ARL-TR-9848 ● DEC 2023

A Wrapper Enabling a VUMAT User Subroutine Originally Developed for Simulating Skull Fracture in Abaqus to be Used in LS-DYNA

by Stephen L Alexander and Tusit Weerasooriya

NOTICES

Disclaimers

ARL-TR-9848 ● DEC 2023

A Wrapper Enabling a VUMAT User Subroutine Originally Developed for Simulating Skull Fracture in Abaqus to be Used in LS-DYNA

Stephen L Alexander
SURVICE Engineering Company

Tusit Weerasooriya
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	3. DATES COVERED END DATE
December 2023	Technical Report	9/01/2023	10/15/2023

4. TITLE AND SUBTITLE
A Wrapper Enabling a VUMAT User Subroutine Originally Developed for Simulating Skull Fracture in Abaqus to be Used in LS-DYNA

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER	5e. TASK NUMBER	5f. WORK UNIT NUMBER

6. AUTHOR(S)
Stephen L Alexander and Tusit Weerasooriya

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-TB Aberdeen Proving Ground, MD 21005	ARL-TR-9848

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID(s): Tusit Weerasooriya, 0000-0003-3299-2166

14. ABSTRACT
This report is a guide for using in LS-DYNA the custom vectorized user material (VUMAT) subroutine we previously developed for simulating skull fracture in Abaqus. Our motivation was that computational simulation of head injury requires knowledge of multiaxial stress thresholds to identify the start of functional impairment, but identifying such thresholds continues to be an ongoing area of research. Therefore, we proposed that skull fracture could serve as an interim indication of injury and developed a microstructurally inspired, mechanism-based Hybrid-Experimental-Modeling-Computational (MIMB-HEMC) concept for finite element simulation of skull fracture. The advantages of a MIMB-HEMC approach were demonstrated by close resemblance between numerical and experimental skull fracture patterns, thus generating substantial interest in the MIMB-HEMC concept within the research community. The concept was originally run with a VUMAT in Abaqus/Explicit. This report provides step-by-step directions for using a wrapper to run the VUMAT in LS-DYNA. It also documents simulations that were run to verify the wrapper implementation. Included is a skull indentation simulation with simplified boundary conditions, showing correspondence between the Abaqus and LS-DYNA computations particularly with respect to failure patterns. Finally, the guide highlights how the present wrapper could be extended for use with other Abaqus VUMATs.

15. SUBJECT TERMS
user-defined material model, head injury, skull fracture, skull failure, UMAT, VUMAT, LS-DYNA, Abaqus, FORTRAN, skull mechanics, MIMB-HEMC, Biological and Biotechnology Sciences, Sciences of Extreme Materials, Terminal Effects

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	41

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Tusit Weerasooriya	(410) 652-9450

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Acknowledgments

Authors gratefully acknowledge the assistance of Carolyn Hampton. She translated the Abaqus input file of the skullcap simulation from our previous publications into an LS-DYNA input file. This input file served as the starting point for the "modified skullcap simulation" that we used in the present document to verify the wrapper.

Authors would also like to acknowledge the input, feedback, and suggestions of Richard Becker. His insights helped us identify the differences between the Abaqus and LS-DYNA codes, which helped to both plan the wrapper and identify and mitigate confounding factors in the skullcap verification simulations.

1. Introduction

1.1 Background

We previously published a microstructurally inspired, mechanism-based Hybrid-Experimental-Modeling-Computational (MIMB-HEMC) concept for finite element (FE) simulation of skull fracture (Weerasooriya and Alexander 2021; Alexander et al. 2021). Here, our motivation for developing the concept was that skull fracture could serve as an interim indicator for head injury while researchers continue to develop cellular and tissue-scale mechanical, electrical, and chemical (MEC) thresholds for the initiation of functional impairments to the brain. The benefits of a MIMB-HEMC approach were demonstrated by simulating a unique experiment published by Gunnarsson et al. (2021) consisting of indenting a so-called “skullcap” specimen. The unique specimen geometry was chosen specifically to induce a multiaxial stress state, which is closer to the in-vivo response than smaller coupon-type specimens such as cubes or 2-D beams. The MIMB-HEMC concept showed remarkable resemblance between numerically predicted and experimental fracture patterns.

The ability of our MIMB-HEMC approach to simulate fracture patterns has drawn the interest of other researchers within and outside of the US Army Combat Capabilities Development Command Army Research Laboratory to implement our concept in their own simulations. However, many researchers use LS-DYNA FE software, even though we originally developed our concept as a custom vectorized user material (VUMAT) in Abaqus, specifically in Abaqus/Explicit. Both LS-DYNA and Abaqus have the capability to use custom user materials written as FORTRAN subroutines. To distinguish between these two interfaces, the framework for custom user materials in Abaqus/Explicit will be referred to as the “Abaqus VUMAT” and the analogous framework in LS-DYNA as the “LS-DYNA user-defined material.” In both, the FE program supplies the current strain increment to the custom subroutine, together with information relating to the current state of the simulation, such as the values of state variables. The primary task of the subroutine is to calculate the corresponding stress, as well as update the state variables. These quantities are passed back to the FE program, which then continues with its calculations.

However, the interfaces for custom user materials are sufficiently different between the two FE programs such that the concept we developed in Abaqus/Explicit could not be directly transferred to LS-DYNA and used as-is. Therefore, we developed a tool to enable researchers to use our Abaqus VUMAT for skull fracture in LS-

DYNA without needing to further modify the Abaqus VUMAT. The tool, a "wrapper," is a LS-DYNA user-defined material subroutine that translates between the LS-DYNA and Abaqus/Explicit conventions while using the Abaqus VUMAT for actual calculations.

The reader should be aware that the interface used by Abaqus/Explicit is also different from the interface used by Abaqus/Implicit. In Abaqus/Implicit, custom user material models are referred to as UMATs. The present wrapper complements a previous wrapper we had published that enabled an Abaqus UMAT containing a non-isotropic material (deformation and failure) model for simulating ultra-high-molecular-weight polyethylene to be used in LS-DYNA (Alexander et al. 2023).

1.2 Purpose and Scope

This report is a user guide for using the wrapper to run in LS-DYNA the skull fracture concept we previously developed for simulating skull failure in Abaqus/Explicit. The reader is referred to the original two reports for complete details of the MIMB-HEMC fracture concept (Weerasooriya and Alexander 2021; Alexander et al. 2021). In summary, a key component of the concept was that the FE mesh of the skull was mapped to the micro-CT dataset of the skull. For each element, the bone volume fraction (BVF or f_{BV}) from the micro-CT data was determined for the physical volume of the skull that the element represented. The element's f_{BV} value was then used to calculate the Young's modulus (E), mass density at the start of the simulation (ρ_{t0}), and thresholds for failure initiation by either compression (σ_f^c), tension (σ_f^t), and/or shear (σ_f^s) as

$$E = 3.0 \cdot (f_{BV})^{1.603} \text{ GPa} \tag{1}$$

$$\rho_{t0} = 1.8 \cdot f_{BV} \text{ g/cm}^3 \tag{2}$$

$$\sigma_f^c = 175.0 \cdot (f_{BV})^{2.0} \text{ MPa} \tag{3}$$

$$\sigma_f^t = 269.4 \cdot (f_{BV})^{2.0} \text{ MPa} \tag{4}$$

$$\sigma_f^s = 150.0 \cdot (f_{BV})^{2.0} \text{ MPa} \tag{5}$$

The concept was implemented in two different VUMAT versions: vA (Weerasooriya and Alexander 2021) and vB (Alexander et al. 2021). The wrapper was specifically developed for use with the latest update, which was published as VUMAT vB.

One distinguishing characteristic of vB was the inclusion of an algorithm to prevent the large, unrealistic force drops that were inherent in vA and were caused by cascades of elements failing in tension. Another distinguishing feature of vB was the automatic removal of elements that exceeded a strain rate threshold, compared to laborious manual removal in vA. The purpose of this feature was to avoid Abaqus terminating the simulation prematurely due to elements becoming too distorted by removing these elements before they became problematic. The VUMAT alerted Abaqus to delete elements by setting the fourth state-dependent variable to a value of zero. VUMAT vB was also distinguished from earlier versions by conforming to the newer Abaqus VUMAT interface used in the Abaqus 2020 release.

However, apart from these distinctions, the two VUMAT versions were largely the same. Therefore, this user guide often refers to "original documentation," meaning that the reader can find the details in either of the previous publications (Weerasooriya and Alexander 2021; Alexander et al. 2021).

The present wrapper is also useful to the wider research community as a foundation from which a more generalized wrapper could be developed for using in LS-DYNA any VUMAT that had originally been developed for Abaqus, apart from our skull fracture concept. Researchers who may have interest in this more generalized use of the present wrapper should be aware that the wrapper was successful for the specific use case of VUMAT vB where the material model of the VUMAT was isotropic and does not have any directionality to deformation or failure. Furthermore, the wrapper only supports a limited number of the arguments that Abaqus would normally provide as input to the VUMAT. Appendix A lists the arguments that the wrapper supports.

