ARL-TR-8954 • MAY 2020

DEVCOM ARMY RESEARCH LABORATORY

Resolution as a Computational Factor in ALEGRA Simulations of a Shaped-Charge Jet

by Daniel J Hornbaker

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8954 ● MAY 2020

Resolution as a Computational Factor in ALEGRA Simulations of a Shaped-Charge Jet

Daniel J Hornbaker
Weapons and Materials Research Directorate, CCDC Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
May 2020	Technical Report	October 2019–April 2020

4. TITLE AND SUBTITLE
Resolution as a Computational Factor in ALEGRA Simulations of a Shaped-Charge Jet

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Daniel J Hornbaker

5d. PROJECT NUMBER
1L162618AH80

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
CCDC Army Research Laboratory
ATTN: FCDD-RLW-PE
Aberdeen Proving Ground, MD 21005-5069

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-8954

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
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID(s): Daniel J Hornbaker, 0000-0002-0746-1614

14. ABSTRACT
Uncertainty associated with changing resolution in ALEGRA simulations of a 65-mm-diameter shaped-charge-jet device is investigated. The ALEGRA code is found to be well-behaved as resolution decreases, with several measured jet properties exhibiting limiting behavior and convergence of outcomes between 2-D and 3-D simulations. Estimates of the expected mean value and distribution widths for these properties as a function of resolution are provided. Mass loss in the jet due to discretization, resolution dependence of jet temperature, and observations of unphysical jet tip morphology at very fine resolutions are also reported.

15. SUBJECT TERMS
numerical simulation, uncertainty quantification, ALEGRA, shaped-charge jet, resolution study

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Daniel J Hornbaker
Unclassified	Unclassified	Unclassified	UU	52	19b. TELEPHONE NUMBER (Include area code) 410-278-7697

Contents

List of Figures

List of Tables

Acknowledgments

This work was carried out under the Terminal Ballistics for Lethality and Protection Sciences Frontier Project, and was supported in part by a grant of computer time from the Department of Defense High Performance Computing Modernization Program at the US Army Combat Capabilities Development Command Army Research Laboratory Department of Defense Supercomputing Resource Center.

1. Introduction

Numerical simulations are routinely employed by the US defense community for the research and development of protection systems. In the early days of massively parallel computing, the successful execution of a handful of terminal ballistics simulations was an achievement in itself. The rapid growth in computational resources over the subsequent decades now enables arrays of hundreds or even thousands of simulations to be undertaken. This growth makes possible robust statistical investigations into variability and uncertainty in numerical modeling.

1.1 Prior Work

Previous work quantified uncertainty in ALEGRA and CTH simulation results for two different classes of threats, kinetic energy (KE) projectiles[1] and shaped-charge jet (SCJ) devices,[2] with a particular focus on the variability attributable to purely computational factors. Computational factors are parameters that relate solely to the mathematical process of a simulation, not to any real-world aspect of the modeled system. The output quantity of interest (QoI) in these studies was the depth of penetration (DoP) of the threat into rolled homogeneous armor (RHA). This choice of QoI reduced complex model system results to a simple scalar value representing a primary measureable outcome in experimental testing. The main results obtained from these studies are summarized in Table 1. In a few cases, changing a particular computational factor had no effect on a simulation. It was more often the case that varying a factor changed the computation. The effect on the QoI in the KE system was often smaller than the measurement precision. In the more complex SCJ system, the DoP typically varied by as much as a few percent of the mean value.

Table 1 Coefficients of variation for DoP into RHA due to various computational factors in simulations of KE and SCJ model systems at 0.500-mm resolution in CTH and ALEGRA

Factor	KE projectile		SCJ device	
	CTH 11.1	ALEGRA 21 May 2015	CTH 12.0	ALEGRA 2017.11.06
Cores/node	0	0	0	0
Total cores	< 0.1%	0	2.8%	0
Decomposition	< 0.1%	0	3.5%	0
Void buffer	< 0.1%	< 0.1%	4.4%	0.3%
Symmetry	< 0.1%	< 0.1%	3.2%	0.9%
Origin placement	< 0.1%	0.4%	3.9%	1.6%

Table 1 Coefficients of variation for DoP into RHA due to various computational factors in simulations of KE and SCJ model systems at 0.500-mm resolution in CTH and ALEGRA (continued)

Factor	KE projectile		SCJ device	
	CTH 11.1	ALEGRA 21 May 2015	CTH 12.0	ALEGRA 2017.11.06
Stop time	< 0.1%	0.3%	0.3%	0.1%
Max time step	< 0.1%	< 0.1%	2.4%	0.5%
Restarts	0	0	> 0	0
Translation	< 0.1%	< 0.1%	3.6%	0.2%
Axial rotation	< 0.1%	< 0.1%	2.1%	0.1%
Threat direction	2.2%	0.4%	14.9%	12.0%
Reference frame	0.3%	0.4%	7.9%	1.8%

= simulations mathematically invariant to this factor
= DoP variance is less than measurement precision
= measurable variance in DoP

Experienced computational scientists will recognize that one important factor missing from Table 1 is resolution. Resolution is a measure of the size of the discrete elements of a simulation. In common parlance, resolution is said to increase or become finer as the size of the elements become smaller. The prior studies employed 3-D Eulerian meshes with uniform cubic elements. Resolution in this context refers to the width of a single element. More precisely, this is the spatial resolution of the simulation. Dynamic simulations also involve discrete steps in time. The size of these steps, or temporal resolution, is typically a variable quantity that is calculated within a simulation based on stability criteria.

A quick resolution study was conducted at the outset for both systems in the prior work, but the main investigation was conducted at a fixed spatial resolution of 0.500 mm. This choice was driven by the special role of resolution in numerical simulations. Keep in mind that computational models in these studies are discretized approximations of an idealized continuum problem. Discretization induces approximation errors, the magnitude of which are generally related to resolution, with coarser resolution generating larger errors. Consider a sequence of simulations S_i having increasingly finer resolution r_i. If S represents the solution of the idealized continuum problem, then the expectation is that

$$\lim_{r_i \to 0} S_i = S \, . \tag{1}$$

The differences between S_i and S constitute the approximation error ε_i. A consequence of Eq. 1 is that this error becomes vanishingly small as resolution goes to zero,

$$\lim_{r_i \to 0} \varepsilon_i = \lim_{r_i \to 0} (S - S_i) = 0 \,. \tag{2}$$

This behavior can be demonstrated using a simple toy model system.

1.2 Numerical Simulation Toy Model

Consider an object dropped from some initial height h_0 (Fig. 1). Gravity will cause the object to accelerate toward the ground at a constant rate g. Suppose we are interested in calculating the path of the object, which is described by its height as a function of time $h(t)$.

At time $t = 0$ the height is $h(0) = h_0$. At some later time $t = \tau$ the object will hit the ground, $h(\tau) = 0$. For times in between ($0 < t < \tau$) the object will be at some height between the ground and the initial height ($0 < h(t) < h_0$).

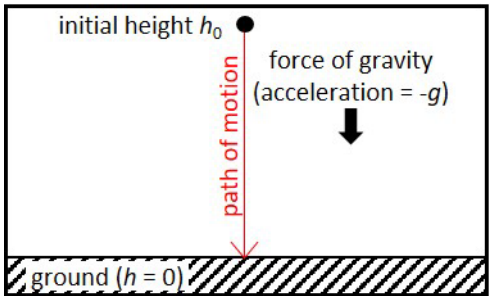

Fig. 1 Diagram of the dropped object problem

This problem is straightforward to solve analytically. The acceleration of the object is the second time derivative of its height,

$$\frac{d^2h(t)}{dt^2} = -g. \tag{3}$$

Integrating twice produces the familiar equation of motion for a body experiencing constant acceleration,

$$h(t) = h_0 + v_0 t - \frac{g}{2} t^2, \tag{4}$$

where v_0 is the initial velocity of the object, which in our example is zero ($v_0 = 0$).

Let us assume for the moment that this exact analytic solution eludes us, so we turn to numerical computation to estimate the motion of the object. We know that the rate of change in height of the object at any given time is that object's velocity v, which can be expressed as the first time derivative of the height,

$$v(t) = \frac{dh(t)}{dt} \equiv \lim_{\delta t \to 0} \frac{h(t+\delta t) - h(t)}{\delta t}, \tag{5}$$

where we have written out a formal definition of the derivative. This basically states that the velocity at any given time is the ratio of the change in height over an infinitesimally small increment of time δt to the size of that increment. Unfortunately, our computer is unable to compute infinitesimally small increments. But it can compute things over small finite increments Δt, so we make an approximation

$$v(t) = \lim_{\delta t \to 0} \frac{h(t+\delta t) - h(t)}{\delta t} \approx \frac{h(t+\Delta t) - h(t)}{\Delta t}, \tag{6}$$

where it is understood that Δt is "small". In this example, the size of our time steps Δt is the (temporal) resolution of our calculation. Rearranging terms gives us a way of estimating the height of our falling object:

$$h(t + \Delta t) \approx v(t)\Delta t + h(t). \tag{7}$$

We know that at time zero $h(0) = h_0$ and $v(0) = 0$, so the height at some small later time Δt is approximately

$$h(\Delta t) \approx v(0)\Delta t + h(0) = h_0. \tag{8}$$

This estimate shows clearly that our computation is an approximation. While the initially motionless object will not move far in a small amount of time, it should move *some*. We can make this explicit by adding a term e_1 for the approximation error to Eq. 8, which changes the approximate equality (represented by the ≈ symbol) to a full equality,

$$h(\Delta t) = v(0)\Delta t + h(0) + e_1 = h_0 + e_1. \tag{9}$$

Because we happen to know the exact solution for this particular example, we can use Eq. 4 to calculate the approximation error:

$$h_0 + e_1 = h(\Delta t) = h_0 - \frac{g}{2}\Delta t^2 \Rightarrow e_1 = -\frac{1}{2} g \Delta t^2. \tag{10}$$

Proceeding to the second time step $t = 2\Delta t$, our computational approximation is

$$h(2\Delta t) \approx v(\Delta t)\Delta t + h(\Delta t). \tag{11}$$

For $h(\Delta t)$ we have Eq. 9, but what of $v(\Delta t)$? Because the acceleration due to gravity is constant, the velocity of the object at any time is simply

$$v(t) = -gt. \tag{12}$$

Substituting into Eq. 11,

$$h(2\Delta t) \approx -g\Delta t^2 + h_0 + e_1. \quad (13)$$

Note that Eq. 13 is still an approximation and requires another term for the approximation error to make it a full equality:

$$h(2\Delta t) = -g\Delta t^2 + h_0 + e_1 + e_2. \quad (14)$$

The new approximation error adds to the approximation error from the previous step. The accumulation of errors over repeated steps is a key aspect of iterative numerical methods. A poorly considered method can result in the total error quickly snowballing. This is a matter of *stability* in numerical simulations.

Rather than consider the added approximation error of each step separately, we will total the accumulated error using

$$E_n = \sum_{s=1}^{n} e_s. \quad (15)$$

Equation 14 is then

$$h(2\Delta t) = -g\Delta t^2 + h_0 + E_2. \quad (16)$$

As before, we can use our knowledge of the exact solution to compute the error,

$$-g\Delta t^2 + h_0 + E_2 = h(2\Delta t) = h_0 - \frac{g}{2}(2\Delta t)^2 \Rightarrow E_2 = -g\Delta t^2. \quad (17)$$

Likewise, for the third step,

$$h(3\Delta t) = -3g\Delta t^2 + h_0 + E_3. \quad (18)$$

$$E_3 = -\frac{3}{2}g\Delta t^2. \quad (19)$$

And so on. In general, at step n, the estimated height and associated total approximation error are

$$h(n\Delta t) = E_n + h_0 - \sum_{s=1}^{n} sg\Delta t^2 = E_n + h_0 - \frac{n(n-1)}{2}g\Delta t^2. \quad (20)$$

$$E_n = -\frac{n}{2}g\Delta t^2. \quad (21)$$

Here we see the special role of resolution in our method. In this simple example, the resolution of our numerical calculation is the size of the time step Δt we use in our estimation of $h(t)$. Equation 21 shows that our total approximation error E_n at any step n is proportional to the square of the resolution. There is a hidden bit of nuance, as the elapsed simulation time t at any step n is $t = n\Delta t$, so the approximation error expressed as a function of time t rather than step n is

$$E(t) = -\frac{t}{2} g \Delta t. \quad (22)$$

The approximation error still remains proportional to the resolution, so we can make our estimate as accurate as we like by selecting a small enough time step Δt. However, the smaller our time step, the more steps we need to compute the motion of the object up to time t. Accuracy comes at a price of computational time.

1.3 Resolution Study Methodology

One implication of the approximation error's dependence on resolution is that the results from the prior studies are tied to the fixed 0.500-mm resolution used therein. If those studies were repeated at increasingly finer resolutions, the observed variances would generally decrease as the discrete approximations more closely resembled the continuum solutions. Another way of framing this is that different choices of computational factor values represent different approximations of the underlying continuum system. The values in Table 1 measure the spread in the resulting QoI across a range of these approximations. As resolution goes to zero, all of these approximations should converge to the continuum solution, so the resulting distribution of the QoI values should collapse onto the continuum result.

