1279      # from origin O to point P, superior, posterior, midline skull
1280      node coordinates    nearest location 33.6315, 25.8, 21.3 as rOP
1281      node displacement   nearest location 33.6315, 25.8, 21.3 as uOP
1282      node velocity       nearest location 33.6315, 25.8, 21.3 as vOP
1283      #
1284      # from origin O to point Q, inferior, posterior, midline skull
1285      node coordinates    nearest location 33.6315, 29.5, 10.2 as rOQ
1286      node displacement   nearest location 33.6315, 29.5, 10.2 as uOQ
1287      node velocity       nearest location 33.6315, 29.5, 10.2 as vOQ
1288      #
1289      # from origin O to point R, superior, anterior, midline skull
1290      node coordinates    nearest location 33.6315, 14.6, 19.7 as rOR
1291      node displacement   nearest location 33.6315, 14.6, 19.7 as uOR
1292      node velocity       nearest location 33.6315, 14.6, 19.7 as vOR
1293      #
1294      #
1295      # Bob bone tracers angular velocity end
1296      #
1297
1298      end heartbeat output hscth_file
1299
1300      begin heartbeat output rigid_body_file
1301
1302          stream name = rigid_history.csv
1303          format = SpyHis
1304          at time 0.0 increment = 0.00002    # seconds, 50,000 Hz acquisition
1305          # at time 0.0 increment = 3.00003e-5 # seconds, 33,333 Hz acquisition
1306          # at time 0.0 increment = 0.0001      # seconds, 10,000 Hz acquisition
1307
1308          #
1309          # rigid body
1310          #
1311
1312          global displx_rigidSkull
1313          global disply_rigidSkull
1314          global displz_rigidSkull
1315          global velx_rigidSkull
1316          global vely_rigidSkull
1317          global velz_rigidSkull
1318          global ax_rigidSkull

```

```

1319     global ay_rigidSkull
1320     global az_rigidSkull
1321     global rotdx_rigidSkull
1322     global rotdy_rigidSkull
1323     global rotdz_rigidSkull
1324     global rotvx_rigidSkull
1325     global rotvy_rigidSkull
1326     global rotvz_rigidSkull
1327     global rotax_rigidSkull
1328     global rotay_rigidSkull
1329     global rotaz_rigidSkull
1330
1331 end heartbeat output rigid_body_file
1332
1333 # See SSM User Guide v4.50, Section 9.6.1 Restart Options, page 777/988
1334 # Here is the most basic restart: restart data is written at the
1335 # last step of analysis or if SSM detects an internal error, such as
1336 # element inversion.
1337 #
1338 begin restart data restart_data
1339     database name = g.rsout # the restart file
1340     at wall time 700m increment = 12h # 700 m = 11 hours 40 minutes
1341     restart = auto
1342 end restart data restart_data
1343
1344 end presto region presto_region
1345
1346 end presto procedure presto_procedure
1347
1348 # -----
1349 # end
1350 # ---
1351
1352 end sierra simulation_name

```

submit_script

```

1 #!/bin/bash
2 sierra -T 12:00:00 --queue-name batch --account FY180100 -j 160
3 --job-name bob_066b --run --post adagio -i bob-1mm-5kg-helmet2-0305-hemi-066b.i

```

F.2. Post-Processing

The Python post-processing code base is extensive, and too large to chronicle here, in report form. The code base is contained in the open-source GitHub repository [Hovey, 2020]. In lieu of extensive detail, here we provide only a high-level outline for the post-processing workflow.

Percentiles Extraction on Chama