1.3 Outline

This user guide starts by providing an overview of the operation of the wrapper. Next, Section 4 supplies step-by-step directions on how to implement the wrapper in LS-DYNA. This section focuses on the use case of the VUMAT vB but also provides generalized instructions for using the wrapper for other VUMATs. Section 5 presents some of the simulations that were run to verify the wrapper by comparing between Abaqus and LS-DYNA outputs with VUMAT vB. Next, the report presents notes related to using the wrapper for other Abaqus VUMATs. Conclusions are drawn in Section 7.

2. Conventions

The start of the simulation is at time t = 0, also identified as t0 (0-th). The FE program progresses through the simulation by advancing through small time increments, Δt. Each time increment ($\Delta t_i = t_i - t_{i-1}$,) is defined as

$$\Delta t = t_2 - t_1, \tag{6}$$

where t2 is the time at the end of the current increment and t1 is the time at the beginning of the current increment (corresponding to the end of the previous increment).

3. Overview

The wrapper is a FORTRAN subroutine called `UMAT43V` and acts as a LS-DYNA user-defined material subroutine (overall concept is summarized in Fig. 1). For each time step, LS-DYNA supplies variables to the wrapper at the beginning of the current time step, t1, including the stress and strain increments, state variables, and material properties. The first part of the wrapper converts these variables from the LS-DYNA convention to the Abaqus convention.

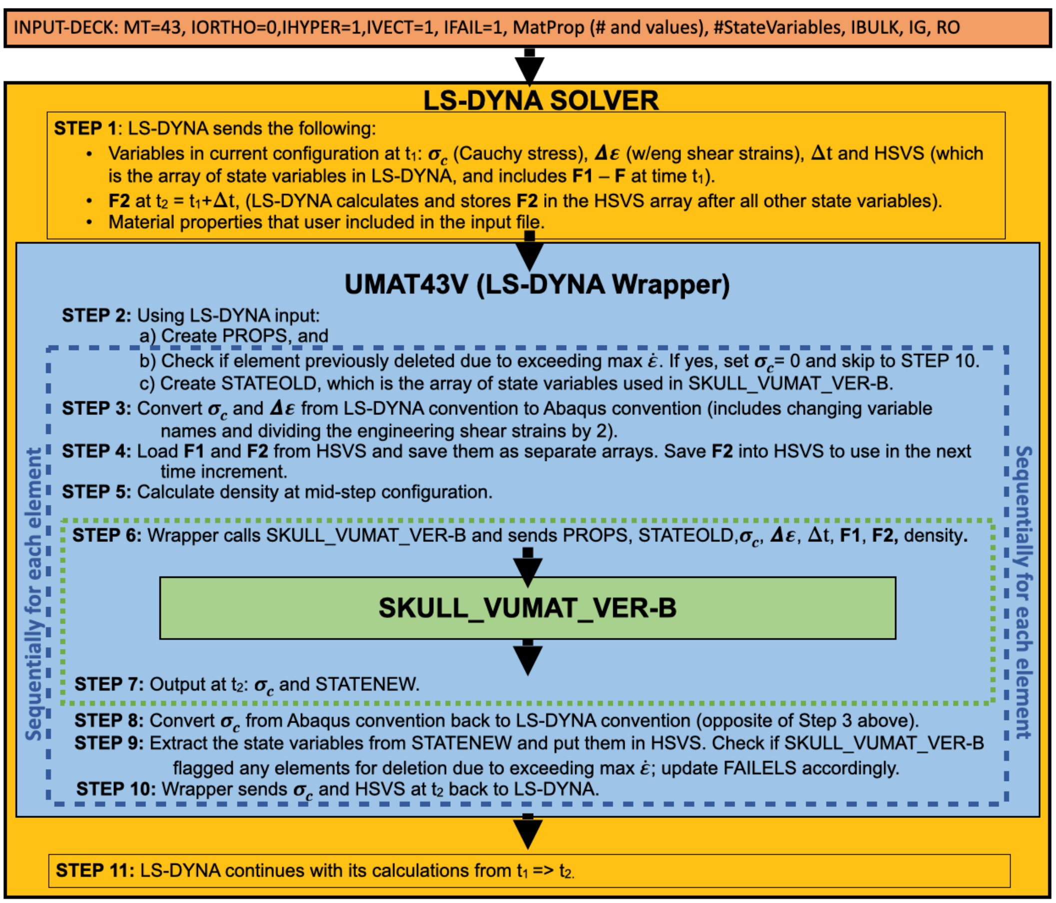


Fig. 1 Summary of wrapper concept. Only the key inputs and outputs at the various levels are included here to help with visualization.

The wrapper checks whether the element had previously exceeded the maximum strain rate threshold. If so, the wrapper automatically sets the stress to 0, without calling the Abaqus VUMAT. For all other elements, the wrapper calculates the density at the mid-step configuration ($\Delta t/2$). The wrapper then calls the Abaqus VUMAT, referred to here as `SKULL_VUMAT_VER-B`. The Abaqus VUMAT calculates the updated values of the stress and history variables for the end of the current time step, t2, as it would when running normally in Abaqus/Explicit. The wrapper then converts the stress and history variables back to the LS-DYNA convention. It also checks whether the Abaqus VUMAT had flagged any elements for deletion due to exceeding the strain rate threshold during the current time increment. For any element flagged for deletion, the wrapper updates the corresponding LS-DYNA variable accordingly.

The wrapper has several write statements that write to terminal output enabling the user to confirm the setup. Appendix B details how state variables are used in the wrapper and in `SKULL_VUMAT_VER-B`.

4. Stepwise Directions to Use Wrapper to Run VUMAT vB in LS-DYNA

4.1 Compile LS-DYNA to Use the Wrapper

The present wrapper was developed with a LS-DYNA distribution that used double-precision variables by default. An example procedure for confirming that LS-DYNA is using double-precision variables has been documented in Appendix C of Alexander et al. (2023).

In LS-DYNA, user-defined material subroutines are included in the source file “`dyn21umatv.f`”. LS-DYNA organizes this source file as a series of subroutines. Each subroutine has a title in the format, `UMATxxV`, where `xx` is a number between 41 and 50. In the original source file from LS-DYNA, each subroutine is simply a placeholder or example. The present wrapper was written to take the place of the subroutine titled `UMAT43V`.

The following steps should be executed prior to compiling LS-DYNA in order to use the Abaqus VUMAT by means of the wrapper:

1) Completely remove the default `UMAT43V` subroutine that LS-DYNA provides as a placeholder in the `dyn21umatv.f` source file.

2) The present wrapper with accompanying utility subroutines was saved in a file called “`wrapper_skullVUMAT.f`”. The user needs to add the

wrapper_skullVUMAT.f file to the same directory as the LS-DYNA dyn21umatv.f source file.

3) Add the following statement to the dyn21umatv.f source file:

   include 'wrapper_skullVUMAT.f'

   This statement must be added at a location that is outside the scope of any of the subroutines in the source file. For example, it could be added to the very top of the file. Or it could be added between the end of the UMAT42V subroutine and the start of the UMAT44V subroutine, taking the place of the UMAT43V subroutine that was deleted in Step 1.

4) The wrapper assumes that the SKULL_VUMAT_VER-B subroutine and its accompanying custom subroutines are saved in a file called "SKULL_VUMAT_VER-B.f". The user needs to add the SKULL_VUMAT_VER-B.f file to the same directory as the LS-DYNA dyn21umatv.f source file.

4.2 Set Up LS-DYNA Input (Keyword) File

The LS-DYNA simulation is set up through an input file, also referred to as the keyword file (k-file). After compilation as described in the previous section, elements can be assigned the SKULL_VUMAT_VER-B material model through the *MAT_USER_DEFINED_MATERIAL_MODELS keyword, abbreviated here as *MAT_USER. This section describes how to specify this keyword in the k-file for the specific use case of the SKULL_VUMAT_VER-B.

The keyword will have three data cards (lines). The first two data cards contain different variables related to the settings for the material model. The third data card contains the material properties sent to LS-DYNA. These material properties include the material properties used by the Abaqus VUMAT, referred to as PROPS, together with additional properties needed by LS-DYNA and the wrapper.

The user should start by determining the values of the PROPS since a unique *MAT_USER keyword will be needed for each element, or set of elements, having different values of PROPS. After determining the values of the PROPS, it is easier to set up the *MAT_USER keyword by listing the LS-DYNA material properties in the third data card prior to specifying the settings in the first two data cards. Therefore, this user guide starts by discussing the values of the PROPS (Section 4.2.1) and then describes the third data card (Section 4.2.2) before describing the first two data cards (Section 4.2.3).