Investigating this limiting behavior is the focus of the present study. An SCJ formation model is used in place of the penetration models used previously. This eliminates the need to simulate penetration into an RHA target over long time periods. The model system is detailed in Section 2. SCJ formation is simulated in 2 and 3 spatial dimensions over a range of resolutions using the ALEGRA code.[3] The computational setup and workflow automation for these simulations are discussed in Section 3. Eliminating RHA target penetration from the model system removes DoP as an obvious choice for a simplifying QoI. Instead, element data from forming jets were periodically exported during simulations, allowing jets to be characterized and compared in a variety of ways. The amount of generated data is too voluminous to present in full, but a selection of findings and observations summarizing the model system behavior is presented in Section 4. This report closes with a few concluding remarks in Section 5.

2. Model System

Resolution effects in simulations of an explosively driven laboratory 65-mm-diameter SCJ device are investigated. Unlike typical simulation studies involving SCJ devices,[4–8] accurately replicating real-world jet formation is not of concern here. More important is that the model represents SCJ formations as a class so that

it is reasonable to infer that behaviors observed in this study are applicable to simulations employing SCJ models having different geometries and materials.

2.1 65-mm SCJ Device

The previously employed precision laboratory 65-mm-diameter SCJ device model[2] (Fig. 2) is used for this investigation.

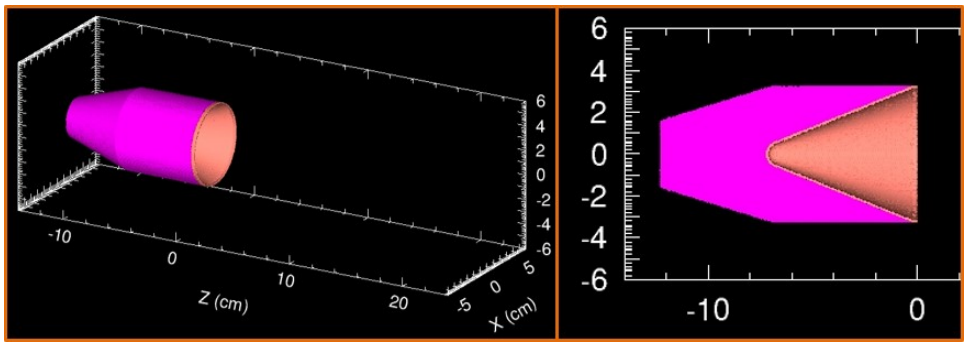

Fig. 2 Perspective (left) and section (right) views of the 65-mm SCJ device model

There are a few changes to the model in this work. The explosive is detonated from the center point of the back face rather than the entire back face plane. The detonation wave in this case is spherical rather than planar.

The Johnson–Cook fracture model was not used for the copper liner material, following suggestions in work by Schraml.[4]

The remaining material models are unchanged. To summarize, the copper liner material is modeled using a Mie–Grüneisen equation of state (EOS) and the Johnson–Cook constitutive model, while the LX-14-0 explosive is modeled using a Jones–Wilkins–Lee EOS. A more detailed discussion of material models, as well as specific model parameter values, can be found in the prior work.[1,2]

Regarding the Mie–Grüneisen EOS, the derivation of one specific version of this type of EOS was previously provided.[1] Upon further investigation it was discovered that CTH and ALEGRA use a different implementation.[9] In interests of accuracy and completeness, this version will be documented here.

2.2 The Mie–Grüneisen Equation-of-State Model

A standard formulation for a Mie–Grüneisen EOS[10] is

$$P = \rho^2 \frac{dU(\rho)}{d\rho} + \left(\frac{\partial P(\rho,E)}{\partial E}\right)_\rho [E - U(\rho)]. \quad (23)$$

This equation relates the thermodynamics state variables pressure P, density ρ, and specific internal energy E (with units of energy per mass). U is the specific interatomic potential energy, which is a portion of E; the remaining part (E-U) accounts for the atomic rotational-vibrational specific energy.

We introduce the Grüneisen parameter Γ, defined as

$$\Gamma(\rho) \equiv \frac{1}{\rho}\left(\frac{\partial P(\rho,E)}{\partial E}\right)_{\rho} = \frac{\alpha_v}{\kappa C_v \rho}. \tag{24}$$

The right-hand side of Eq. 24 expresses the Grüneisen parameter in terms of various measurable material properties: the volumetric thermal expansion coefficient α_v, the isothermal compressibility κ, and the specific heat capacity C_v. For many materials these properties are treated as being independent of density, which implies $\rho\,\Gamma(\rho) = \rho_0\Gamma(\rho_0) \equiv \rho_0\Gamma_0$. Substituting into Eq. 23 gives us

$$P = \rho^2 \frac{dU(\rho)}{d\rho} + \rho_0\Gamma_0[E - U(\rho)]. \tag{25}$$

U can be eliminated if the thermodynamic states of the material are known along some reference curve $\{(P', \rho', E')\}$. A "reference curve" denotes a set of known equilibrium states for a material. For our purposes, we require a reference curve that spans some range of densities. For any fixed density ρ^*, the relationship between pressure and specific energy at that density (the isodensity curve) is

$$P = (\rho^*)^2 \frac{dU(\rho^*)}{d\rho} + \rho_0\Gamma_0[E - U(\rho^*)]. \tag{26}$$

If ρ^* is within the range of density spanned by our reference curve, then a known point on the isodensity curve is the reference state $(P^*, \rho^*, E^*) \in \{(P', \rho', E')\}$:

$$P^* = (\rho^*)^2 \frac{dU(\rho^*)}{d\rho} + \rho_0\Gamma_0[E^* - U(\rho^*)]. \tag{27}$$

Subtracting Eq. 27 from Eq. 26 gives us a relationship between a change in pressure and a change in energy along the ρ^* isodensity curve,

$$P - P^* = \rho_0\Gamma_0(E - E^*). \tag{28}$$

This allows us to compute a value for pressure given the specific internal energy (or vice versa) for any density within the range of the reference curve. Stated another way, general thermodynamic equilibrium states (P, ρ, E) can be computed from a reference state $(P^*, \rho^*, E^*) \in \{(P', \rho', E')\}$ for any $\rho = \rho^* \in \{\rho'\}$. An important requirement is that the density ρ must match a density ρ^* within the range

of the reference curve $\{(P', \rho', E')\}$ to cancel the density and interatomic potential contributions in Eqs. 26 and 27.

Where shock physics are involved, such as in ballistic impacts, the Hugoniot relations provide a natural reference curve. A Hugoniot curve is the set of thermodynamic states $\{(P_h, \rho_h, E_h)\}$ achievable from a given initial state (P_0, ρ_0, E_0) via shock. The choice of initial state is important, as different starting states result in different Hugoniot curves; the state at normal temperature and pressure is commonly used. For compressive shocks the Hugoniot states form a curve along a continuous density path from ρ_0 to some maximum attainable shock compression $\rho_{h\max}$. Equations for the Hugoniot states can be generated using the conservation equations for material properties across a shock front,

$$\text{Mass:} \quad \frac{\rho_0}{\rho_h} = \frac{u_s - u_p}{u_s} = 1 - \frac{u_p}{u_s} = 1 - \chi_h, \tag{29}$$

$$\text{Momentum:} \quad P_h - P_0 = \rho_0 u_s u_p = \rho_0 \chi_h u_s^2, \tag{30}$$

$$\text{Energy:} \quad E_h - E_0 = \frac{u_p^2}{2} = \frac{\chi_h}{2\rho_0}(P_h + P_0), \tag{31}$$

where we have defined a new variable based on density

$$\chi \equiv 1 - \frac{\rho_0}{\rho}. \tag{32}$$

Material density potentially ranges from 0 to $+\infty$, so χ has values between $-\infty$ and 1. $\chi < 0$ represents material dilation relative to initial state ρ_0, and $\chi > 0$ represents compression. Note that the initial pressure P_0 is often taken to be approximately 0, but we will retain this term in our derivation.

The quantities u_s and u_p are the shock front velocity and material velocity, respectively. From Eq. 29 we see that the ratio of these velocities is χ_h, and

$$u_p = \chi_h u_s. \tag{33}$$

A polynomial function is often fit to experimental measurements of these velocities,

$$u_s = \mathrm{C}_s + s_1 u_p + \frac{s_2}{\mathrm{C}_s} u_p^2. \tag{34}$$

C_s is the bulk sound speed of the material, while s_1 and s_2 are empirical fit parameters. From here we will consider only the linear case ($s_2 = 0$),

$$u_s = \mathrm{C}_s + s_1 u_p = \mathrm{C}_s + s_1 \chi_h u_s \rightarrow u_s = \frac{\mathrm{C}_s}{1 - s_1 \chi_h}. \tag{35}$$

With the Hugoniot as a reference curve, Eq. 28 can be used to relate P and E for any ρ within the range of the Hugoniot,

$$P - P_h = \rho_0 \Gamma_0 (E - E_h);\ (\rho_0 \leq \rho < \rho_{h\max}). \tag{36}$$

Substituting for P_h using Eq. 30,

$$P - P_0 - \rho_0 \chi u_s^2 = \rho_0 \Gamma_0 (E - E_h);\ (0 \leq \chi < \chi_{h\max}), \tag{37}$$

where $\chi_{h\max} = 1 - \rho_0/\rho_{h\max}$.

In the previous derivation of a Mie–Grüneisen EOS,[1] at this stage Eq. 31 is used to substitute for E_h, and (P, E) are then related back to the initial state (P_0, E_0). The CTH/ALEGRA implementation makes use of the relationship between specific internal energy E and temperature T,

$$\mathrm{C}_v \equiv \left(\frac{\partial E}{\partial T}\right)_\rho, \tag{38}$$

where C_v is the constant-volume specific heat capacity. A common approach is to assume C_v is constant. Integrating Eq. 38 from some Hugoniot reference state yields

$$E - E_h = \mathrm{C}_v (T - T_h). \tag{39}$$

T_h is the temperature associated with the Hugoniot reference state. Substituting Eq. 39 into Eq. 37,

$$P - P_0 = \rho_0 \chi u_s^2 + \mathrm{C}_v \rho_0 \Gamma_0 (T - T_h);\ (0 \leq \chi < \chi_{h\max}). \tag{40}$$

We must now devise a means of determining T_h. We begin by considering thermodynamic states along the T = 0 K isotherm, commonly called the "cold curve". For the states $\{(P_c, \rho_c, E_c)\}$ along the cold curve with densities within the range of the Hugoniot curve, an expression for the Hugoniot temperature T_h can be obtained using Eq. 39,

$$E_c - E_h = \mathrm{C}_v (0 - T_h) \rightarrow \mathrm{C}_v T_h = E_h - E_c;\ (\rho_0 \leq \rho_c < \rho_{h\max}). \tag{41}$$

We can use Eq. 28 to relate (P_c, E_c) for densities within the Hugoniot range,

$$P_c - P_h = \rho_0 \Gamma_0 (E_c - E_h);\ (\rho_0 \leq \rho_c < \rho_{h\max}). \tag{42}$$

Going back to the initial EOS equation, the right side of Eq. 23 conveniently separates the specific internal energy into athermal and thermal contributions. At 0 K only the athermal component remains,

$$E_c = U(\rho), \tag{43}$$

and Eq. 23 becomes

$$P_c = \rho^2 \frac{dE_c}{d\rho}. \tag{44}$$

A clever manipulation of Eq. 32 gives us

$$\frac{d\chi}{d\rho} = \frac{d}{d\rho}\left(1 - \frac{\rho_0}{\rho}\right) = \frac{\rho_0}{\rho^2} \rightarrow \rho^2 = \rho_0 \frac{d\rho}{d\chi}. \tag{45}$$

Combining Eqs. 44 and 45,

$$P_c = \rho^2 \frac{dE_c}{d\rho} = \rho_0 \frac{d\rho}{d\chi}\frac{dE_c}{d\rho} = \rho_0 \frac{dE_c}{d\chi}. \tag{46}$$

Substituting Eq. 46 into Eq. 42,

$$\rho_0 \frac{dE_c}{d\chi} = P_h + \rho_0 \Gamma_0 (E_c - E_h);\ (\rho_0 \le \rho_c < \rho_{h\max}). \tag{47}$$

We now have a differential equation relating P_h, E_h, E_c and χ. Rearranging terms,

$$\rho_0 \left(\frac{dE_c}{d\chi} - \Gamma_0 E_c\right) = P_h - \rho_0 \Gamma_0 E_h;\ (\rho_0 \le \rho_c < \rho_{h\max}). \tag{48}$$

The left side is an exact differential,

$$\rho_0 e^{\Gamma_0 \chi}\left[\frac{d}{d\chi}\left(\frac{E_c}{e^{\Gamma_0 \chi}}\right)\right] = P_h - \rho_0 \Gamma_0 E_h;\ (\rho_0 \le \rho_c < \rho_{h\max}). \tag{49}$$

Moving the coefficient terms to the right and integrating both sides from the initial state ρ_0 (i.e., $\chi = 0$) to some density within the Hugoniot range,

$$\frac{E_c(\chi)}{e^{\Gamma_0 \chi}} - E_c(\rho_0) = \int_0^\chi \frac{P_h - \rho_0 \Gamma_0 E_h}{\rho_0 e^{\Gamma_0 \chi}} d\chi\,;\ (0 \le \chi < \chi_{h\max}). \tag{50}$$