```
1 [chama-login8] /nscratch/chovey/casco_sim/bob-1mm-5kg-helmet2-0305-hemi-066b]$  
2 $ module purge  
3 $ module load anaconda3  
4 $ module load seacas  
5 $ epu -auto output_field.e.160.000  
6  
7 $ epu -auto output_field.e-s0003.160.000 # first restart, if applicable  
8 $ epu -auto output_field.e-s0002.160.000 # second restart, if applicable  
9  
10 # find 95th threshold for a SSM element variable for all time steps  
11 $ vim gray_white_master.json  
12  
13 # gray_white_master.json  
14 {  
15     "file_exodus": "output_field.e",  
16     "variable": "max_principal_green_lagrange_strain",  
17     "#variable": "max_principal_green_lagrange_strain_rate",  
18     "##variable": "max_principal_log_strain",  
19     "###variable": "max_principal_rate_of_deformation",  
20     "blocks": [7, 8],  
21     "percentile": 95.0,  
22     "time_step_start": 1,  
23     "time_step_stop": 126,  
24     "time_step_increment": 1  
25 }  
26  
27 $ python ../exodus_percentile.py gray_white_master.json  
28 # writes the following four output files  
29 # first variable file  
30 $ Extracted variable written to file: max_principal_green_lagrange_strain.csv  
31 #  
32 # second variable file  
33 $ Extracted variable written to file: max_principal_green_lagrange_strain_rate.csv  
34 #  
35 # third variable file  
36 $ Extracted variable written to file: max_principal_log_strain.csv  
37 #  
38 # fourth variable file  
39 $ Extracted variable written to file: max_principal_rate_of_deformation.csv  
40  
41 # Check these four files into GitLab so they are accessible as raw csv data
```

Percentiles Time Evolution on Chama (or local)

```
1 [local] ~/sibl/io/mil_spec_paper]$  
2 # git pull the above .csv files if on local  
3
```

```

4 # edit ~/sibl/io/mil_spec_paper/h1-master-strain.json
5 $ python ../../xyfigure/client.py h1-master-strain.json
6 # creates h1-master-strain.pdf
7
8 # edit ~/sibl/io/mil_spec_paper/h1-master-strain-rate.json
9 $ python ../../xyfigure/client.py h1-master-strain-rate.json
10 # creates h1-master-strain-rate.pdf

```

Strain versus Strain Rate Cloud Plots with Histograms on Chama

```

1 [chama-login8] /nscratch/chovey/casco_sim/bob-1mm-5kg-helmet2-0305-hemi-066b]$
2 $ module purge
3 $ module load anaconda3
4 $ module load seacas
5
6 # modify
7 # gray_white_master.json
8 # to include the single time step of interest to produce the
9 # strain versus strain rate population of gray and white matter
10 # from max_principal_green_lagrange_strain.csv:
11 # time = 8.4000 ms, time_step = 43, max_GL_strain_rate = 10.6 s^-1
12 # from max_principal_green_lagrange_strain.csv:
13 # time = 13.600 ms, time_step = 69, max_GL_strain = 5.64 percent
14 {
15   "file_exodus": "output_field.e",
16   "variable": "max_principal_green_lagrange_strain",
17   "#variable": "max_principal_green_lagrange_strain_rate",
18   "##variable": "max_principal_log_strain",
19   "###variable": "max_principal_rate_of_deformation",
20   "blocks": [7, 8],
21   "percentile": 95.0,
22   "time_step_start": 1,
23   "time_step_stop": 126,
24   "time_step_increment": 1,
25   "time_step_population": 43,
26   "#time_step_population": 69
27 }
28
29 $ python ./exodus_population.py gray_white_master.json
30 # writes the following output file
31 # max_principal_green_lagrange_strain_ts_43.csv
32 #
33 # successive updates to gray_white_master.json and subsequent
34 # runs of
35 $ python ./exodus_population.py gray_white_master.json
36 # produce the following files, used for histograms:
37 # max_principal_green_lagrange_strain_rate_ts_43.csv
38 # and
39 # max_principal_green_lagrange_strain_ts_69.csv
40 # max_principal_green_lagrange_strain_rate_ts_69.csv
41 #

```