4.2.1 Determine the Values of the Material Properties (PROPS) Used by the Abaqus VUMAT

The user must create a unique *MAT_USER keyword for each element, or set of elements, having different values of PROPS.

The SKULL_VUMAT_VER-B material model used three material properties:

- PROPS(1): the bone volume fraction (f_{BV}) of the element at the start of the simulation. In the original documentation, f_{BV}, was calculated for each element by mapping the element to the location within the micro-CT data that the volume represented.
- PROPS(2): the Poisson's ratio (ν). The original documentation used $\nu = 0.3$ for all elements.
- PROPS(3): the mass density at t0 (ρ_{t0}). In the original documentation, ρ_{t0} was calculated by Eq. 2.

4.2.2 Specify LS-DYNA Material Properties in the Third Data Card

The entries in the third data card are the material properties to be read by LS-DYNA. These properties consist of the material properties, PROPS, used by the Abaqus VUMAT together with four additional properties required by LS-DYNA and the wrapper. The number of material properties used by the Abaqus VUMAT is referred to as NPROPS. The total number of material properties specified in the third data card is referred to here as NPROPS_DYNA, where NPROPS_DYNA=NPROPS+4. For the SKULL_VUMAT_VER-B model, NPROPS=3 and NPROPS_DYNA = 7.

The entries in the third data card are specified as follows:

- First and second entries: estimates for the bulk and shear moduli for the custom user material model. LS-DYNA requires that estimates for the bulk, K, and shear, G, moduli of the custom user material be specified as material properties in the *MAT_USER keyword: "All user-defined material models require bulk and shear moduli for transmitting boundaries, contact interfaces, rigid body constraints, and time-step calculations" (ANSYS 2023a, Appendix A).

  The values for K and G can be calculated from f_{BV} by the following method. First, the Young's modulus is calculated by Eq. 1. Next, K and G are given by

$$K = \frac{E}{3(1-2\nu)}\ GPa,\ G = \frac{E}{2(1+\nu)}\ GPa \tag{7}$$

The present wrapper assumes that K and G are listed as the first two material properties. The user can choose which comes first and second.

- Third entry = 4, the number of state variables used in `SKULL_VUMAT_VER-B`. This number is an integer variable called `NSTATEV`.
- Fourth entry = 3, the number of material properties used in `SKULL_VUMAT_VER-B`. This number is an integer variable called `NPROPS`.
- Fifth and subsequent entries: these are the material properties that are used by the Abaqus VUMAT, as discussed in Section 4.2.1. The wrapper collects these properties into an array, `PROPS`, and sends them to the Abaqus VUMAT. For the `SKULL_VUMAT_VER-B`, the entries are as follows:
    - Fifth entry: f_{BV}. Wrappers stores as `PROPS(1)`.
    - Sixth entry: ν. Wrapper stores as `PROPS(2)`.
    - Seventh entry: ρ_{t0}. Wrapper stores as `PROPS(3)`.

4.2.3 Specify Variables in the First Two Cards of the Keyword

The first two data cards (lines) of the `*MAT_USER` keyword contain different variables related to LS-DYNA settings for proper execution of the material model. The specific values required for each of these variables to use the wrapper for `SKULL_VUMAT_VER-B` are described here. The user is referred to the *LS-DYNA Keyword User's Manual, Volume II* (ANSYS 2023b) for additional information regarding each variable and its specific meaning.

- `MID`: a material identifier. The user must choose a value that is different from any other material used in the simulation.
- `RO`: must be set to the mass density of the material model in consistent units. For the case of `SKULL_VUMAT_VER-B`, this density is ρ_{t0}, which is calculated by Eq. 2 and is also listed as the seventh material property sent to LS-DYNA (Section 4.2.2).
- `MT`: must be set to 43 to match the title of the wrapper subroutine, which was `UMAT43V`.
- `LMC` and `LMCA`: must be specified based on `NPROPS_DYNA`. The general procedure for determining `LMC` and `LMCA` has been described in Appendix D of Alexander et al. (2023). For the case of `SKULL_VUMAT_VER-B`, these values are `LMC`=7 and `LMCA`=0.

- NHV: refers to the Number of History Variables to be allocated by LS-DYNA. Appendix B provides additional information regarding the use of history (or state) variables by the wrapper and by SKULL_VUMAT_VER-B. In summary, NHV must be set to NSTATEV + 9. NSTATEV is the number of state variables used in the Abaqus VUMAT, which was listed as the third material property sent to LS-DYNA (see Section 4.2.2). For the case of SKULL_VUMAT_VER-B, NSTATEV=4 and therefore NHV should be 13.

- IORTHO: this flag controls whether LS-DYNA sends the variables to the wrapper in the current global coordinate system (if IORTHO=0) or a local coordinate system (IORTHO=1). For the case of SKULL_VUMAT_VER-B, we have been setting IORTHO=0 for convenience. Setting IORTHO=1 requires additional data cards to define the local coordinate system when specifying the *MAT_USER keyword. The wrapper should work for either setting because SKULL_VUMAT_VER-B is an isotropic material model, but we have not tested or verified the wrapper for IORTHO=1.

- IBULK and IG: these indicate the location of the estimates for the bulk and shear modulus, K and G (Eq. 7), within the listing of material properties as described previously. The present wrapper assumes that K and G are listed in the first two slots for material properties. The user can choose which comes first and second.

- IVECT: must be set to 1. The present wrapper is written as a vectorized user-defined material subroutine rather than as a scalar subroutine. The IVECT=1 flag indicates this to LS-DYNA.

- IFAIL: must be set to 1. This setting prompts LS-DYNA to send an array of logical values, called FAILELS, to the wrapper. LS-DYNA then erodes any elements for which the wrapper sets the corresponding position in FAILELS to TRUE. SKULL_VUMAT_VER-B uses the fourth state dependent variable (SDV4) as a flag to indicate that an element has exceeded the strain-rate threshold and should be removed. If an element has exceeded the threshold, the VUMAT sets the value of SDV4 for that element to 0. The present wrapper checks SDV4 after calling SKULL_VUMAT_VER-B. The wrapper sets the corresponding position in FAILELS to TRUE if SDV4 is 0.

- ITHERM: determines if LS-DYNA sends temperature information to the user material model. We have been using ITHERM=0. The wrapper does not explicitly handle any temperature variables.

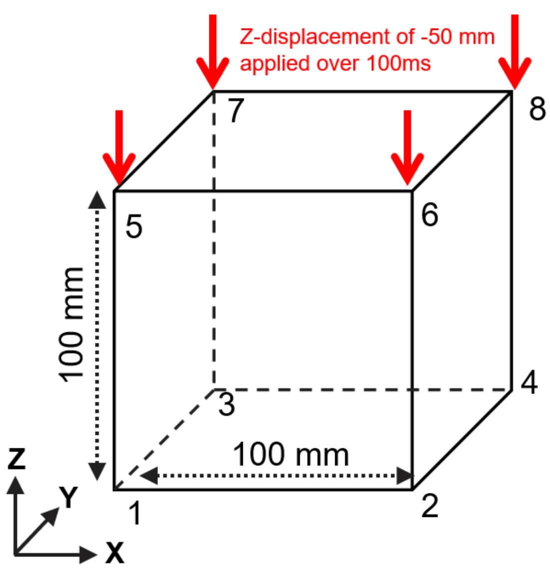
- `IHYPER`: must be set to 1 in order for LS-DYNA to pass the deformation gradient at the end of the current timestep. Appendix B provides additional information.
- `IEOS`: is for equation of state. We have been using `IEOS`=0.

5. Verification of Wrapper by Comparing Abaqus and LS-DYNA Simulations

The implementation of the wrapper was verified by running simulations in LS-DYNA using the `SKULL_VUMAT_VER-B` material model by means of the wrapper. Analogous simulations were run in Abaqus/Explicit directly using the `SKULL_VUMAT_VER-B` material model. Numerical outputs from LS-DYNA and Abaqus were compared.

5.1 Single Element Simulations without Failure

Figure 2 describes the boundary conditions for single element simulations. The element was assigned properties based on the simulations in the original documentation, assuming pure bone: $f_{BV} = 1.0$, $\nu = 0.3$, and $\rho_{t=0} = 1.8 \times 10^{-6}$ kg/mm^3. The failure thresholds (originally given by Eqs. 3–5) were artificially increased for these single element simulations to ensure that no failure occurred during simulation.



Node	Allowed Motion	Displacement Constraints X	Y	Z
1	None	X	X	X
2	X only		X	X
3	Y only	X		X
4	X and Y			X
5	Z only	X	X	
6	Z and X		X	
7	Z and Y	X		
8	X,Y, and Z	[none]		

Fig. 2 Compression boundary conditions used for single element simulations in LS-DYNA and Abaqus/explicit to verify wrapper

Figure 3 compares the stress in the z-direction between LS-DYNA and Abaqus. At the end of the simulation, the value computed in LS-DYNA differed from that in Abaqus by 0.011%.