Substituting for E_h using Eq. 31,

$$\begin{aligned}\frac{E_c(\chi)}{e^{\Gamma_0 \chi}} - E_c(\rho_0) &= \frac{1}{\rho_0}\int_0^\chi e^{-\Gamma_0 \chi}\left[P_h\left(1 - \frac{\Gamma_0 \chi}{2}\right) - \frac{\Gamma_0 P_0 \chi}{2} - \rho_0 \Gamma_0 E_0\right] d\chi \\ &= \frac{1}{\rho_0}\int_0^\chi \frac{P_h}{e^{\Gamma_0 \chi}}\left(1 - \frac{\Gamma_0 \chi}{2}\right) d\chi - \frac{P_0}{\rho_0}\int_0^\chi \frac{\Gamma_0 \chi}{2e^{\Gamma_0 \chi}} d\chi - E_0 \int_0^\chi \frac{\Gamma_0}{e^{\Gamma_0 \chi}} d\chi. \end{aligned} \tag{51}$$

The first integral can be integrated by parts,

$$\begin{aligned}\int_0^\chi \frac{P_h}{e^{\Gamma_0 \chi}}\left(1 - \frac{\Gamma_0 \chi}{2}\right) d\chi &= \int_0^\chi \left[\frac{P_h}{2\chi}\right]\left[\frac{2\chi}{e^{\Gamma_0 \chi}}\left(1 - \frac{\Gamma_0 \chi}{2}\right)\right] d\chi \\ &= \left[\frac{P_h}{2\chi}\frac{\chi^2}{e^{\Gamma_0 \chi}}\right]_{\chi=0}^{\chi} - \int_0^\chi \frac{\chi^2}{e^{\Gamma_0 \chi}}\left[\frac{d}{d\chi}\left(\frac{P_h}{2\chi}\right)\right] d\chi \end{aligned}$$

$$= \frac{P_h \chi}{2e^{\Gamma_0 \chi}} - \int_0^{\chi} \frac{\chi^2}{e^{\Gamma_0 \chi}} \left[\frac{d}{d\chi}\left(\frac{P_h}{2\chi}\right)\right] d\chi. \quad (52)$$

Using Eq. 30 for P_h, the differential term in Eq. 52 becomes

$$\frac{d}{d\chi}\left(\frac{P_h}{2\chi}\right) = \frac{d}{d\chi}\left(\frac{P_0 + \rho_0 \chi u_s^2}{2\chi}\right) = -\frac{P_0}{2\chi^2} + \rho_0 u_s \frac{du_s}{d\chi}. \quad (53)$$

Substituting Eq. 53 into Eq. 52,

$$\int_0^{\chi} \frac{P_h}{e^{\Gamma_0 \chi}} \left(1 - \frac{\Gamma_0 \chi}{2}\right) d\chi = \frac{P_h \chi}{2e^{\Gamma_0 \chi}} + \int_0^{\chi} \frac{P_0}{2e^{\Gamma_0 \chi}} d\chi - \int_0^{\chi} \frac{\rho_0 \chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi. \quad (54)$$

Carrying Eq. 54 into Eq. 51,

$$\frac{E_c(\chi)}{e^{\Gamma_0 \chi}} - E_c(\rho_0) = \frac{P_h \chi}{2\rho_0 e^{\Gamma_0 \chi}} + \frac{P_0}{\rho_0} \int_0^{\chi} \frac{1 - \Gamma_0 \chi}{2e^{\Gamma_0 \chi}} d\chi$$

$$- E_0 \int_0^{\chi} \frac{\Gamma_0}{e^{\Gamma_0 \chi}} d\chi - \int_0^{\chi} \frac{\chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi. \quad (55)$$

The first two integrals have exact solutions:

$$\int_0^{\chi} \frac{1 - \Gamma_0 \chi}{2e^{\Gamma_0 \chi}} d\chi = \frac{\chi}{2e^{\Gamma_0 \chi}}. \quad (56)$$

$$\int_0^{\chi} \frac{\Gamma_0}{e^{\Gamma_0 \chi}} d\chi = 1 - \frac{1}{e^{\Gamma_0 \chi}}. \quad (57)$$

Combining Eqs. 55, 56, and 57,

$$\frac{E_c(\chi)}{e^{\Gamma_0 \chi}} - E_c(\rho_0) = \frac{(P_h + P_0)\chi}{2\rho_0 e^{\Gamma_0 \chi}} - E_0 \left(1 - \frac{1}{e^{\Gamma_0 \chi}}\right) - \int_0^{\chi} \frac{\chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi. \quad (58)$$

Rearranging and collecting terms,

$$E_c(\chi) = \left[E_0 + \frac{(P_h + P_0)\chi}{2\rho_0}\right] + e^{\Gamma_0 \chi}[E_c(\rho_0) - E_0] - e^{\Gamma_0 \chi} \int_0^{\chi} \frac{\chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi. \quad (59)$$

Recognizing the term in the first bracket as E_h from Eq. 31, we use Eq. 41 to arrive at

$$C_v T_h = E_h - E_c(\chi) = e^{\Gamma_0 \chi}[E_0 - E_c(\rho_0)] + e^{\Gamma_0 \chi} \int_0^{\chi} \frac{\chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi. \quad (60)$$

Noting that the temperature T_h at initial condition ρ_0 must be the initial state temperature T_0,

$$C_v T_0 = E_0 - E_c(\rho_0), \quad (61)$$

Eq. 60 becomes

$$T_h = e^{\Gamma_0 \chi} T_0 + \frac{e^{\Gamma_0 \chi}}{\mathrm{C}_v} \int_0^{\chi} \frac{\chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi \,;\ (0 \le \chi < \chi_{h\max}). \tag{62}$$

The remaining integral term must be computed numerically. The CTH/ALEGRA implementation fits a polynomial *Y* defined as

$$Y(\chi) = \frac{e^{\Gamma_0 \chi}}{\chi^2 u_s^2} \int_0^{\chi} \frac{\chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi. \tag{63}$$

Equation 62 then becomes

$$T_h = e^{\Gamma_0 \chi} T_0 + \frac{\chi^2 u_s^2}{\mathrm{C}_v} Y(\chi);\ (0 \le \chi < \chi_{h\max}). \tag{64}$$

The combination of Eq. 40 and Eq. 64 gives an EOS relating (P, ρ, T) over the Hugoniot density range $0 \le \chi < \chi_{h\max}$. From Eq. 34, $\chi_{h\max} = 1/s_1$, where u_s becomes infinite.

In the rarefaction region $\chi < 0$, T_h is linearly extrapolated using

$$T_h(\chi < 0) = T_h(\chi = 0) + \chi \lim_{\chi \to 0^+} \frac{dT_h}{d\chi}, \tag{65}$$

where the derivative is

$$\frac{dT_h}{d\chi} = \frac{d}{d\chi}\left[e^{\Gamma_0 \chi} T_0 + \frac{\chi^2 u_s^2}{\mathrm{C}_v} Y(\chi)\right] = \Gamma_0 e^{\Gamma_0 \chi} T_0 + \frac{d}{d\chi}\left[\frac{\chi^2 u_s^2}{\mathrm{C}_v} Y(\chi)\right]. \tag{66}$$

The last term in Eq. 66 vanishes at $\chi = 0$, and Eq. 65 becomes

$$T_h(\chi < 0) = T_0 + \chi\, \Gamma_0 T_0 = (1 + \chi\, \Gamma_0) T_0. \tag{67}$$

An additional condition when $\chi < 0$ is that the shock front velocity u_s is set equal to the material sound speed C_s, which allows Eq. 35 to be extended to include rarefaction shocks,

$$u_s = \frac{\mathrm{C}_s}{1 - s_1 \max(\chi, 0)};\ (\chi < \chi_{h\max}). \tag{68}$$

The full CTH/ALEGRA Mie–Grüneisen EOS is

$$P - P_0 = \rho_0 \chi u_s^2 + \mathrm{C}_v \rho_0 \Gamma_0 (T - T_h) \tag{69}$$

$$T_h = \begin{cases} e^{\Gamma_0 \chi} T_0 + \dfrac{\chi^2 u_s^2}{\mathrm{C}_v} Y(\chi) & ;\ \chi \ge 0 \\ (1 + \chi\, \Gamma_0) T_0 & ;\ \chi < 0 \end{cases} \tag{70}$$

$$Y(\chi) = \frac{e^{\Gamma_0 \chi}}{\chi^2 u_s^2} \int_0^{\chi} \frac{\chi^2 u_s}{e^{\Gamma_0 \chi}} \frac{du_s}{d\chi} d\chi\ . \tag{71}$$

One final task is calculating the specific internal energy. In the compression regime $\chi > 0$, applying Eq. 33 to Eq. 31 yields

$$E_h - E_0 = \frac{u_p^2}{2} = \frac{\chi_h^2 u_s^2}{2}. \quad (72)$$

Substituting for E_h in Eq. 39 gives us an equation for E in compression:

$$E(\chi > 0) = \frac{\chi_h^2 u_s^2}{2} + \mathrm{C}_v(T - T_h) + E_0. \quad (73)$$

In the initial state $\chi_h = 0$, $T_h(\chi_h = 0) = T_0 = T(\chi = 0)$, and $E(\chi = 0) = E_0$, as it should.

The rarefaction condition $\chi < 0$ requires more work. If we integrate Eq. 38 starting from a cold curve reference state, we have

$$E - E_c = \mathrm{C}_v(T - T_c) = \mathrm{C}_v T. \quad (74)$$

Taking the expression for P_c from Eq. 46 and inserting into the EOS relation Eq. 69 yields

$$P_c - P_0 = \rho_0 \chi u_s^2 + \mathrm{C}_v \rho_0 \Gamma_0 (T_c - T_h) = \rho_0 \chi \mathrm{C}_s^2 - \mathrm{C}_v \rho_0 \Gamma_0 T_h, \quad (75)$$

where we have made use of $u_s = \mathrm{C}_s$ in rarefaction. Substituting the equation for T_h from Eq. 70 and rearranging terms gives

$$P_c = \rho_0 \chi (\mathrm{C}_s^2 - \mathrm{C}_v \Gamma_0^2 T_0) + \rho_0 \left(\frac{P_0}{\rho_0} - \mathrm{C}_v \Gamma_0 T_0 \right). \quad (76)$$

From Eq. 46,

$$\frac{dE_c}{d\chi} = \frac{P_c}{\rho_0} = \chi (\mathrm{C}_s^2 - \mathrm{C}_v \Gamma_0^2 T_0) + \left(\frac{P_0}{\rho_0} - \mathrm{C}_v \Gamma_0 T_0 \right). \quad (77)$$

We can integrate to obtain

$$E_c(\chi < 0) = \frac{\chi^2}{2} (\mathrm{C}_s^2 - \mathrm{C}_v \Gamma_0^2 T_0) + \chi \left(\frac{P_0}{\rho_0} - \mathrm{C}_v \Gamma_0 T_0 \right) + E_c(\rho_0). \quad (78)$$

The last term can be replaced using Eq. 61,

$$E_c(\chi < 0) = \frac{\chi^2}{2} (\mathrm{C}_s^2 - \mathrm{C}_v \Gamma_0^2 T_0) + \chi \left(\frac{P_0}{\rho_0} - \mathrm{C}_v \Gamma_0 T_0 \right) - \mathrm{C}_v T_0 + E_0. \quad (79)$$

Substituting into Eq. 74 yields the specific internal energy in rarefaction,

$$E(\chi < 0) = \frac{\chi^2}{2} (\mathrm{C}_s^2 - \mathrm{C}_v \Gamma_0^2 T_0) + \chi \left(\frac{P_0}{\rho_0} - \mathrm{C}_v \Gamma_0 T_0 \right) + \mathrm{C}_v (T - T_0) + E_0. \quad (80)$$

Again, $E(\chi = 0) = E_0$, so Eqs. 73 and 80 together form a continuous curve. With a few additional steps these equations can be combined into a single expression. Rearranging the expression for T_h in the rarefaction regime from Eq. 70,

$$\chi \Gamma_0 T_0 = T_h - T_0. \tag{81}$$

Substituting into Eq. 80 and rearranging terms yields

$$E(\chi < 0) = \frac{\chi^2 C_s^2}{2} + \frac{\chi P_0}{\rho_0} + C_v \left(T - T_h - \frac{[\chi \Gamma_0]^2}{2} T_0 \right) + E_0. \tag{82}$$

Finally, we are free to set the zero point of the energy scale, which we can do by selecting a specific value for E_0. Setting $E_0 = C_v T_0$, a combined form of Eqs. 73 and 82 can be written as

$$E = \frac{\chi^2 u_s^2}{2} + \frac{\max(\chi,0) P_0}{\rho_0} + C_v \left[T - T_h + T_0 \left(1 - \frac{[\max(\chi,0)\Gamma_0]^2}{2} \right) \right]. \tag{83}$$

3. Computational Setup

All simulations were run on an SGI ICE XA supercomputer with hostname CENTENNIAL located at the US Army Combat Capabilities Development Command Army Research Laboratory and managed by its Department of Defense Supercomputing Resource Center. CENTENNIAL features 1784 standard compute nodes with dual Intel Xeon E5-2698 v4 2.2-GHz 20-core processors and 128 GB of memory each.