```

42 # These four files are then used as inputs to
43 # ~/sibl/process/exodus/visualization.py
44 # to produce the histograms, via,
45 $ cd ~/sibl/process/exodus
46 $ vim visualization.py # edit as necessary
47 $ conda activate siblenv
48 $ (siblenv) $ python visualization.py

```

F.2.1. Notices

Sandia

Copyright 2020 National Technology & Engineering Solutions of Sandia, LLC (NTESS). Under the terms of Contract DE-NA0003525 with NTESS, there is a non-exclusive license for use of this work by or on behalf of the U.S. Government. Export of this data may require a license from the United States Government.

NOTICE: For five (5) years from 10/22/2020, the United States Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this data to reproduce, prepare derivative works, and perform publicly and display publicly, by or on behalf of the Government. There is provision for the possible extension of the term of this license. Subsequent to that period or any extension granted, the United States Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this data to reproduce, prepare derivative works, distribute copies to the public, perform publicly and display publicly, and to permit others to do so. The specific term of the license can be identified by inquiry made to National Technology and Engineering Solutions of Sandia, LLC or DOE.

NEITHER THE UNITED STATES GOVERNMENT, NOR THE UNITED STATES DEPARTMENT OF ENERGY, NOR NATIONAL TECHNOLOGY AND ENGINEERING SOLUTIONS OF SANDIA, LLC, NOR ANY OF THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT, OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.

Any licensee of "Blunt Impact Brain Injury using Cellular Injury Criterion Code" has the obligation and responsibility to abide by the applicable export control laws, regulations, and general prohibitions relating to the export of technical data. Failure to obtain an export control license or other authority from the Government may result in criminal liability under U.S. laws.

NumPy

From the NumPy website, <https://numpy.org/about/>,

NumPy is an open source project aiming to enable numerical computing with Python. It was created in 2005, building on the early work of the Numerical and Numarray libraries. NumPy will always be 100% open source software, free for all to use and released under the liberal terms of the modified BSD license.

From the NumPy LICENSE.txt file:

```

Copyright (c) 2005-2020, NumPy Developers. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provided
with the distribution.
* Neither the name of the NumPy Developers nor the names of any contributors may be used to
endorse or promote products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

```

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Matplotlib

Matplotlib is Copyright (c) 2012-2013 Matplotlib Development Team; All Rights Reserved. The `license.html` file states, from the matplotlib website, <https://matplotlib.org/>, states:

License Matplotlib only uses BSD compatible code, and its license is based on the PSF license. See the Open Source Initiative licenses page for details on individual licenses. Non-BSD compatible licenses (e.g., LGPL) are acceptable in matplotlib toolkits. For a discussion of the motivations behind the licencing choice, see [Licenses](#).

Copyright Policy John Hunter began matplotlib around 2003. Since shortly before his passing in 2012, Michael Droettboom has been the lead maintainer of matplotlib, but, as has always been the case, matplotlib is the work of many.

Prior to July of 2013, and the 1.3.0 release, the copyright of the source code was held by John Hunter. As of July 2013, and the 1.3.0 release, matplotlib has moved to a shared copyright model.

matplotlib uses a shared copyright model. Each contributor maintains copyright over their contributions to matplotlib. But, it is important to note that these contributions are typically only changes to the repositories. Thus, the matplotlib source code, in its entirety, is not the copyright of any single person or institution. Instead, it is the collective copyright of the entire matplotlib Development Team. If individual contributors want to maintain a record of what changes/contributions they have specific copyright on, they should indicate their copyright in the commit message of the change, when they commit the change to one of the matplotlib repositories.

The Matplotlib Development Team is the set of all contributors to the matplotlib project. A full list can be obtained from the git version control logs.

License agreement for matplotlib 3.3.2 1. This LICENSE AGREEMENT is between the Matplotlib Development Team ("MDT"), and the Individual or Organization ("Licensee") accessing and otherwise using matplotlib software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, MDT hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use matplotlib 3.3.2 alone or in any derivative version, provided, however, that MDT's License Agreement and MDT's notice of copyright, i.e., "Copyright (c) 2012-2013 Matplotlib Development Team; All Rights Reserved" are retained in matplotlib 3.3.2 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates matplotlib 3.3.2 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to matplotlib 3.3.2.

4. MDT is making matplotlib 3.3.2 available to Licensee on an "AS IS" basis. MDT MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, MDT MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF MATPLOTLIB 3.3.2 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. MDT SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF MATPLOTLIB 3.3.2 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING MATPLOTLIB 3.3.2, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between MDT and Licensee. This License Agreement does not grant permission to use MDT trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using matplotlib 3.3.2, Licensee agrees to be bound by the terms and conditions of this License Agreement.

License agreement for matplotlib versions prior to 1.3.0 1. This LICENSE AGREEMENT is between John D. Hunter ("JDH"), and the Individual or Organization ("Licensee") accessing and otherwise using matplotlib software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, JDH hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use matplotlib 3.3.2 alone or in any derivative version, provided, however, that JDH's License Agreement and JDH's notice of copyright, i.e., "Copyright (c) 2002-2009 John D. Hunter; All Rights Reserved" are retained in matplotlib 3.3.2 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates matplotlib 3.3.2 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to matplotlib 3.3.2.

4. JDH is making matplotlib 3.3.2 available to Licensee on an "AS IS" basis. JDH MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, JDH MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF MATPLOTLIB 3.3.2 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. JDH SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF MATPLOTLIB 3.3.2 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING MATPLOTLIB 3.3.2, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between JDH and Licensee. This License Agreement does not grant permission to use JDH trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using matplotlib 3.3.2, Licensee agrees to be bound by the terms and conditions of this License Agreement.

DISTRIBUTION

Email—External

Name	Company Email Address	Company Name
Ashfaq Adnan	aadnan@uta.edu	U of Texas at Arlington US
Amit Bagchi	amit.bagchi@nrl.navy.mil	Naval Research Laboratory US
Timothy Bentley	timothy.b.bentley@navy.mil	Office of Naval Research US
Rika Wright Carlsen	carlsen@rmu.edu	Robert Morris University US
YungChia Chen	yungchia.chen@nrl.navy.mil	Office of Naval Research US
Alice Fawzi	alice.fawzi@brown.edu	Brown University US
Christian Franck	cfranck@wisc.edu	U Wisconsin–Madison US
Diane Hoffman-Kim	diane.hoffman-kim@brown.edu	Brown University US
Wonmo Kang	wonmo.kang@asu.edu	Arizona State University US
Haneesh Kesari	haneesh.kesari@brown.edu	Brown University US
Stylianos Koumlis	koumlis@mines.edu	Colorado School of Mines US
Leslie Lamberson	les@mines.edu	Colorado School of Mines US
Sushant Malave	smalave@teamwendy.com	Team Wendy Inc. US
Luke Summey	summey@wisc.edu	U Wisconsin–Madison US
Ron Szalkowski	rsza-lkowski@teamwendy.com	Team Wendy Inc. US
Patricia VanPoole	patri-cia.m.vanpoole.ctr@navy.mil	Office of Naval Research US
Tony White	Tony.White@trekbikes.com	Trek Bicycle Inc. US

Email—Internal

Name	Org.	Sandia Email Address
John Niederhaus	1443	jhniede@sandia.gov
Michael Burns	5000	mjburns@sandia.gov
Dennis Helmich	5400	drhelmi@sandia.gov
Daniel Kelly	5400	dpkelly@sandia.gov

Name	Org.	Sandia Email Address
Gary Polansky	5400	gfpolan@sandia.gov
Alex Roesler	5420	awroesl@sandia.gov
Scott Miller	1542	stmille@sandia.gov
Kendall Pierson	1542	khpiers@sandia.gov
Kyran Mish	1555	kd mish@sandia.gov
Jason Brown	5421	jbrown2@sandia.gov
Candice Cooper	5421	cfcoope@sandia.gov
Douglas Dederman	5421	dadeder@sandia.gov
William Engleman	5421	wrengle@sandia.gov
Shivonne Haniff	5421	shaniff@sandia.gov
Chad Hovey	5421	chovey@sandia.gov
Carter Sanford	5421	chovey@sandia.gov
Adam Sokolow	5421	acsokol@sandia.gov
Ryan Terpsma	5421	rjterps@sandia.gov
Jonathan Christensen	5422	jbchris@sandia.gov
Jeremy Laflin	5422	jlaflin@sandia.gov
Scott McEntire	5440	rmcenti@sandia.gov
Anirudh Patel	5447	anipate@sandia.gov
Anton Sumali	5447	hsumali@sandia.gov
Technical Library	01177	libref@sandia.gov



**Sandia
National
Laboratories**

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.