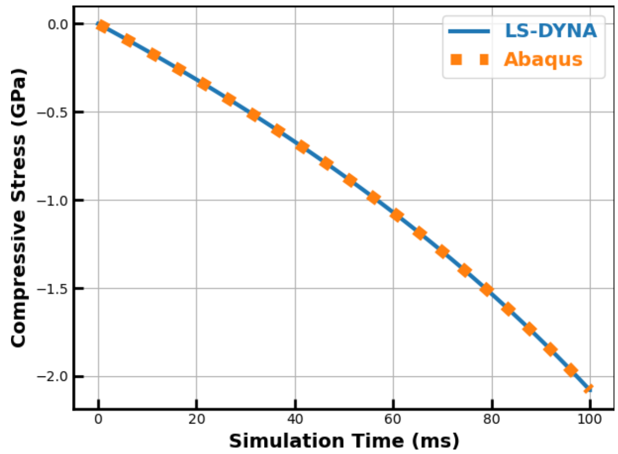


Fig. 3 Compressive stress in the single element verification simulations

In addition, the maximum and minimum principal stresses were compared between the two FE programs at the end of the simulation. It was important to verify principal stresses because the material model in `SKULL_VUMAT_VER-B` checked for element failure by comparing the principal stresses with the failure thresholds (Eqs. 3–5). At the end of the simulation, the minimum principal stress computed by Abaqus (–2.07942247 GPa) and LS-DYNA (–2.07919121 GPa) differed by 0.000231 GPa (0.01%). The maximum principal stresses computed by Abaqus and LS-DYNA were 0.00003 and 0.00004 GPa, respectively. Since the cube is in uniaxial compression, the dominant stress for this loading configuration was the minimum principal stress, and the maximum principal stress should be zero. The nonzero values could be due to stress wave interaction as well as numerical noise. The difference in maximum principal stresses between the two programs was 0.000011 GPa, corresponding to 0.0005% of the minimum principal stress in Abaqus, which is negligible.

We also compared the calculation of relative volume, V_{rel}, between the two FE programs. This quantity was important because the `SKULL_VUMAT_VER-B` used V_{rel} in determining the stress of elements that had previously failed by exceeding one of the failure thresholds. The `SKULL_VUMAT_VER-B` calculated V_{rel} as

$$V_{rel} = \frac{\rho_{t0}}{\rho_{t=t1+\Delta t/2}} \quad (8)$$

In Eq. 8, ρ_{t0} was the mass density at the start of the simulation, which was one of the material properties listed in the `*MAT_USER` keyword (Section 4.2.1). The denominator, $\rho_{t=t1+\Delta t/2}$, was supplied by Abaqus via the `density` variable when using `SKULL_VUMAT_VER-B` in the Abaqus software. This was the density at the midstep configuration, halfway between t1 and t2.

However, LS-DYNA does not supply the midstep density to its subroutines used for custom material models as far as we know. Therefore, the wrapper needed to calculate $\rho_{t=t1+\Delta t/2}$ from other quantities that LS-DYNA supplies in order to use `SKULL_VUMAT_VER-B` in LS-DYNA. First, the wrapper calculates V_{rel} from the deformation gradient at t1, `F1`, and the deformation gradient at t2, `F2`, as

$$V_{rel} = \frac{\det(F1)+\det(F2)}{2} \tag{9}$$

The wrapper calculates $\rho_{t=t1+\Delta t/2}$ by equating Eqs. 8 and 9 to give

$$\rho_{t=t1+\Delta t/2} = \frac{2\rho_{t0}}{\det(F1)+\det(F2)} \tag{10}$$

To verify the calculation of relative volume with the wrapper, we first confirmed that Eqs. 8 and 9 gave the same value for V_{rel} up to eight significant digits when running the single element simulation in Abaqus. Next, running the analogous simulation in LS-DYNA, V_{rel} calculated by Eq. 9 at the end of the simulation deviated by less than 0.0005% from the value of V_{rel} calculated in the Abaqus simulation.

5.2 Skullcap Simulation Using Simplified Boundary Conditions

The wrapper was further verified by rerunning the skullcap simulation from the original documentation. In the original Abaqus simulations, the skullcap rested on an aluminum backing plate. Contact was specified between the skullcap and the backing plate. We tried to run the analogous in LS-DYNA with the wrapper. However, these preliminary simulations of indenting the skullcap resting on the backing plate substantially deviated from the Abaqus response. Most noticeably, the initial linear portion of the load-displacement response was softer than the Abaqus response. This was likely due to differences in how the two FE programs handled contact between the skull and the plate.

Therefore, the plate was removed for the present verification simulations in order to facilitate direct comparison between Abaqus and LS-DYNA. Instead of the backing plate, the modified skullcap simulations (MSS) constrained all nodes within 0.3 mm of the bottom of the skullcap from motion in all three directions.

Figures 4 and 5 compare the predicted response of the MSS between the two programs, showing close agreement. We believe that discrepancies are due to differences in the finite element solvers that are external to the wrapper, rather than deficiencies in the wrapper itself. For example, although the issue of contact between the nodes and the plate was avoided by removing the plate, the MSS still had contact defined between the rigid indenter and the top of the skullcap. There are likely differences in the way that Abaqus and LS-DYNA handle this contact. In addition, the algorithm in `SKULL_VUMAT_VER-B` for computing the stress of high-BVF elements that had failed in tension is dependent on the size of Δt (this algorithm was previously documented in Section 3.3 of Alexander et al. [2021]). To minimize discrepancies between Abaqus and LS-DYNA, the timestep as a function of time was output from the Abaqus MSS and was used as a cap on the timestep in the MSS in LS-DYNA. However, the timestep in the LS-DYNA simulation still deviated from that of Abaqus simulation. Appendix C provides further information on the timestep. The appendix also includes an additional LS-DYNA simulation in which the timestep was not constrained to be less than or equal to the Abaqus timestep. This additional simulation illustrates the effect of timestep on the response.

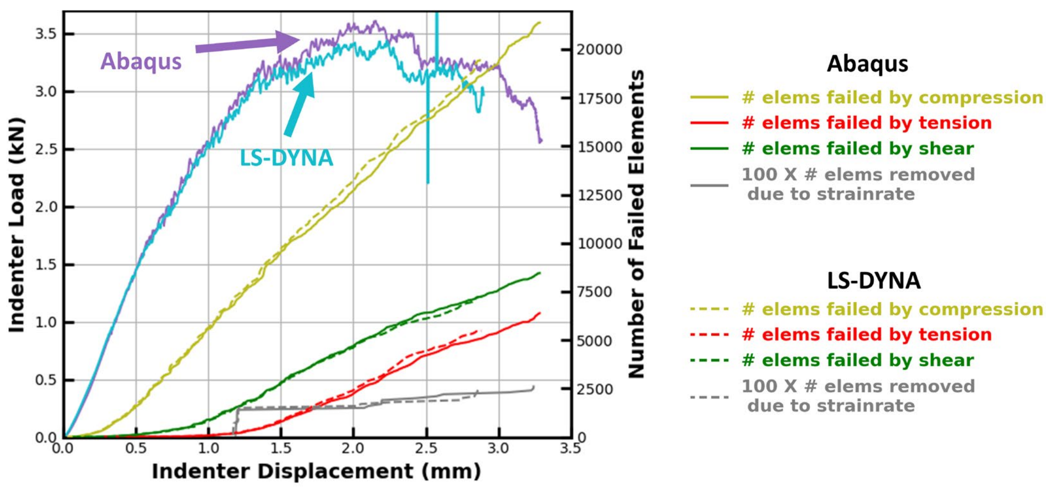


Fig. 4 Comparison of modified skullcap simulation using `SKULL_VUMAT_VER-B` in Abaqus (directly) and in LS-DYNA (via wrapper). Load-displacement is plotted (purple from Abaqus and cyan from LS-DYNA) together with the number of elements that failed by each mechanism and the number of elements that were deleted due to exceeding the strain-rate threshold.