In total, 101 2-D simulations and 84 3-D simulations were completed. The total computing time used was 102,000 core-h for the 2-D simulations, and 545,000 core-h for the 3-D simulations.

3.1 Simulation Parameters

Simulations were run using version 7.7, release 2019.12.18 of the ALEGRA code.[3] Code parameters were configured as in previous work,[2] with the only difference being a reduction in the memory allocation multiplier from the default value of 5 by setting FIELD ALLOCATION: MULTIPLIER = 1.5.

2-D simulations used a cylindrical coordinate system with the *x*-axis corresponding to the radial direction and the *y*-axis being the symmetry axis of the SCJ device. In this coordinate system *x* is restricted to nonnegative values. The end of the SCJ liner is initially positioned at the coordinate origin, and the device is oriented so that the jet advances in the positive *y* direction. The extent of the computational

domain varied with resolution r. The number of radial elements was determined by setting the maximum coordinate value $x_{\max}$ to the smallest value greater than 6 cm,

$$x_{\max}^{(2\text{-D})} = r * \text{ceiling}(6/r). \tag{84}$$

The coordinate limits along the y axis were calculated similarly, with the target values for the minimum and maximum being –14 cm and 24 cm, respectively:

$$y_{\min}^{(2\text{-D})} = -r * \text{ceiling}(14/r), \tag{85}$$

$$y_{\max}^{(2\text{-D})} = r * \text{ceiling}(24/r). \tag{86}$$

3-D simulations used a Cartesian coordinate system, with the SCJ device positioned so that the jet advanced along the z-axis in the positive direction. The end of the liner was placed at $z = 0$, and the domain extents given by

$$x_{\min}^{(3\text{-D})} = -r * \text{ceiling}(6/r). \tag{87}$$

$$x_{\max}^{(3\text{-D})} = r * \text{ceiling}(6/r). \tag{88}$$

$$y_{\min}^{(3\text{-D})} = -r * \text{ceiling}(6/r). \tag{89}$$

$$y_{\max}^{(3\text{-D})} = r * \text{ceiling}(6/r). \tag{90}$$

$$z_{\min}^{(3\text{-D})} = -r * \text{ceiling}(14/r). \tag{91}$$

$$z_{\max}^{(3\text{-D})} = r * \text{ceiling}(24/r). \tag{92}$$

With this structure the coordinate origin is always coincident with a mesh node. All domains had free outflow boundaries.

The only objects in these simulations are the copper liner and the explosive charge, with the remainder of the domain being void space. As in prior work,[2] a minimal set of material discards were used to ensure simulations would reliably run to completion. All material with temperature over 10,000 K or specific internal energy less than –100 kJ/g was discarded. Low-pressure explosive gas was removed by discarding explosive material with pressure less than 101,325 Pa (1 atm) and density less than 0.05 g/cm^3. All but one simulation ran to completion with these discards. The exception was a 3-D simulation at 210-μm resolution, which failed between 32 and 33 μs due to an element inversion. An additional discard eliminating jet material with density greater than 25 g/cm^3 after 32.7 μs allowed this simulation to run to completion.

Simulations were run to 40 μs of simulated time. Image data were generated in 1-μs intervals using the integrated Spymaster capability.[11] The computational domain was recorded every 5 μs in the Exodus II format,[12] with data at 5 μs skipped as the detonation wave had not yet reached the liner at that time.

3.2 Project Workflow

Simulations were run in 10-μm increments from a resolution of 990 μm down to the smallest resolution that would run to completion within 168 h on no more than 4000 cores. Additional 2-D simulations were run at 999, 25, and 15 μm. Additional 3-D simulations were run at 999 and 175 μm.

Simulation workflow (Fig. 3) was heavily scripted to minimize user intervention and maintain process uniformity. The process begins by inputting a resolution, specified as an integer number of microns from 1 to 999, to a Python control script. The script calculates the required number of nodes for the simulation based on prior scaling tests, then constructs an ALEGRA input file and a script for the portable batch system (PBS) job scheduler (Altair's PBS Professional version 19.2.4) from master template files. The ALEGRA input file is passed through the APREPRO preprocessor,[13] then the PBS script is executed to begin the ALEGRA simulation.

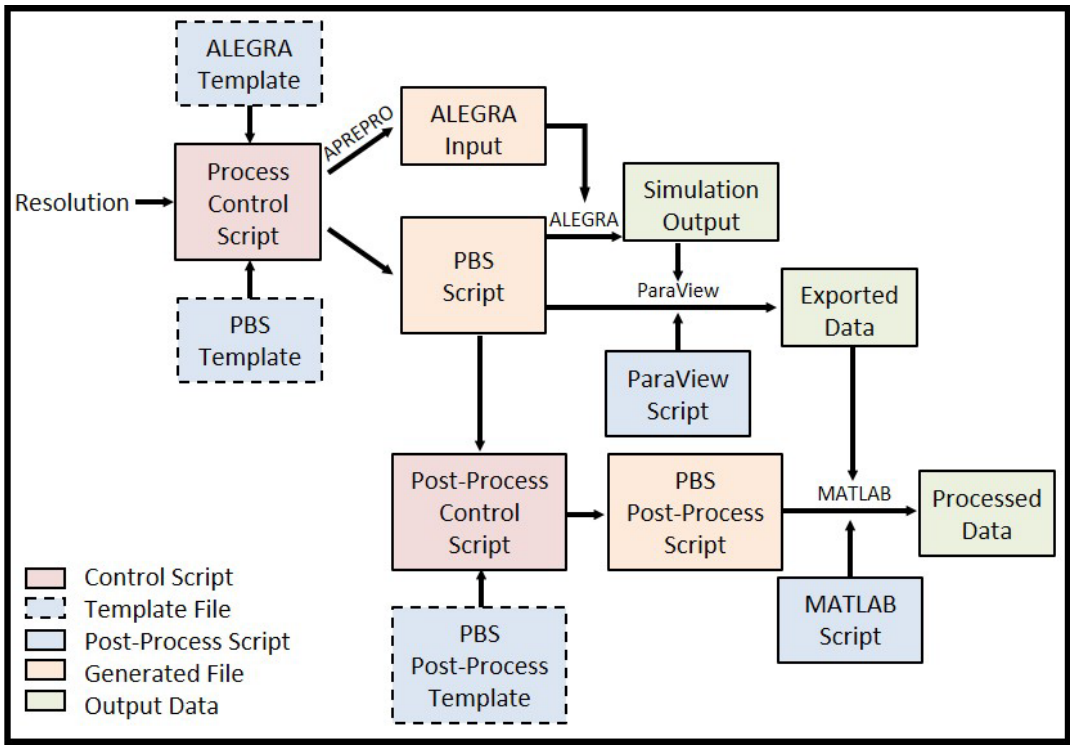

Fig. 3 Project workflow diagram

Data are exported using ParaView[14] version 5.6.1 after the simulation is completed. Copper material data in elements with material volume fraction greater than 1E-16 are written to text files. This is well below the ALEGRA minimum volume fraction of 1E-12, so this should account for all liner material in the domain. ALEGRA data is structured as a combination of element- and node-centered quantities. Element data include material volume fraction, density, pressure, temperature, and specific internal energy. Node data include coordinates and velocity. Coordinate and velocity data for element centers are generated from node data using the ParaView Point-to-Cell function.

A moment of consideration is warranted for this method of translating velocity data. The Point-to-Cell function averages the quantities of all nodes associated with an element, 8 corner nodes in the case of cubic elements. The relationship between element mass and node mass in ALEGRA is that the total mass of an element is divided equally among its associated nodes. It follows that obtaining element velocity by averaging the node velocities is equivalent to conserving linear momentum. Kinetic energy is not conserved, however.

Suppose the cubic element in Fig. 4 has total mass M. Then each of the 8 associated nodes receives mass $M/8$ from this element. The total mass on a given node is the sum of contributions from all elements connected to that node. It follows that the element mass is not the same as the average of masses of its nodes. In a regular cubic mesh, however, the converse is true—the total mass of a node is equal to the average of the masses of its connected elements.

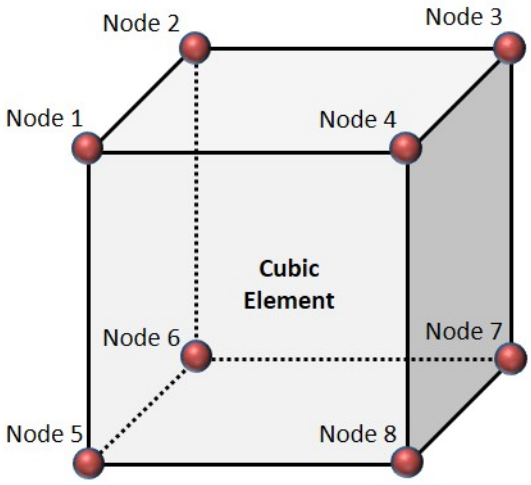

Fig. 4 Illustration of a cubic element and its associated nodes

If the velocity of node n is specified as v_n, then the total linear momentum of the nodes associated with the central element mass is

$$\frac{M}{8}\sum_{n=1}^{8} v_n = M\left(\frac{\sum_{n=1}^{8} v_n}{8}\right). \tag{93}$$

Since M is the total element mass, the element velocity that conserves linear momentum is given by the parenthetical term on the right, which is just the average node velocity. However, the kinetic energy of this mass on the nodes is generally not the same as that for the element. For example, let us set $M = 1$, and suppose $v_1 = 1$ and $v_2 = 2$, while the remaining nodes have zero velocity. The element kinetic energy is

$$\frac{1}{2}M\left(\frac{\sum_{n=1}^{8} v_n}{8}\right)^2 = \frac{1}{2}\left(\frac{1+2}{8}\right)^2 = \frac{9}{128}, \tag{94}$$

while the nodal kinetic energy associated with the element mass is

$$\frac{1}{2}\frac{M}{8}\sum_{n=1}^{8} v_n^2 = \frac{1}{16}(1^2 + 2^2) = \frac{5}{16} = \frac{40}{128} \neq \frac{9}{128}. \tag{95}$$

After the ParaView process is complete, the PBS script invokes a Python postprocess control script, which constructs and submits a PBS script for execution of a MATLAB (version 9.5.0, The MathWorks, Inc.) data analysis process.

The MATLAB analysis begins by extracting simulation parameters from various process files. Data at each recorded time are then processed in sequence. Exported element and node data are read. In a parallel simulation, individual nodes can belong to multiple pieces of the decomposed computational domain. ParaView will export data for each instance of such nodes, leading to duplicate data entries that must be filtered out.

Derived quantities are then computed. For both elements and nodes, derived quantities are material mass and absolute velocity magnitude. The derivation of node mass can be a lengthy calculation, as the set of connected elements must be identified for each node. For 3-D simulations, the radial distance from the z-axis and radial velocity component are also derived. All data are then written out as MATLAB matrix data for efficient storage.

The analysis then proceeds to calculate, record, and plot an array of jet characteristics. Examples include distributions of mass as a function of material density, pressure, temperature, axial position, and radial position; distributions of mass, specific internal energy, kinetic energy, and mass-averaged temperature as a function of material axial velocity; a reverse cumulative mass distribution of the jet as a function of axial velocity (commonly referred to as a V_jM plot)[4]; and global scalar values including total mass, total kinetic energy, total axial momentum, and thermodynamic state values at the locus of maximum jet material pressure (referred to as the "pressure head" from here on).

Upon completion of the MATLAB analysis, which can take several days for high-resolution 3-D simulations, the postprocess PBS script collects and organizes the process scripting, input files, and output data for transfer and archiving.

4. Results and Discussion

As indicated in the previous section, an abundance of information was generated for each simulation, too much for a full accounting in this report. A selection of results and analyses best illustrating how simulation outcomes vary with changing resolution is provided.

4.1 Run Time

One simple outcome of a simulation is its run time to completion. In parallel computing, the run time of a simulation as measured by the clock depends on how many cores are used. In principle, a problem that takes 2 days to complete using 10 cores could be run in 1 day on 20 cores. The reality is that parallelization comes with some overhead, so the scaling is not perfect, but it remains generally true that increasing the number of cores used reduces run time.

A way to account for both the run time and the amount of computing resources applied is to track the computational time, which is the total number of cores used multiplied by the run time (Fig. 5). For resolution r in millimeters, the measured computational time T_C in core-h is well fit by

$$T_C^{(3\text{-}D)} = \frac{48.73}{r^{4.304}}, \tag{96}$$

$$T_C^{(2\text{-}D)} = \max\left[0.99 - 0.288r, 2.61 - 2.93r, \frac{0.12}{r^{3.087}}\right]. \tag{97}$$

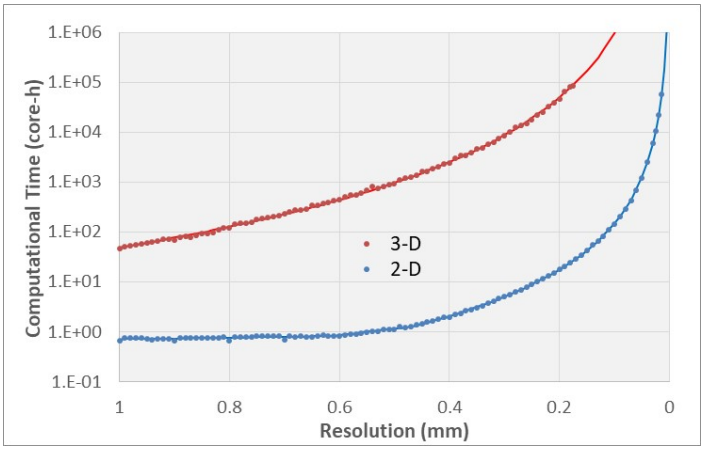

Fig. 5 Computational time (log scale) vs. resolution

Note the exponent values of r in the curve fits. Higher-resolution simulations require greater computational run times not only because they have more elements in the domain, but also because more time steps are needed to reach the simulation end time. A way of understanding this is to recognize that information should not travel farther than one element in each time step. This is a fundamental stability requirement in an explicit time formulation. The flow rate of information is set by physical properties such as shock front velocity. If the element size is reduced, smaller time steps must be taken to maintain stability. So halving the resolution doubles the number of elements along each coordinate axis and also doubles the number of time steps required. If d is the number of spatial dimensions, then the computational run time of a simulation will scale roughly as the $(d+1)^{\text{th}}$ power of inverse resolution, as seen in Eqs. 96 and 97.

Computational time can also be compared with the size of a simulation S_C as measured by total element-steps taken, which is the number of domain elements multiplied by the number of time steps taken to reach the simulation stop time (Fig. 6). The relationship is highly linear. The fit coefficients provide a measure of performance for the hardware and build of the code. The dimensionality cost can also be quantified, as 3-D element-steps take a factor of 1.76 longer to compute than 2-D element-steps.

$$T_C^{(3\text{-}D)} = 12.35\, S_C^{1.020}, \tag{98}$$

$$T_C^{(2\text{-}D)} = \ \ 7.03\, S_C^{0.999}. \tag{99}$$

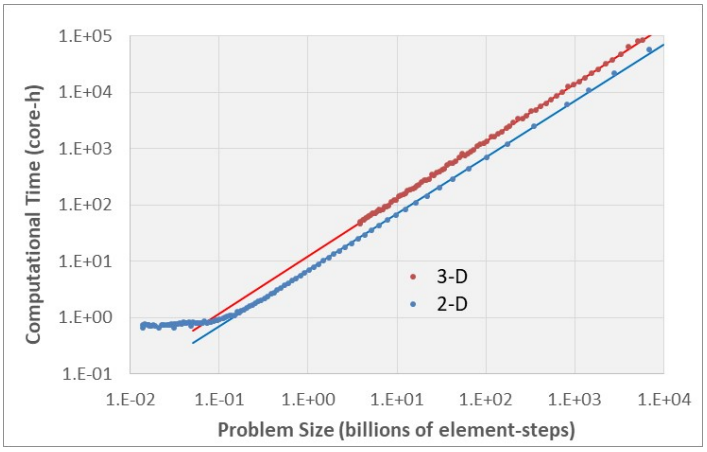

Fig. 6 Log-log plot of computational time vs. problem size

4.2 Total Liner Mass

The total mass of the copper liner in the continuum limit is 79.482 g. The process of calculating this value is described in the Appendix. At finite resolutions,

geometric model descriptions are used by the space-filling DIATOM routine to create discretized approximations of objects within the computational domain. The fidelity of this process is set by the ITERATION parameter, which sets the number of element subdivisions performed when locating object boundaries. ITERATION = 5 for all simulations in this work.

Figure 7 plots the measured initial liner mass as a function of resolution. The scatter in the 2-D data reflect the non-Cartesian nature of the 2-D cylindrical domain. 2-D elements represent annuli, with the associated volumes increasing with distance from the axis. Small changes in individual 2-D element filling will have much larger effects on total mass than in corresponding 3-D elements.

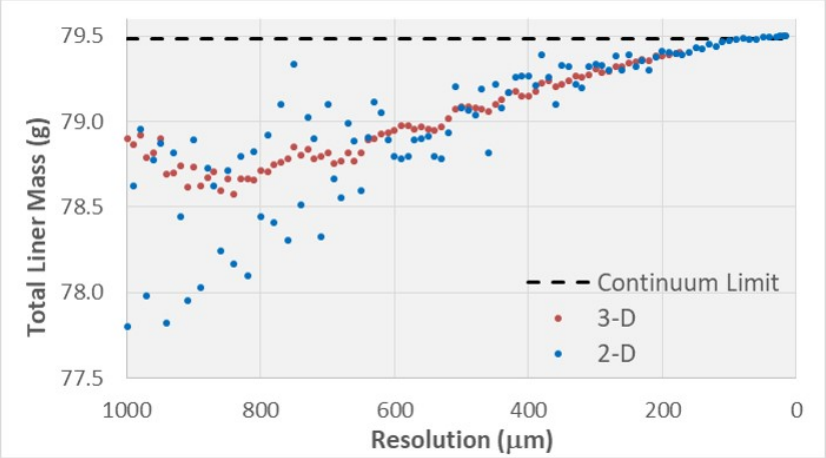

Fig. 7 Total liner mass vs. resolution at 0 μs

From 800-μm to 150-μm resolution, the difference between the continuum liner mass and the initial simulated liner mass follows a linear trend

$$\Delta M = 1.19\, r - 0.156, \tag{100}$$

where r is in millimeters and the mass difference ΔM is in grams. The coefficient of determination over this range is $R^2 = 0.75$. At the highest resolutions in 2-D, the initial simulated liner mass exceeds the continuum limit by 0.01–0.02 g.

The discretization error can exceed 1% of the continuum value in coarsely meshed simulations. Where mass loss occurs can be found by comparing the mass distributions in simulations having the largest and smallest differences from the continuum limit. In 2-D, the 999-μm simulation had an initial liner mass of 77.7981 g, while the 60-μm simulation had 79.4814 g. Radial histogram plots of the initial mass distribution show clearly that most of the 1.6838-g difference between these two simulations occurs at the end of the liner (Fig. 8).

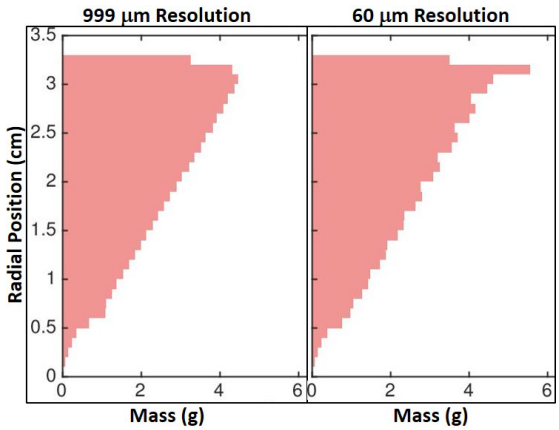

Fig. 8 Radial distribution of initial liner mass in 2-D simulations at 999-µm (left) and 60-µm (right) resolution

In 3-D, the 840-µm simulation had an initial mass of 78.5741 g, while the 175-µm simulation had 79.4056 g, a difference of 0.8315 g. A radial histogram mass plot is not as clear for the 3-D data because the radial bin size of 1 mm leads to discrete jumps in the number of included cells in the 840-µm plot. A cumulative mass difference plot presents a clearer picture (Fig. 9), with the cumulative difference scattered around zero until reaching the end of the liner.

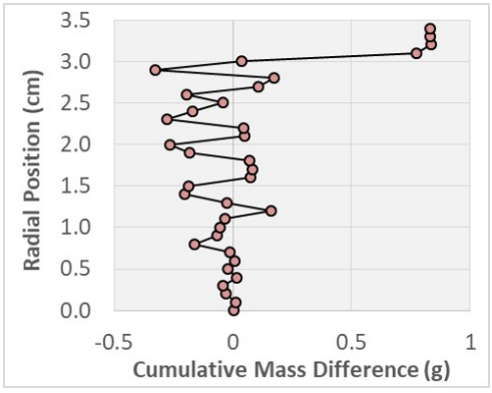

Fig. 9 Cumulative difference in initial liner mass with increasing radial position between 3-D simulations at 175-µm and 840-µm resolution

Also of interest is any measured change in total mass over the duration of a simulation. The total copper mass at 40 µs matched the initial mass to a precision of 0.0001 g in most simulations. For reference, 0.0001 g of ambient copper would just about fill a single 3-D cubic element at 224-µm resolution.

In 2-D simulations at 60-µm and better resolution, mass losses of about 0.015 g were typically measured at 40 µs. These losses can be traced to a flow of particles emanating from the end of the liner and exiting the domain radially (Fig. 10).

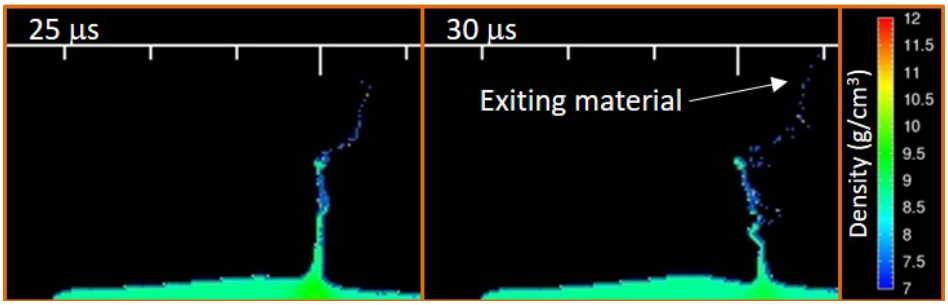

Fig. 10 Copper material density at 25 µs (left) and 30 µs (right) in a 2-D simulation at 25-µm resolution

In the 2-D simulation at 350-µm resolution, a slight mass *increase* of 0.0001 g was observed from 0 to 10 µs. While there are ways for mass to increase in certain types of simulations, that does not apply here. This change is likely due to precision or round-off errors in the ParaView or MATLAB data analyses.

In 3-D the only measured change in mass occurred at 270-µm resolution, where the total mass decreased 0.0002 g between 35 and 40 µs. Based on image data, this mass loss was due to a few small jet particles exiting the domain at the positive *z* boundary just prior to the 40-µs stop time.

4.3 Jet Temperature

Resolution has a marked effect on the temperature of the resulting jet. Jet temperature profiles are not uniform but can be reduced to an intuitive scalar value through mass averaging (Fig. 11). More heat energy develops during jet formation at coarse resolutions in both 2-D and 3-D.

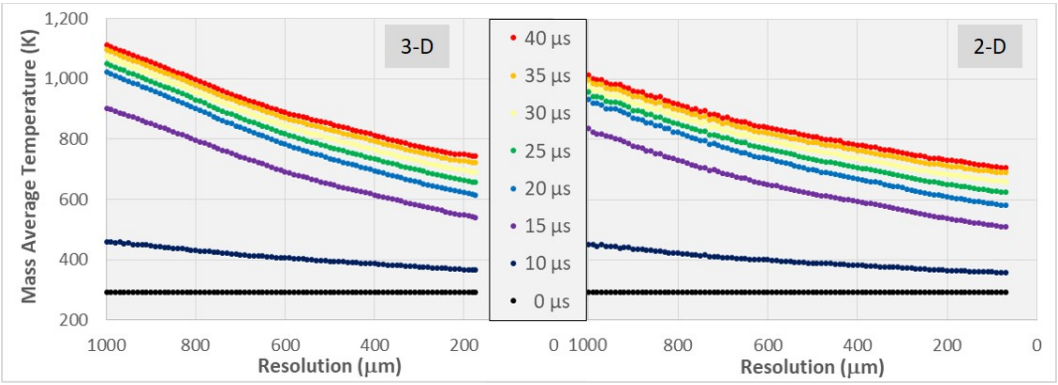

Fig. 11 Mass average temperature vs. resolution at various times in 3-D (left) and 2-D (right) simulations

The resolution dependence of jet temperature can be traced to the thermodynamic conditions at the pressure head. The pressure head is defined as the element with the highest pressure among those whose center is radially within two resolution

lengths of the jet axis, with an additional condition that the element must contain a volume fraction of copper greater than 0.5.

Peak pressure in the pressure head occurs during jet formation sometime between 10 and 20 μs. Plots of the pressure head thermodynamic state at 15 μs reveal significant dependence on resolution (Fig. 12). Data at 10 and 20 μs trend in the same way, indicating that pressure head development is not merely shifting in time with resolution, but that the thermodynamic state at peak pressure is changing.