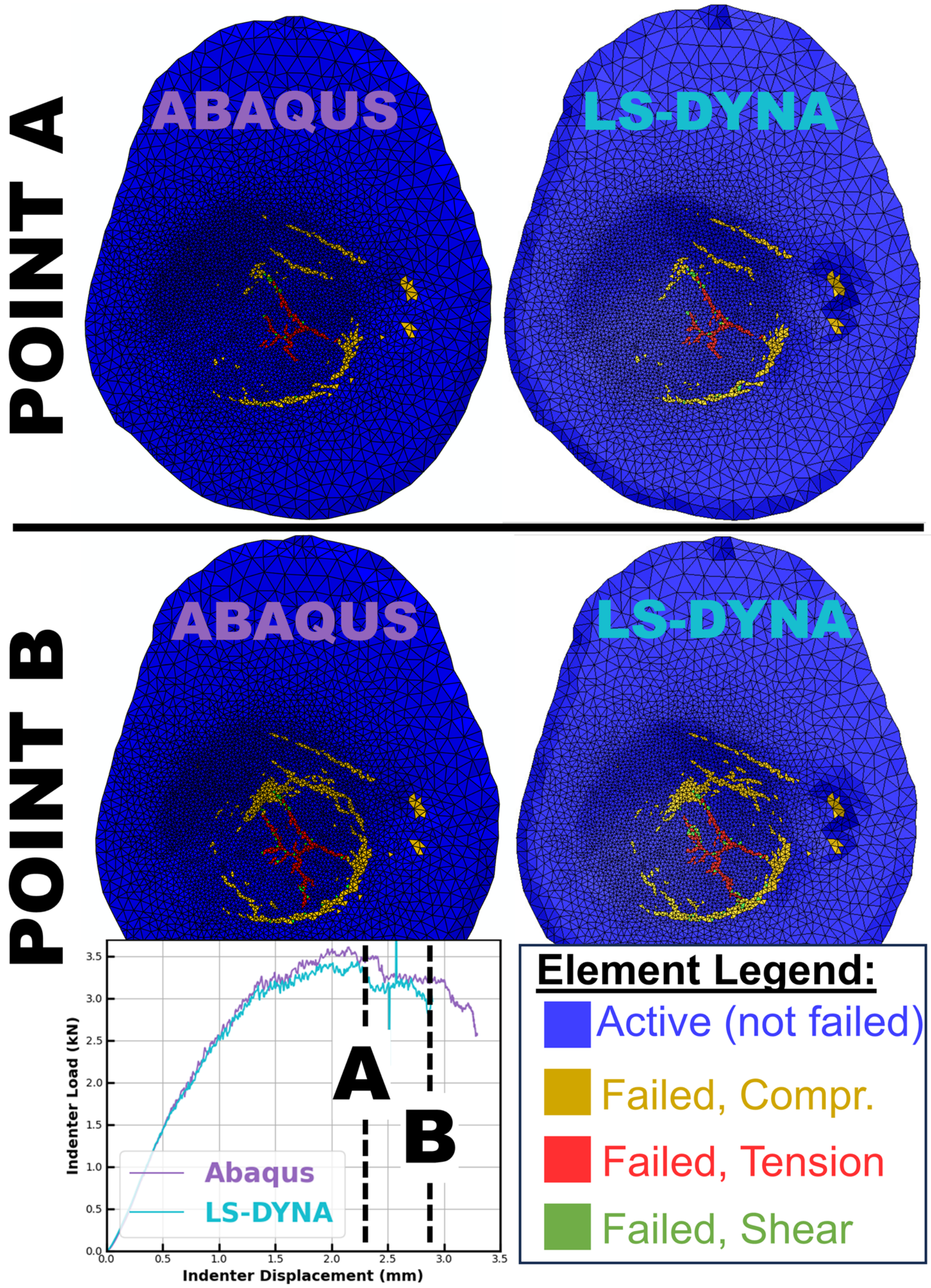


Fig. 5 Comparison of failure patterns from the modified skullcap simulation using `SKULL_VUMAT_VER-B` in Abaqus (directly) and in LS-DYNA (via wrapper). The back surface of the skull mesh is shown with elements colored according to the element legend. Failure patterns are shown at two timepoints, indicated by A and B in the load-displacement curve.

6. Limitations of Using Wrapper for Other Abaqus VUMATs

The wrapper assumes the material model is isothermal and is written for 3-D simulations. Shells and VUMATs coded for 2-D would require modification.

In addition, the present wrapper is limited by only assigning values to the specific arguments in the Abaqus VUMAT header that were needed to run the `SKULL_VUMAT_VER-B`. Moreover, the wrapper assumes that the Abaqus VUMAT header follows the convention that was adopted in the 2020 release of Abaqus and has been in use to the present (2023 release). The distinguishing difference was the introduction in the 2020 release of the `jinfoArray` argument. Appendix A lists the arguments from the Abaqus VUMAT header that are supported by the wrapper. To apply the present wrapper for an Abaqus VUMAT other than `SKULL_VUMAT_VER-B`, users should consult Appendix C to ensure that the Abaqus VUMAT does not require information that is not supported by the present wrapper. One important example is that the present wrapper does not support field variables.

7. Conclusions

The wrapper described in this report can be directly used to run LS-DYNA simulations with the VUMAT vB material model for skull that was originally developed by Alexander et al. (2021) for use in Abaqus/Explicit. The verification simulations documented here indicate the functionality of the wrapper, specifically the close matching of back-surface failure patterns. However, the verifications also highlight that the user should, in general, not expect exact correspondence between LS-DYNA and Abaqus when using the wrapper because of confounding differences between the two programs, such as contact algorithms and timesteps. These differences cause the load-displacement responses to be slightly different. The pattern of failed elements, while still showing discrepancies, is less noticeably affected by the differences between the two programs.

The present wrapper was limited in application to the specific case of the isotropic VUMAT vB material model. It can also be applied to other VUMATs originally developed in Abaqus, given the limitations on function arguments and history variables described in this report. Finally, it also serves as a starting point from which a more generalized wrapper could be developed that would allow any VUMAT from Abaqus to be used in LS-DYNA.

8. References

Alexander SL, Baumer T, Fagan B, Weerasooriya T. Hybrid experimental modeling computational (HEMC) skullcap simulation: Elemental to layer simplification and application to microstructural stochasticity. DEVCOM Army Research Laboratory (US); 2021 Sep. Report No.: ARL-TR-9296. https://apps.dtic.mil/sti/pdfs/AD1148408.pdf

Alexander SL, Becker R, Weerasooriya T. User guide to a wrapper that allows a custom user-defined material model (UMAT) originally developed for abaqus/EPIC to be used in LS-DYNA simulations. DEVCOM Army Research Laboratory (US); 2023 Oct. Report No.: ARL-TN-1177.

ANSYS. LS-DYNA keyword user's manual. Ver. R14.0. ANSYS; 2023a July 31. (Vol. I).

ANSYS. LS-DYNA keyword user's manual. Ver. R14.0. ANSYS; 2023b Feb 24. (Material models; Vol. II).

Gunnarsson CA, Alexander SL, Weerasooriya T. Mechanical response and fracture of human skull to blunt indentation loading. DEVCOM Army Research Laboratory (US); 2021 Feb 3. Report No.: ARL-TR-9142. https://apps.dtic.mil/sti/pdfs/AD1122025.pdf.

Weerasooriya T, Alexander S. Mechanism and microstructure based concept to predict skull fracture using a hybrid-experimental-modeling-computational approach. J Mech Behav Biomed Mat. 2021;121:104599. https://doi.org/10.1016/j.jmbbm.2021.104599.

Appendix A. Variables from the Abaqus VUMAT Interface That Are Supported by the Wrapper

The present wrapper assumes that the header of the Abaqus VUMAT is in the form shown by Fig. A-1. This form is based on the VUMAT interface used in Abaqus/Explicit since the 2020 Abaqus release.[1]

```
      subroutine SKULL_VUMAT_VER-B(
C Read only (unmodifiable)variables -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, jInfoArray,
     2  stepTime, totalTime, dtArray, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only (modifiable) variables -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
```

Fig. A-1 Form of the header for the Abaqus VUMAT that is assumed by the present wrapper

As shown in Fig. A-1, a total of 33 arguments to the function are included in the call to the VUMAT. Abaqus would support all of these arguments for a VUMAT running in Abaqus: it would provide the information to the "read only" arguments and use the values calculated within the VUMAT for the "write only" arguments. However, the present wrapper did not support all of these arguments. The wrapper, prior to calling the VUMAT, only assigned values to the subset of arguments that were needed for the computations within `SKULL_VUMAT_VER-B`. Furthermore, after calling the Abaqus VUMAT, the wrapper only passed information from `stressNew` and `stateNew` back to LS-DYNA.

The subset of arguments supported by the wrapper is listed here with additional information. Arguments are arrays, where the dimension of the array is given within the parentheses—for example, "`props(nprops)`"; all other arguments are scalars.

- `nblock`
  - "number of material points to be processed in this call to VUMAT"[1]
  - This argument is hard-coded in the wrapper to a value of 1.
- `ndir`
  - "number of direct components in a symmetric tensor"[1]
  - This argument is hard-coded in the wrapper to a value of 3.
- `nshr`
  - "number of indirect components in a symmetric tensor"[1]

[1] Abaqus. SIMULIA user assistance 2023. Dassault Systèmes; 2023.

- This argument is hard-coded in the wrapper to a value of 3.

- `nstatev`
  - "number of user-defined state variables that are associated with this material type"[1]
  - The value of this argument is specified by the user in the LS-DYNA input deck (see Section 4.2.2). However, the user should be aware of the following. In Abaqus, the `stateOld` and `stateNew` arrays (described below) are dimensioned using `nstatev`. Since the wrapper cannot know `nstatev` prior to compilation, the wrapper dimensions these arrays using an assumed value of 100. The wrapper checks if `nstatev` is greater than 100 and issues an informative warning and stops the program if it is.
  - For the case of `SKULL_VUMAT_VER-B`: the skull VUMAT used four state variables as described in Appendix B. Therefore, `nstatev` should be 4.

- `nprops`
  - "user-specified number of user-defined material properties"[1]
  - The value of this argument is specified by the user in the LS-DYNA input deck (see Section 4.2.2). However, the user should be aware of the following. In Abaqus, the `props` array (described below) is dimensioned using `nprops`. Since the wrapper cannot know `nprops` prior to compilation, the wrapper dimensions this array using an assumed value of 100. The wrapper checks if `nprops` is greater than 100 and issues an informative warning and stops the program if it is.
  - For the case of `SKULL_VUMAT_VER-B`: the skull VUMAT used three material properties (see Section 4.2.1). Therefore, `nprops` should be 3.

- `jInfoArray(*)`
  - When running a VUMAT within the Abaqus finite element program, this array would contain the information listed here. However, the only functionality supported by the current wrapper is the last item in the list in green font; all other items in red font are not supported by the present wrapper.
    - `jInfoArray(1) = lAnneal`, "flag indicating whether the routine is being called during an annealing process" [1]

- `jInfoArray(2) = intPt`, "integration location number" [1]
- `jInfoArray(3) = iLayer`, "layer number" [1]
- `jInfoArray(4) = kspt`, "section point number in the current layer" [1]
- `jInfoArray(5) = iUpdateEffMod`, "flag indicating whether the bulk modulus and shear modulus must be updated" [1]
- `jInfoArray(6) = ptrjElemNum`, "start location in jInfoArray for user-defined element numbers for all elements in nblock"[1]

- `stepTime`
  - "Value of time since the step began".[1] Abaqus simulations can consist of multiple steps. In Abaqus, "steps" refer to a different concept than the time increment (defined as Δt in the main body of this user guide). For example, a compress-decompress simulation could be divided in an Abaqus simulation into one step for the compression followed by a second step for the decompression. However, the present wrapper assumes that there is only one step in the simulation. The wrapper sets `stepTime` equal to the present simulation time.
- `totalTime`
  - This value of time differs from `stepTime` only when the Abaqus simulation consists of multiple steps (see previous note regarding `stepTime`). However, the present wrapper assumes that there is only one step in the simulation. The wrapper sets `totalTime` equal to the present simulation time, which is the same value that the wraps assigns to `stepTime`.
- `dtArray(2*(nblock)+1)`
  - The only functionality of this array that is supported by the present wrapper is the following:
    - `dtArray(1)` = time increment size
- `props(nprops)`
  - The material properties that are used by the Abaqus VUMAT (see Section 4.2.1).
- `density(nblock)`
  - "Current density at the material points in the midstep configuration".[1]

- Present wrapper calculates this value by Eq. 5 as $\rho_{t=t1+\Delta t/2}$.

- `strainInc(nblock, ndir+nshr)`
  - "Strain increment tensor at each material point" [1]
  - Important to note that Abaqus/Explicit supplies the VUMAT with tensorial shear strain components rather than engineering components.
  - Components are in order of 11, 22, 33, 12, 23, 13.
- `defgradOld(nblock, ndir+2*nshr)`
  - Deformation gradient at t1 in global coordinate system.
  - Components are in order of 11, 22, 33, 12, 23, 31, 21, 32, 13.
- `stressOld(nblock, ndir+nshr)`
  - Cauchy stress tensor at t1. Component ordering is same as `strainInc`.
- `stateOld(nblock, nstatev)`
  - Values at t1 of the state variables used within the Abaqus VUMAT
- `defgradNew(nblock, ndir+2*shr)`
  - Deformation gradient at t2 in global coordinate system.
  - Component ordering is same as `defgradOld`.
- `stressNew(nblock, ndir+nshr)`
  - Cauchy stress tensor at t2. Component ordering is same as `strainInc`.
- `stateNew(nblock, nstatev)`
  - Values at t2 of the state variables used within the Abaqus VUMAT.

Appendix B. The Use of State Variables by the Wrapper, Including How Deformation Gradients Are Stored

B.1 General Information

The number of state variables (also referred to as history variables) used within the present wrapper is defined by the following three different quantities in order of increasing value:

- NSTATEV: the number of state variables used within the Abaqus VUMAT.
- NHV: the number of state variables that the user specifies on the *MAT_USER_DEFINED_MATERIAL_MODELS keyword (abbreviated here as *MAT_USER). This should be

$$\text{NHV} = \text{NSTATEV+9} \tag{B-1}$$

- Total number of history variables that LS-DYNA sends to the wrapper, which is NHV+9.

These differences arise from how LS-DYNA and the present wrapper store the deformation gradients. The wrapper requires the deformation gradient at t1, F1, and the deformation gradient at t2, F2. The wrapper needs F1 and F2 to calculate the midstep density (see Eq. 5). The wrapper also requires F1 and F2 in order to send them to the Abaqus VUMAT as defgradOld and defGradNew, respectively.

LS-DYNA supplies F2 when the user sets the IHYPER flag in the *MAT_USER_DEFINED_MATERIAL_MODELS keyword to 1. When IHYPER=1, LS-DYNA saves the nine components of F2 within the array of state variables at indices of NHV+1 to NHV+9. However, as far as we know, LS-DYNA does not supply F1. Therefore, for each time increment, the wrapper stores F2 in the history variables at indices of NSTATEV+1 to NSTATEV+9. In the subsequent time increment, the wrapper assigns F1 to the values stored from the previous increment.

In summary, by setting IHYPER=1 and NHV according to Eq. A-1, the state variable array that LS-DYNA sends to the wrapper has the following form:

- Indices 1 to NSTATEV: state variables that are used by the Abaqus VUMAT.
- Indices NSTATEV+1 to NSTATEV+9: the nine components of F1, which is the deformation gradient from the end of the previous time increment.
- Indices NSTATEV+10 to NTATEV+18: the nine components of F2, which is the deformation gradient at the end of the current time increment.

B.2 State Variables for the Use Case of SKULL_VUMAT_VER-B

Table B-1 lists the state variables used by the wrapper for the case of SKULL_VUMAT_VER-B. The SKULL_VUMAT_VER-B used four state variables, which are explained in Table B-1. Thus, NSTATEV=4, NHV=13, and the total number of history variables was 22.

Table B-1 State variables used by wrapper for SKULL_VUMAT_VER-B

Index (position in array)	Variable name	Description
The first four state variables are specific to SKULL_VUMAT_VER-B. *Refer to original documentation for details.*[1,2]		
1	SDV1	Flag indicating element status: indicating whether element was active or by what mechanism it failed (compression, tension, or shear)
2	SDV2	Value of damage parameter D
3	SDV3	# increments in which D was used to calculate Young's modulus
4	SDV4	Flag indicating whether element should be deleted due to exceeding the strain rate threshold
Remaining state variables are used to hold F1 and F2.		
5	F1(1,1)	Deformation gradient at end of previous time increment
6	F1(2,1)	Deformation gradient at end of previous time increment
7	F1(3,1)	Deformation gradient at end of previous time increment
8	F1(1,2)	Deformation gradient at end of previous time increment
9	F1(2,2)	Deformation gradient at end of previous time increment
10	F1(3,2)	Deformation gradient at end of previous time increment
11	F1(1,3)	Deformation gradient at end of previous time increment
12	F1(2,3)	Deformation gradient at end of previous time increment
13	F1(3,3)	Deformation gradient at end of previous time increment
14	F2(1,1)	Deformation gradient at end of current time increment
15	F2(2,1)	Deformation gradient at end of current time increment
16	F2(3,1)	Deformation gradient at end of current time increment
17	F2(1,2)	Deformation gradient at end of current time increment
18	F2(2,2)	Deformation gradient at end of current time increment
19	F2(3,2)	Deformation gradient at end of current time increment
20	F2(1,3)	Deformation gradient at end of current time increment
21	F2(2,3)	Deformation gradient at end of current time increment
22	F2(3,3)	Deformation gradient at end of current time increment

[1] Alexander SL, Baumer T, Fagan B, Weerasooriya T. Hybrid experimental modeling computational (HEMC) skullcap simulation: elemental to layer simplification and application to microstructural stochasticity. DEVCOM Army Research Laboratory (US); 2021 Sep. Report No.: ARL-TR-9296. https://apps.dtic.mil/sti/pdfs/AD1148408.pdf

[2] Weerasooriya T, Alexander S. Mechanism and microstructure based concept to predict skull fracture using a hybrid-experimental-modeling-computational approach. J Mech Behav Biomed Mat. 2021;121:104599. https://doi.org/10.1016/j.jmbbm.2021.104599

Appendix C. Timestep Discrepancies between LS-DYNA and Abaqus

Section 5.2 presents the modified skullcap simulation (MSS) that was used to verify the wrapper by comparing between LS-DYNA and Abaqus. When running the MSS in LS-DYNA, we specified two controls on the timestep via the `*CONTROL_TIMESTEP` keyword[1]:

1) The Time Step Scale Factor (`TSSFAC`) was set to a value of 0.5.