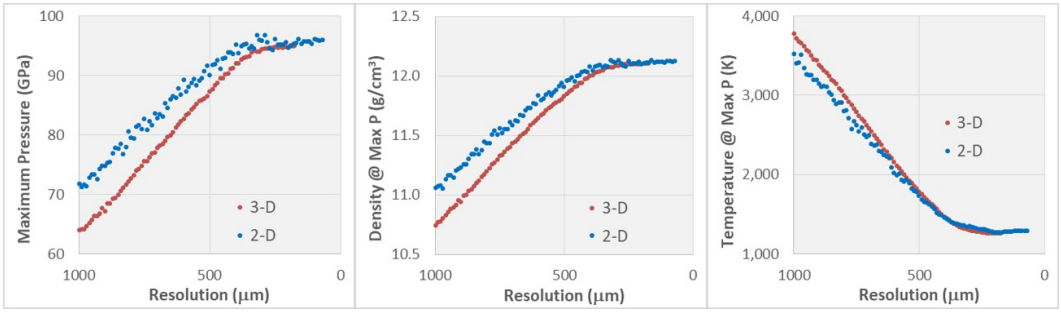

Fig. 12 Maximum pressure (left) along with the associated density (center) and temperature (right) vs. resolution at 15 μs

How resolution affects the thermodynamic state of the pressure head can be investigated by considering the numerical operations of each time step. As indicated by the forms of Eqs. 69 and 83, the EOS takes density and temperature as inputs and computes pressure and specific internal energy as outputs. The input values are the result of the equations governing dynamic material flow. Material flow is driven by pressure gradients. In an SCJ device, pressure from the shaped explosive drives liner material into the pressure head. The device geometry allows pressure relief through the base of the liner along the axis, which is why a jet forms. The material response under pressure is determined by its constitutive model combined with conservation of mass and momentum. Changes in element density will come from net material flux, while temperature changes arise from heat generated by the work of material deformation. Both of these are impacted by resolution, which not only sets the characteristic volumes and surface areas of elements, but also influences the typical duration of time steps. An interesting test of spatial versus temporal dependence would be to run a set of simulations at fixed spatial resolution and manually limit the time steps to increasingly smaller values to see if the pressure head changes shown in Fig. 12 are driven predominantly by temporal or spatial effects.

The substantial drop in jet temperature from the coarsest to the finest resolutions is clearly displayed in Fig. 13, which shows the spatial temperature profile through the $x = 0$ plane of a 3-D jet at 40 μs for both the lowest- and highest-resolution

simulations. At 999-μm resolution the leading portion of the jet exceeds 2500 K, well above the material model melting temperature of 1381 K. In contrast, the simulation at 175 μm shows the jet remaining much cooler.

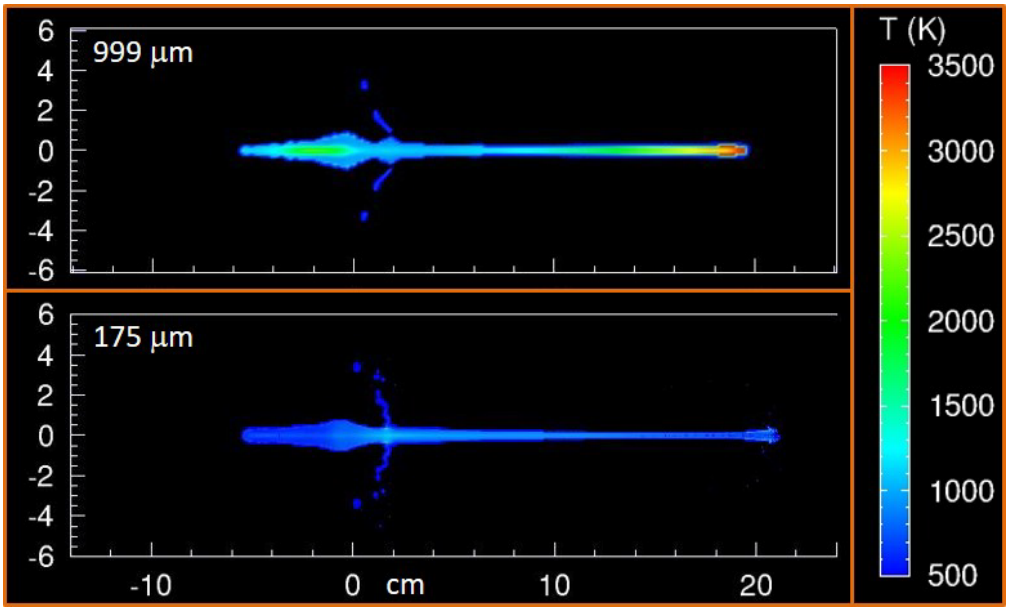

Fig. 13 Cross section images showing copper material temperature at 40 μs in 3-D simulations with 999-μm (top) and 175-μm (bottom) resolution

How changes in density ρ and temperature T affect pressure P at the pressure head can be explored using Eq. 69. For $\rho = 11.5$ g/cm^3 and $T = 2500$ K, roughly in the middle of the ranges in Fig. 12, $P = 79.7$ GPa. Increasing T to 3000 K while holding ρ fixed raises P to 83.2 GPa, while lowering T to 2000 K decreases P to 76.2 GPa. The temperature dependence of pressure is linear, which can be seen directly by differentiating Eq. 69,

$$\left(\frac{\partial P}{\partial T}\right)_{\rho} = \mathrm{C}_v \rho_0 \Gamma_0. \tag{101}$$

Holding the temperature fixed at 2500 K and increasing density to 12.0 g/cm^3 produces a pressure of 99.2 GPa, while $\rho = 11.0$ g/cm^3 gives $P = 62.7$ GPa. For the data in Fig. 12, the increase in pressure head density dominates over decreasing temperature to drive the peak pressure higher with decreasing resolution.

4.4 Resolution-Dependent Numerical Uncertainty

Figures 11 and 12 exemplify two characteristic behaviors of these simulations. The first is that jet properties change in a generally, though not strictly, monotonic manner toward some presumed limiting value at 0-μm resolution. The second is

that differences between 2-D and 3-D simulation results decrease as resolution increases. These are highly desirable traits.

Recall that 2-D and 3-D approaches can be viewed as alternate ways of approximating the same continuum system. Because they employ different computations, they can be expected to generate different outcomes. But when used to approximate the same continuum system, their limiting behavior should coincide as resolution goes to zero. 2-D simulations are typically preferred over 3-D simulations as they have shorter run times and can be performed at higher resolution. But 2-D simulations require (and enforce) a geometry with perfect cylindrical symmetry, which limits their applicability.

With some assumptions, resolution series data such as that in Figs. 11 and 12 can be used to estimate the inherent numerical variance of jet properties in a form similar to the results in Table 1, but with the added benefit that the resolution dependence is explicitly included.

Consider the total kinetic energy E_K of a jet at 40 μs in 2-D simulations (Fig. 14; note that data below 70-μm resolution are omitted due to the influence of lost mass, as discussed in Section 4.2).

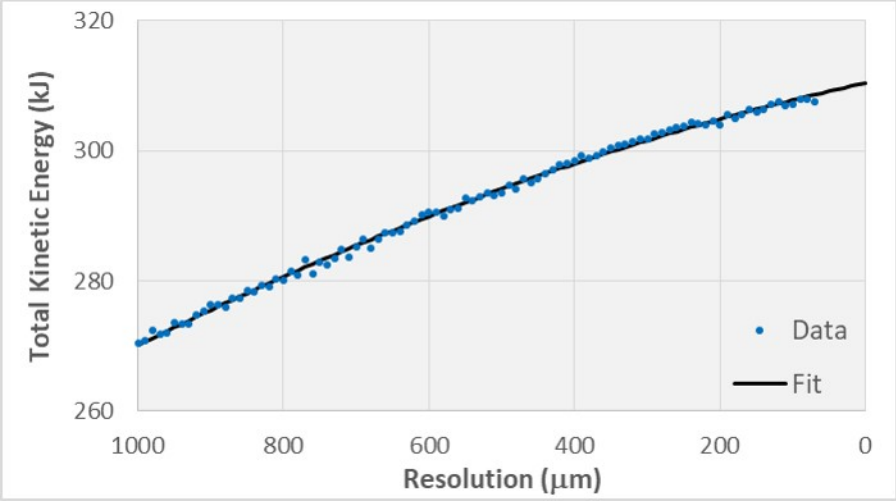

Fig. 14 Total kinetic energy vs. resolution at 40 μs in 2-D simulations

E_K, measured in kilojoules, is well fit by quadratic polynomial of resolution r in millimeters,

$$E_K = (-15.467)r^2 + (-24.895)r + 310.475, \qquad (102)$$

with a coefficient of determination $R^2 = 0.9976$. To be precise, this fit estimates E_K for simulations in which all computational factors other than resolution are fixed at the values used in this work. Holding resolution fixed and varying the

computational factors listed in Table 1 would produce a distribution of E_K values at that specific resolution. Repeating this process over a range of resolutions would allow for a fit of the actual mean E_K value as a function of resolution.

This would involve considerable work. Instead, the data in Fig. 14 are assumed to be normally distributed about the actual mean values at each resolution. The fit in Eq. 102 should then approximate the actual mean E_K as a function of resolution. Note, however, that regardless of the truth of this assumption, the limiting value of 310.475 kJ at zero resolution is the expected value of E_K for the continuum system. Recall the discussion in Section 1.3 regarding the effect of resolution on numerical variance. As resolution goes to zero, not only does the mean value of the resolution-specific distributions converge to the continuum value, but the width of the distributions themselves also collapses to zero. So even if the assumption is poor, the difference between the fit in Eq. 102 and the true mean E_K values will decrease to zero in the limit of zero resolution.

A subtle point of the working assumptions worth stating explicitly is that the differences between the measured E_K values and the estimated mean E_K values associated with the fitting curve Eq. 102 are taken as numerical estimation errors representative of the same class of variation that arises when the computational factors listed in Table 1 are varied. This type of variation is distinct from that associated with uncertainty in the selection of or parameter values for material models, or variation arising from physical aspects of the system, such as geometric deviations in real devices.

If it is accepted that Eq. 102 approximates mean E_K and the data in Fig. 14 are normally distributed about the mean, then the resolution-dependent numerical variance in E_K can be quantified as follows. Consider a pair of curves centered on the data fit in Eq. 102,

$$E_K^{(\pm)} = (-15.467 \pm A_\sigma)r^2 + (-24.895 \pm B_\sigma)r + 310.475. \quad (103)$$

These curves have the same functional form and limiting factor as the data fit and form a continuously narrowing band centered on the estimated mean E_K that vanishes as resolution goes to zero. In a normal distribution, about 68% of randomly selected data can be expected to fall within 1 standard deviation of the mean. There are 94 data points in Fig. 14, so curves $[A_\sigma, B_\sigma]$ that enclose 64 points (68.1% of the data) can be considered an estimate of the standard deviation of E_K.

Aside from hinging on the validity of debatable assumptions, a further complication is that the bounding curves are described by two variables, so there exists a set of bounding curves $\{[A_\sigma, B_\sigma]\}$, each enclosing 64 points. At the extremes, both

$[A_\sigma, B_\sigma]$ = [0.0000, 1.3505] and $[A_\sigma, B_\sigma]$ = [3.0443, 0.0000] satisfy the criteria. Another condition is needed to decide among the possible alternatives.

The assumed randomness of the E_K data provides such a condition. The plots in Fig. 15 indicate whether E_K data points fall within (1) or outside (0) the bounds formed by the indicated curves. Similar data can be plotted for all curves $\{[A_\sigma, B_\sigma]\}$.

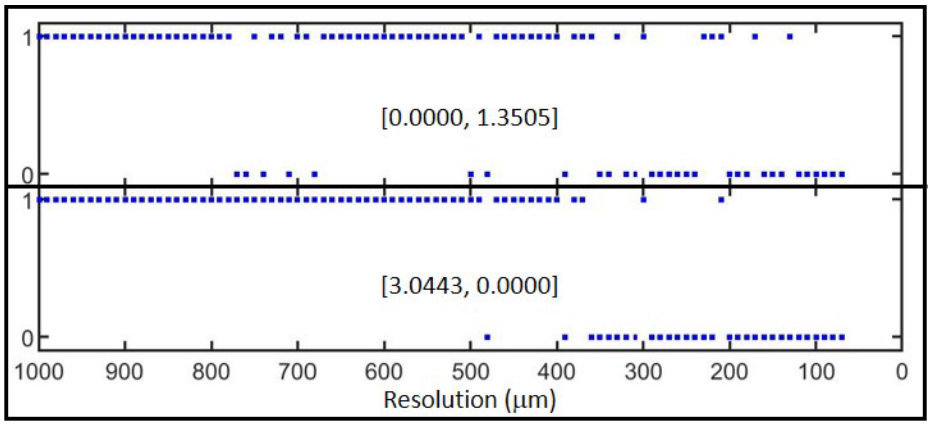

Fig. 15 Inclusion plots indicating which data points are inside (1) or outside (0) the bounds formed by two different curves

Data of this type are amenable to a statistical run test to measure the likelihood of randomness.[15] A run is simply a sequence of consecutive values. For example, a series of eight coins flips might produce a sequence of results HHTHHTTT. This sequence has four runs: HH, T, HH, TTT. For sequences of randomly occurring binomial values, the expected number of runs $\overline{N_R}$ is

$$\overline{N_R} = \frac{2n_+n_-}{n_+ + n_-} + 1, \tag{104}$$

where n_+ and n_- are the number of data points with each of the two different binomial outcomes. In this case, n_+ = 64 and n_- = 30, for an expected 41.85 runs. Furthermore, the distribution of runs from random arrangements of these data points has a standard deviation S_R given by

$$S_R^2 = \frac{2n_+n_-(2n_+n_- - n_+ - n_-)}{(n_+ + n_-)^2\,(n_+ + n_- - 1)}, \tag{105}$$

which gives S_R = 4.18. For a data set with N_R runs, the Z-score

$$Z = \frac{N_R - \overline{N_R}}{S_R} \tag{106}$$

indicates how far N_R is from the value $\overline{N_R}$ expected for a random process. For the examples in Fig. 15, the top data set has 26 runs for a Z-score of –3.79, while the

data plotted on the bottom has 10 runs and a Z-score of –7.61. Neither set is particularly random, as a Z-score with absolute value |Z| > 3.09 indicates a nonrandom data sequence at a 0.2% significance level. Still, data for the [0.0000, 1.3505] bounding curves are clearly the more randomly distributed of the two examples; in fact, they have the Z-score closest to zero out of the entire set of curves $\{[A_\sigma, B_\sigma]\}$. The lack of general randomness in the data indicates that some or all of the underlying assumptions used in this analysis are poor, or that the fitting function does not have quite the right shape, or some combination of both.