2) We constrained the LS-DYNA timestep to never exceed the timestep used in the MSS that was run in Abaqus. This control was implemented by outputting the timestep as a function of time from the Abaqus MSS as tabular data and then pointing the `LCTM` flag in LS-DYNA to this table.

Due to constraint on the maximum timestep, the LS-DYNA simulation ran with the above two controls will be referred to as MSS_WITHCAP.

We also ran an additional simulation in LS-DYNA that was identical to the MSS_WITHCAP but without the constraint on the maximum timestep. This additional simulation will be referred to as MSS_NOCAP. The MSS_NOCAP had only the first control listed above, `TSSFAC`=0.5.

Figure C-1 compares the load-displacement response and the number of failed or removed elements. Figure C-2 compares the back-surface contours. The MSS_NOCAP is shown to have had a large load drop at an indenter displacement of approximately 2.3 mm. This load drop corresponded to a spike of elements that failed in tension. The `SKULL_VUMAT_VER-B` featured an algorithm designed to prevent these load drops due to elements failing in tension, but the algorithm was dependent on the timestep size (Section 3.3 of Alexander et al. [2021]). Figure C-2 compares the finite element (FE) solver timestep, showing that the MSS_NOCAP had a significantly higher timestep than the Abaqus simulation in the period immediately before the large load drop at a displacement of approximately 2.3 mm. Therefore, we concluded that the load drop in the MSS_NOCAP simulation was due to the timestep being too high.

For the LS-DYNA simulations that are shown in Fig. C-3, setting `TSSFAC`=0.5 to reduce the timestep likely imposed a timestep of around 0.2 ms in the beginning of the simulation. If the simulations were run without `TSSFAC`=0.5, the timestep before displacement = 0.5 mm would have been larger and somewhat similar to what it was with Abaqus.

[1]ANSYS. LS-DYNA keyword user's manual. Ver. R14.0. ANSYS; 2023a July 31. (Vol. I).

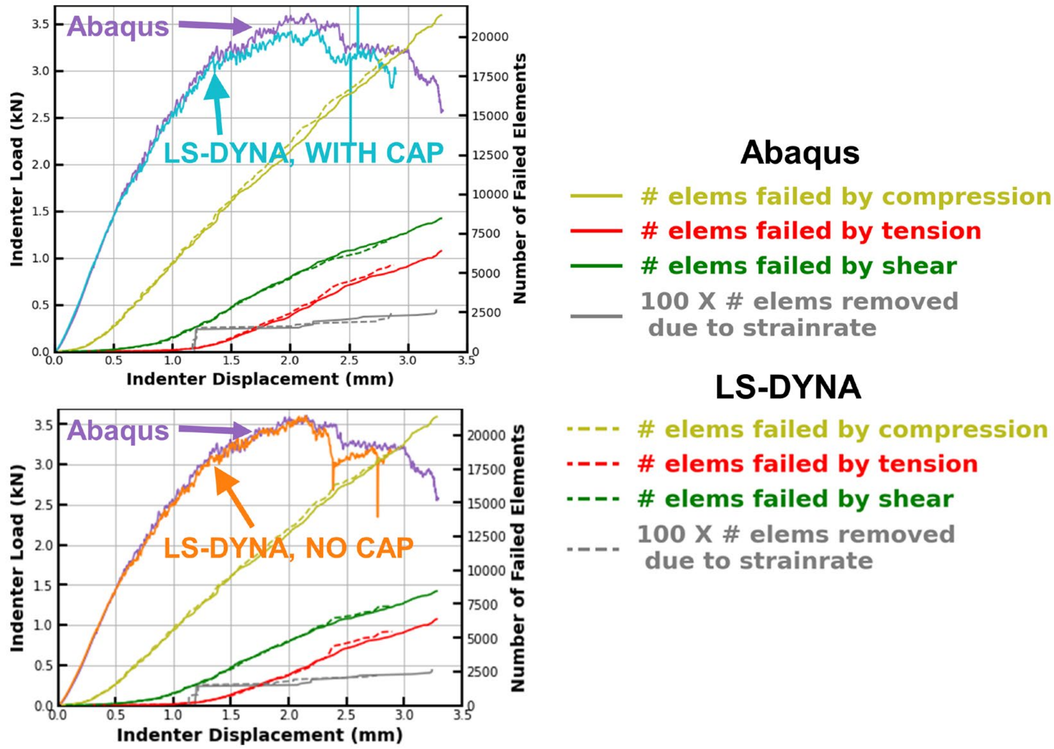


Fig. C-1 Comparison of load-displacement and number of failed and removed elements

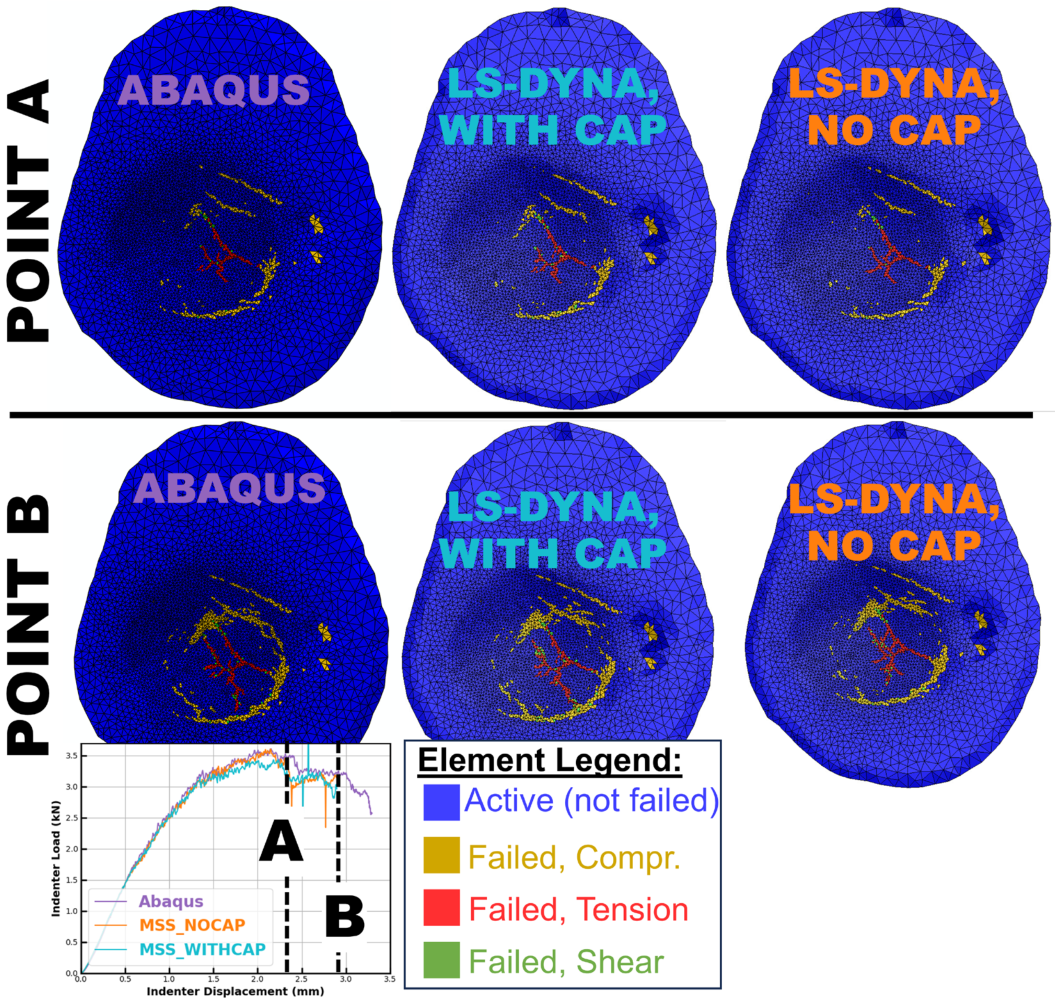


Fig. C-2 Back-surface failure patterns. The skull mesh is shown with elements colored based on status as shown in the element legend. Contours are shown at two different timepoints during the simulation, marked as A and B in the load-displacement plot.