In spite of this, a quantitative estimate of the variance in E_K due to computational factors as a function of resolution can now be made. At any resolution r, the estimated standard deviation σ of the distribution of E_K outcomes is

$$\sigma = E_K^{(+)} - E_K = A_\sigma r^2 + B_\sigma r = 1.3505\, r. \tag{107}$$

Expressed in the same form as the results in Table 1, the estimated coefficient of variation CoV due to computational factors for E_K is

$$CoV = \frac{\sigma}{E_K} = \frac{1.3505r}{310.475 - 24.895r - 15.467r^2}. \tag{108}$$

So at 0.500-mm resolution, for example, the estimated expected E_K value is 294.8 kJ. With a CoV of 0.23%, the estimated ±3-sigma range of the expected E_K distribution due to inherent numerical variability (spanning 99.7% of the expected total distribution) is 292.1–296.2 kJ. The actual measured E_K at 0.500-mm resolution with the specific simulation conditions in this report falls within this range at 293.5 kJ. A true test of the methodology used to produce these estimates would be to perform a battery of simulations at 0.500-mm resolution over a range of computational factor values and see if the resulting E_K distribution is consistent with the estimated distribution. Some discrepancy would be expected given the Z-score of the bounding curves, but it is encouraging that observed CoV values for SCJ penetration into RHA in ALEGRA (see Table 1) are similar across several computational factors to the 0.23% estimate for E_K.

The analysis developed in this section was repeated for four different jet properties at 40 μs in both 2-D and 3-D simulations (Table 2). 2-D data sets contained 94 points and 3-D sets contained 84 points. Quadratic functions provided excellent data fits for three of the jet properties. Data and fits for the fourth property, total axial momentum, are plotted in Fig. 16. In spite of the relatively poor data fit and clearly nonrandom data distribution for the 3-D data, the resulting CoV estimates are within the range spanned by the other three properties.

Table 2 Data fits and estimated σ for jet properties at 40 μs

Jet property at 40 μs		Data fit $A\,r^2 + B\,r + C$				σ $A_\sigma\,r^2 + B_\sigma\,r$		CoV at r = 0.500 mm (%)
		A	B	C	R^2	A_σ	B_σ	
Total kinetic energy (kJ)	2-D	–15.467	–24.895	310.475	0.9976	0.000	1.351	0.23
	3-D	–20.075	–31.399	312.274	0.9996	0.000	0.570	0.10
Total internal energy (kJ)	2-D	6.283	3.987	22.512	0.9986	0.000	0.237	0.46
	3-D	9.256	4.558	22.821	0.9995	0.119	0.116	0.32
Mass average temperature (K)	2-D	182.632	127.695	699.396	0.9988	0.000	6.965	0.43
	3-D	246.174	164.286	704.888	0.9995	8.100	0.936	0.29
Total axial momentum (N)	2-D	–5.323	6.020	130.959	0.4577	0.000	0.976	0.37
	3-D	–0.902	2.096	132.044	0.5903	0.000	0.489	0.18

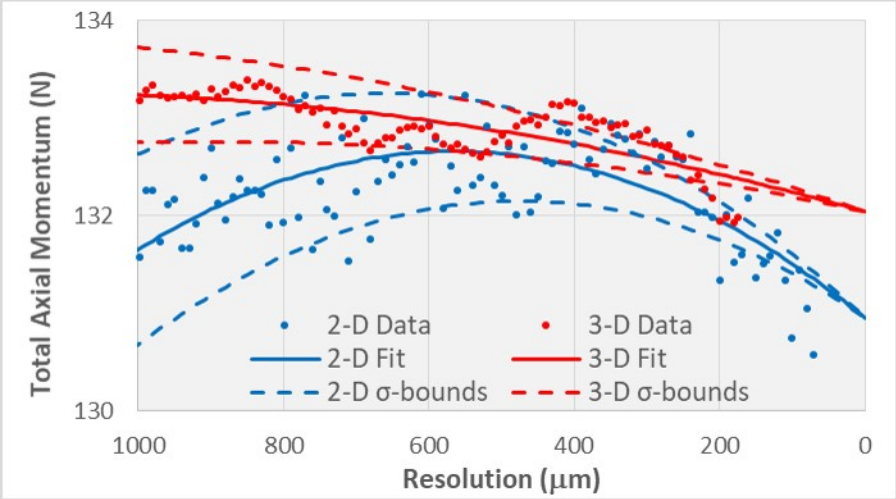

Fig. 16 Total axial momentum vs. resolution at 40 μs

One simple check on the results in Table 2 is to compare the constant terms between the 2-D and 3-D fits for each jet property. In principle, fits for a property should converge to the same value at zero resolution, so the constant terms should be equal. The difference in constant terms is less than 1.5% for all properties.

4.5 Jet Tip Morphology

One aspect of SCJ formation that is difficult to reduce to single scalar value or simple histogram is the structural morphology of the jet. High-resolution images of the leading portion of the jet were rendered to observe how the structure of the jet tip changes with resolution. Domain cells containing a volume fraction of copper greater than 0.01 are colored in these images. The resulting array of image data is

too extensive to review here in its entirety. A selection of images illustrating the changing jet tip morphology is shown in the next two figures. Figure 17 shows images from 2-D and 3-D simulations with resolutions of 500 μm and coarser, while Fig. 18 shows those for resolutions finer than 500 μm.

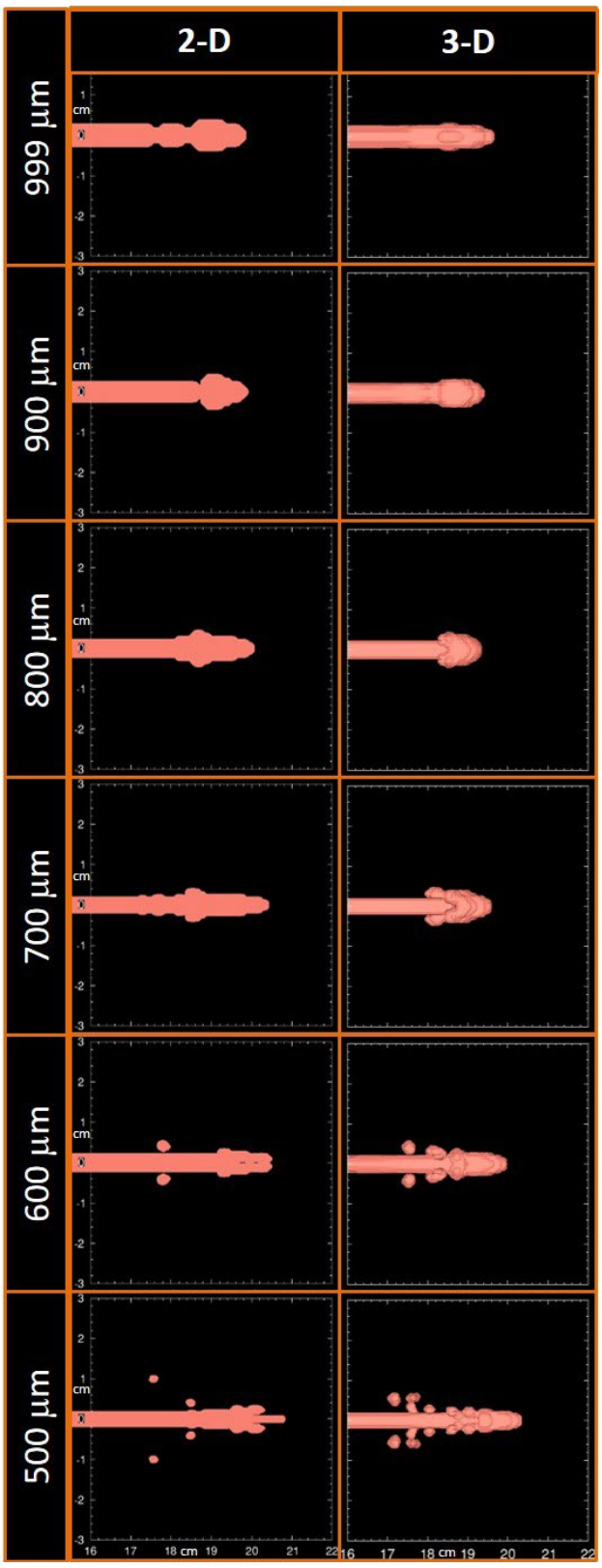

Fig. 17 Jet tip morphology at 40 µs in 2-D (left) and 3-D (right) simulations with 500 µm and coarser resolution

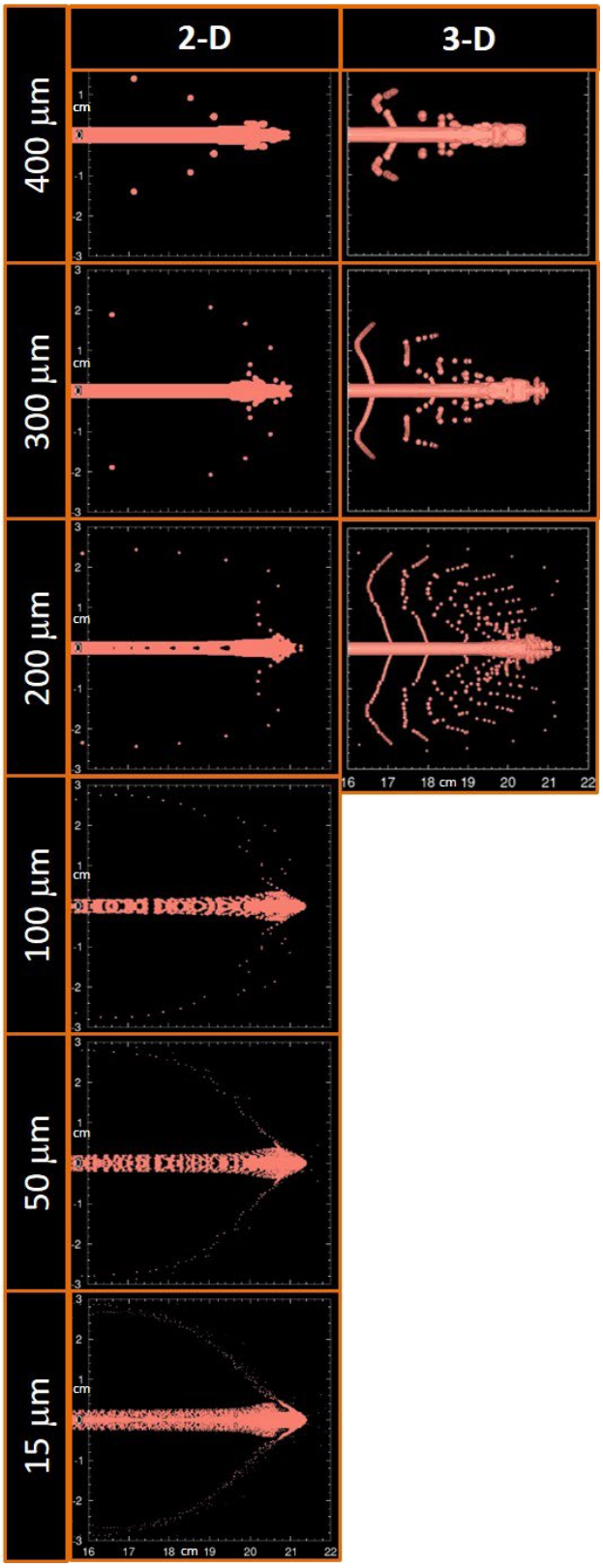

Fig. 18 Jet tip morphology at 40 µs in 2-D (left) and 3-D (right) simulations with resolution finer than 500 µm

The correspondence between 2-D and 3-D morphology is generally good across the full range of resolution. At the coarsest resolutions the tip has a simple bulb shape, which increasingly elongates at 800-μm and 700-μm resolution.

At 600 μm, detached material appears around the tip. Recall that apparent particles of material in 2-D actually represent solid axisymmetric rings. In contrast, 3-D simulations allow true particulation, although the characteristic lengths observed in particle shroud structures are greatly dependent on mesh resolution. The regular patterning seen in the shroud in the 3-D simulations 300-μm and 200-μm resolution are numerical artifacts arising from the regular discrete cubic mesh structure, and not representative of true physical processes.

Voids appear along the axis in 2-D simulations at resolutions of 200 μm and below. Evidence of voids can also be seen along the axis in cross-sectional images of the highest-resolution 3-D simulations (Fig. 19). These voids likely form from radial expansion of material as the jet exits the high-pressure region at the pressure head. As the resolution becomes increasingly fine in 2-D simulations, axial voids transition into a lattice-like morphology reminiscent of expanded metal structures. At 15-μm resolution the jet tip takes on a fibrous or fuzzy appearance.

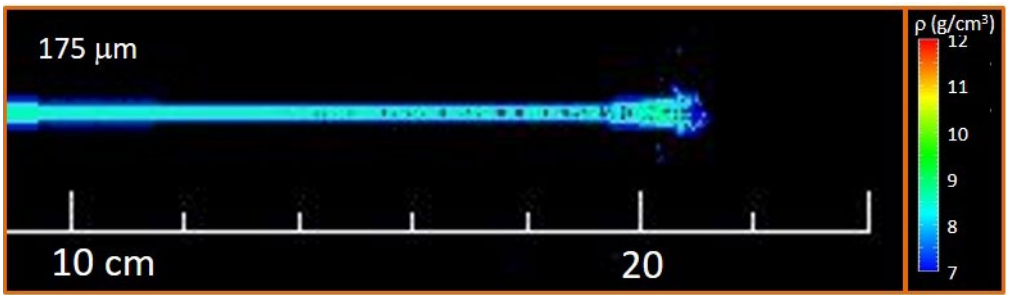

Fig. 19 Cross-section image showing copper density at 40 μs in a 3-D simulation with 175-μm resolution

These open morphologies do not reflect known jet structure and may arise from unexpected behavior in some computational algorithm at very high resolution. One candidate is the FORCE FRACTURE option in the VOID INSERTION material fracture model. This option initiates fracture whenever material density falls below a threshold value specified by the user, which is 7.114 g/cm^3 for copper in these simulations (80% of the ambient density). Repeating the ultra-high-resolution 2-D simulations with FORCE FRACTURE switched off should reveal if this is the case.

5. Conclusion

The aim of this study is to quantify the irreducible numerical uncertainty associated with resolution in ALEGRA simulations of SCJ formation. Estimates of the expected mean value and distribution widths for several jet properties as a function of resolution in 2-D and 3-D simulations are provided. Mass loss due to discretization errors are identified and measured. Resolution effects on the thermodynamic state of the pressure head are shown to affect the resulting temperature profiles of SCJs. A systemic study of jet tip morphology reveals the troublesome formation of axial voids at ultra-high resolutions.

Overall, the ALEGRA code demonstrates highly desirable behavior with decreasing resolution in these SCJ formation simulations, including generally monotonic behavior of jet properties with apparent limiting behavior, and convergence of outcomes between 2-D and 3-D simulations. Whether these behaviors carry over to output QoI in more complex scenarios, such as penetration of a jet into an RHA block, remains an open question.

Several follow-up lines of inquiry have been identified. The estimation method demonstrated in Section 4.4 can be tested by performing arrays of simulations at fixed resolutions while varying multiple computational parameters to generate a distribution of outcomes. This would indicate whether the method estimates are sufficient enough to be of use.

Regarding the underlying source of resolution dependence in the thermodynamic state of the pressure head, running a set of simulations at fixed spatial resolution and manually limiting the time steps to increasingly smaller values should reveal if the dependences shown in Fig. 12 are driven predominantly by temporal or spatial effects.

Finally, performing 2-D simulations at ultra-high resolution with FORCE FRACTURE turned off would show whether material fracture algorithms are behind the appearance of axial voids in the jet tip.

With this initial study of resolution effects in ALEGRA completed, the next logical step will be to repeat this work for the upcoming version 12.2 of the CTH code when it becomes available.

6. References

1. Hornbaker DJ. Quantifying uncertainty from computational factors in simulations of a model ballistic system. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2017 Aug. Report No.: ARL-TR-8074.

2. Hornbaker DJ. Quantifying uncertainty from computational factors in simulations of a shaped charge jet. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2019 Jan. Report No.: ARL-TR-8618.

3. Bova SW, Carleton JB, Granzow BN, Hansen GA, Love E, Luchini CB, Niederhaus JHJ, O'Brien CJ, Powell M, Robinson AC, et al. ALEGRA user manual. Albuquerque (NM): Sandia National Laboratories; 2020 Jan 21. Report No.: SAND2020-DRAFT (supersedes SAND2019-0784).

4. Schraml S. Simulation of shaped-charge jet formation and penetration using ALE3D. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2016 Aug. Report No.: ARL-TR-7744.

5. Fermen-Coker M. A 2-D axisymmetric model of a stretching jet in ALEGRA. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2004 Sep. Report No.: ARL-TR-3309.

6. Doney RL, Niederhaus JHJ, Fuller TJ, Coppinger MJ. Effects of EOS and constitutive models on simulating copper shaped charge jets in ALEGRA. Int J Impact Eng. 2020;136:103428.

7. Attaway SW, Hills RG, Giunta AA. CTH shaped charge simulations. Albuquerque (NM): Sandia National Laboratories; 2009 July. Report No.: SAND2009-0726.

8. Niederhaus JHJ, Uhlig WC. Analysis and validation for shaped charge jet simulations using ALEGRA. Albuquerque (NM): Sandia National Laboratories; 2011 Apr. Report No.: SAND2011-2819.

9. Hertel ES Jr, Kerley GI. CTH reference manual: the equation of state package. Albuquerque (NM): Sandia National Laboratories; 1998 Apr. Report No.: SAND98-0947.

10. Segletes SB. An analysis on the stability of the Mie-Grüneisen equation of state for describing the behavior of shock-loaded materials. Aberdeen Proving Ground (MD): Ballistic Research Laboratory (US); 1991 Mar. Report No.: BRL-TR-3214.

11. Crawford DA, Schmitt RG, Butler RJ. Spymaster user's guide, version 6.0. Albuquerque (NM): Sandia National Laboratories; 2016 Jun 23.

12. Schoof LA, Yarberry VR. EXODUS II: a finite element data model. Albuquerque (NM): Sandia National Laboratories; 1995 Nov. Report No.: SAND92-2137.

13. Sjaardema GD. APREPRO: an algebraic preprocessor for parameterizing finite element analyses. Albuquerque (NM): Sandia National Laboratories; 1992 Dec (updated 1999 Sep). Report No.: SAND92-2291.

14. Ayachit U. The ParaView guide, community edition. Clifton Park (NY): Kitware Inc.; 2019 Jan 21.

15. NIST/SEMATECH e-handbook of statistical methods. Gaithersburg (MD): National Institute of Standards and Technology; updated 2012 Apr. (Section 1.3.5.13, Runs Test for Detecting Non-randomness.) [accessed 2020 Apr 28]. http://www.itl.nist.gov/div898/handbook/

Appendix. Shaped-Charge Jet Device Model Liner Mass in the Continuum Limit

The copper liner of the shaped-charge jet (SCJ) device model is a cylindrically symmetric thin shell structure with exterior and interior profiles consisting of straight lines and circular arcs (Fig. A-1). The total material volume is determined by calculating the volume enclosed by the exterior profile and subtracting the volume enclosed by the interior profile. The simple geometry makes this is a textbook calculus exercise.

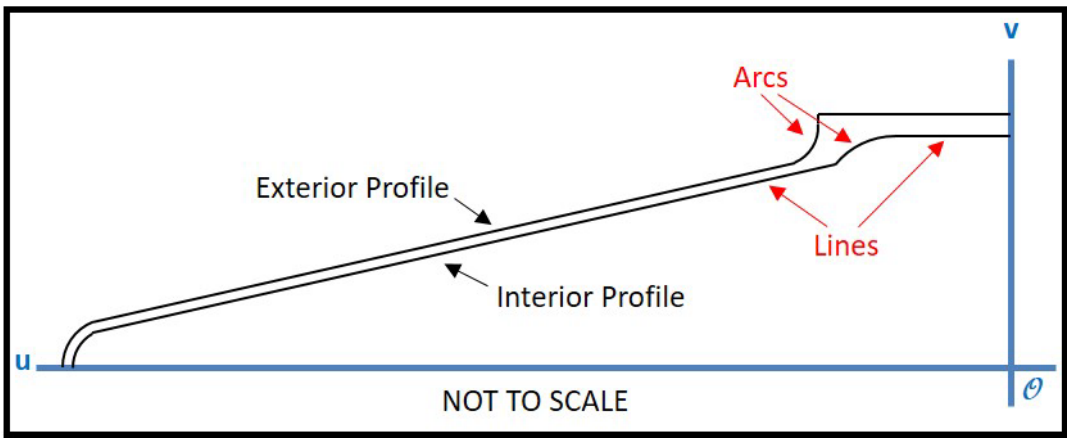

Fig. A-1 SCJ device model profile

The general equation for a line between two points (u_1, v_1) and (u_2, v_2) is

$$(v - v_1) = \left(\frac{v_2 - v_1}{u_2 - u_1}\right)(u - u_1). \tag{A-1}$$

When both v_1 and v_2 are nonnegative, the volume enclosed by revolving the line about the u-axis is

$$V_{line} = \int_{u_1}^{u_2} \pi v^2 \, du = \pi \int_{u_1}^{u_2} \left[\left(\frac{v_2 - v_1}{u_2 - u_1}\right)(u - u_1) + v_1\right]^2 du$$

$$= \frac{\pi}{3}\left(\frac{u_2 - u_1}{v_2 - v_1}\right)[v^3]_{u_1}^{u_2} = \frac{\pi}{3}\left(\frac{u_2 - u_1}{v_2 - v_1}\right)(v_2^3 - v_1^3). \tag{A-2}$$

This can be transformed into the more common form for the volume of a truncated cone by factoring the last term,

$$V_{line} = \frac{\pi}{3}\left(\frac{u_2 - u_1}{v_2 - v_1}\right)(v_2^3 - v_1^3) = \frac{\pi}{3}\left(\frac{u_2 - u_1}{v_2 - v_1}\right)(v_2 - v_1)(v_2^2 + v_2 v_1 + v_1^2)$$

$$= \frac{\pi}{3}(u_2 - u_1)(v_2^2 + v_2 v_1 + v_1^2). \tag{A-3}$$

A circular arc centered at (u_c, v_c) with radius r_c satisfies

$$r_c^2 = (v - v_c)^2 + (u - u_c)^2. \tag{A-4}$$

Arcs in the profile can extend from $u_c - r_c \le u_1 < u_2 \le u_c + r_c$, with either $v \le v_c$ or $v \ge v_c$. For arcs with $v \ge 0$, the volume enclosed by revolving about the u-axis is

$$V_{arc} = \int_{u_1}^{u_2} \pi v^2 \, du = \pi \int_{u_1}^{u_2} \left(v_c \pm \sqrt{r_c^2 - (u-u_c)^2}\right)^2 du$$
$$= \pi \int_{u_1}^{u_2} (v_c^2 + r_c^2) - (u-u_c)^2 \pm 2v_c\sqrt{r_c^2 - (u-u_c)^2}\, du, \quad \text{(A-5)}$$

where the ± sign in the integrand selects the upper or lower arc. The solution is

$$V_{arc} = \pi \left[(v_c^2 + r_c^2)u - \frac{(u-u_c)^3}{3} \pm v_c(u-u_c)\sqrt{r_c^2 - (u-u_c)^2} \right.$$
$$\left. \pm v_c r_c^2 \sin^{-1}\left(\frac{u-u_c}{r_c}\right)\right]_{u_1}^{u_2}. \quad \text{(A-6)}$$

As a check, for $v_c = 0$, $u_1 = u_c - r_c$ and $u_2 = u_c + r_c$, the enclosed volume is

$$V_{arc} = \pi \left[r_c^2 u - \frac{(u-u_c)^3}{3}\right]_{u_c+r_c}^{u_c+r_c}$$
$$= \pi \left\{\left[r_c^2(u_c + r_c) - \frac{(r_c)^3}{3}\right] - \left[r_c^2(u_c - r_c) - \frac{(-r_c)^3}{3}\right]\right\}$$
$$= \pi \left\{u_c r_c^2 + r_c^3 - \frac{r_c^3}{3} - \left[u_c r_c^2 - r_c^3 + \frac{r_c^3}{3}\right]\right\}$$
$$= \pi \left\{1 - \frac{1}{3} + 1 - \frac{1}{3}\right\} r_c^3 = \frac{4}{3}\pi r_c^3, \quad \text{(A-7)}$$

which is the correct result for the volume of a sphere.

Table A-1 shows the calculated exterior and interior volumes for each section of the SCJ device model. The overall net liner volume is 8.9006 cm^3. The copper material model has an initial density of 8.930 g/cm^3, giving an initial liner mass in the continuum limit of 79.482 g.

Table A-1 SCJ device model volume calculation

Section	Volume (cm^3)		
	Exterior	**Interior**	**Net**
Arc 1	0.2372	0.1414	...
Line 1	86.4333	80.8436	...
Arc 2	0.5184	1.3977	...
Line 2	9.8187	5.7244	...
TOTAL	97.0076	88.1070	**8.9006**

List of Symbols, Abbreviations, and Acronyms

2-D	two-dimensional
3-D	three-dimensional
DoP	depth of penetration
EOS	equation of state
KE	kinetic energy
PBS	portable batch system
QoI	quantity of interest
RHA	rolled homogeneous armor
SCJ	shaped-charge jet

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 CCDC ARL
(PDF) FCDD RLD CL
TECH LIB

1 CCDC ARL
(PDF) FCDD RLW PE
D HORNBAKER