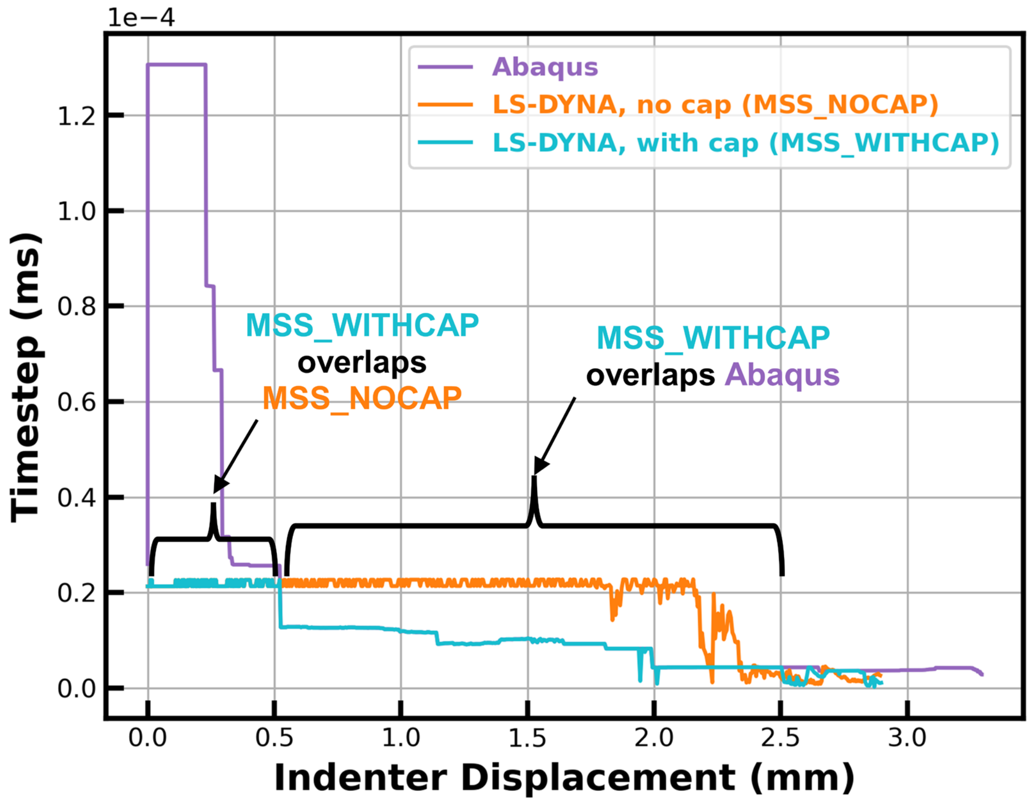


Fig. C-3 Comparison of timesteps in the FE solvers

List of Symbols, Abbreviations, and Acronyms

2-D	two-dimensional
3-D	three-dimensional
BVF	bone volume fraction
CT	computed tomography
FE	finite element
HEMC	hybrid-experimental-modeling-computational
MEC	mechanical, electrical, and chemical
MIMB	microstructurally inspired, mechanism based
MSS	modified skullcap simulation
SDV4	fourth state dependent variable
UMAT	user material
VUMAT	vectorized user material

1 (PDF) DEFENSE TECHNICAL INFORMATION CTR
DTIC OCA

1 (PDF) DEVCOM ARL
FCDD RLD CL
TECH LIB

13 (PDF) DEVCOM SC
M G CARBONI
D COLANTO
R DILLALLA
B FASEL
C HEWITT
J KIREJCZYK
R KOLLAR
D KUBIAK
C LINNINGTON
D OTTERSON
J PARKER
W SHAW
A STRZEPEK

4 (PDF) PEO SOLDIER
A DEGROOT
J HOPPING
P LOOMIS
N NGUYEN

1 (PDF) PEO IEW&S
A FOURNIER

1 (PDF) MTRL SCIENCES DIV
LBNL
R RITCHIE

3 (PDF) SWRI
C ANDERSON JR
S CHOCRON
D NICOLELLA

2 (PDF) NIST
A FORSTER
M VANLANDINGHAM

1 (PDF) IDA
Y MACHERET

5 (PDF) MRDC DOD BIRCO
R GUPTA
A LEWIS
T PIEHLER
R SHOGE
R SPENCER

1 (PDF) USMC
C STEPHENS

4 (PDF) USAARL
F BROZOSKI
V CHANCEY
B MCENTIRE
T ROOKS

1 (PDF) IWTSD
L TANYAG

1 (PDF) DEVCOM GVSC
R SCHERER

1 (PDF) DEVCOM AMSRD PE
D RUSIN

2 (PDF) DEVCOM CBC
M HORSMON
N VINCELLI

1 (PDF) OSD DOT&E
J IVANCIK

5 (PDF) NRL
A BAGCHI
A ILIOPOULOS
J MICHOPOULOS
K TEFERRA
X TAN

3 (PDF) DAC
FCDD DAG S
K LOFTIS
FCDD DAS LBW
G DIETRICH
FCDD DAS LBE
J GURGANUS

2 (PDF) DEVCOM ATLANTIC
S COLEMAN
H PIETSCH

3 (PDF) USSOCOM
M CLARK
D GUILLEMIN
N TSANTINIS

142 (PDF) DEVCOM ARL
FCDD RLA A
S KARNA
P FRANASZCZUK
J NEWILL
A RAWLETT
S SCHOENFELD
FCDD RLA B
A WEST
M TSCHOPP
FCDD RLA CB
R BECKER
J CAMPBELL
P GILLICH
J LASALVIA
A TONGE
FCDD RLA FF
W HAIRSTON
FCDD RLA HC
A DAGRO
A EIDSMORE
T THOMAS
FCDD RLA M
B CHEESEMAN
K CHO
C HOPPEL
FCDD RLA MA
K BERNETICH
T BOGETTI
S BOYD
J CAIN
D KNORR
M NEBLETT
E SANDOZ-ROSADO
J SANDS
J STANISZEWSKI
M YEAGER
FCDD RLA MB
G GAZONAS
D GRAY
D MAGAGNOSC
P MOY
D O'BRIEN
J SIETINS
T WALTER
FCDD RLA MC
D CRAWFORD
R JENSEN
J SNYDER
FCDD RLA MD
A BUJANDA
J LA SCALA
E WETZEL
FCDD RLA ME
P PATEL
J SWAB
L VARGAS-GONZALEZ
FCDD RLA MF
K DARLING
D FIELD
A GIRI
S GRENDAHL
C HAINES
K LIMMER
H MURDOCH
FCDD RLA MG
J LENHART
R MROZEK
J ORLICKI
T SIRK
I YEH
FCDD RLA T
M FERMEN-COKER
R FRANCART
T HOLDREN
R YEAGER
FCDD RLA TA
S BILYK
M GRAHAM
S TURNAGE
W UHLIG
C WILLIAMS
FCDD RLA TB
S ALEXANDER
T BAUMER
D CASEM
J CLAYTON
B FAGAN
A GOERTZ
A GUNNARSSON
C HAMPTON
M KLEINBERGER
D KRAYTERMAN
E MATHEIS
J MCDONALD
P MCKEE
C MEREDITH
T PLAISTED
K RAFAELS
S SATAPATHY
L SHANNAHAN
M TEGTMEYER
C WEAVER
T WEERASOORIYA
S WOZNIAK
T ZHANG
FCDD RLA TD

R DONEY
R GUPTA
M KEELE
D KLEPONIS
B KRZEWINSKI
K MASSER
F MURPHY
C RANDOW
S SCHRAML
K STOFFEL
M ZELLNER
FCDD RLA TE
M BURKINS
D GALLARDY
W GOOCH
E KLIER
J LLOYD
M LOVE
P SWOBODA
FCDD RLA TF
J ANGEL
W BRUCHEY
J CAZAMIAS
R COATES
T EHLERS
P JANNOTTI
E KENNEDY
R LEAVY
J LEE
L MAGNESS
D MALLICK
C MEYER
J RUNYEON
FCDD RLA TG
C CUMMINS
D FOX
N GNIAZDOWSKI
S HUG
S KUKUCK
C PECORA
FCDD RLA V
S SILTON
FCDD RLA VA
R EMERSON
A GHOSHAL
FCDD RLA VB
A HALL
FCDD RLA W
T V SHEPPARD
FCDD RLA WA
J BRENNAN
FCDD RLA WC
M MINNICINO
FCDD RLB
J ZABINSKI
FCDD RLR A
J CIEZAK-JENKINS
D STEPP
FCDD RLR EM
A BROWN
C VARANASI
FCDD RLR ET
B ASHFORD
D COLE
D FORD
FCDD RLR EW
R ANTHENIEN

1 (PDF) SNL
B SANBORN

1 (PDF) NASA LANGLEY RESEARCH CENTER
J CLINE