Modeling Nonlinear and Dynamic Mechanical Behavior

Matthew T. Shanaman, Nicholas A. Vavalle, and Michael A. Lapera

ABSTRACT

Highly nonlinear and dynamic mechanical behavior involving impact, crash, and blast is common in some of the work done at the Johns Hopkins University Applied Physics Laboratory (APL). Modeling these behaviors involves finite element analysis (FEA) that reaches beyond typical static analyses. APL researchers are able to model complex nonlinear dynamic behavior without oversimplifying or converting the problem to a so-called equivalent static problem. Presented here is an overview of dynamics and nonlinearity and a brief summary of the options available for modeling these behaviors. The article concludes with several case studies that demonstrate how APL's expertise in this area contributes to the safety of our nation's warfighters and diplomatic personnel.

INTRODUCTION

Blast, impact, and ballistics are the legacies and the poster children of nonlinear dynamic finite element analysis (FEA). Sometimes termed *hydrocodes* because of their original uses for solving hydrodynamic problems[1] such as those researched in the Department of Energy labs, nonlinear dynamics FEA software programs, as well as the challenges in using them, are often associated with specialty users at national laboratories or in academia. In fact, researchers in APL's Research and Exploratory Development Department (REDD) and its progenitors have been tackling problems like these for years on projects that span the entire Laboratory. The field has earned a reputation for being difficult and expensive and requiring long lead times because of the resources required to set up, debug, and run the simulations. However, REDD researchers have combined advances in software and hardware with their extensive experience to open the door to incorporating nonlinear dynamic FEA into the normal design and evaluation process.

BACKGROUND

Dealing with Dynamics

FEA has become ingrained in engineering design, especially with the advent of graphical user interface (GUI)-driven model development in legacy FEA software packages like ANSYS, which streamlined analyses for the average user. Most engineers and analysts with experience in structural FEA are familiar with static problems, and possibly even problems with a small degree of nonlinearity. However, most of these same analysts are just as unfamiliar with dynamic FEA. In fact, many will seemingly go the extra mile to convert a dynamic problem into a static one, or come up with static-equivalent loads to avoid a dynamic problem. (Refer to Box 1 for more detail on statics, dynamics, linearity, and nonlinearity.)

Strictly speaking, all loads could be considered "dynamic" in the sense that any load on a structure must have been applied over time. All structural FEM codes are built from the same set of physical conservation equations: mass, momentum, and energy. The momentum relation ends up being the equation on which much of the code pivots since it controls the movement of the nodes, and it is here that the distinction between a static and a dynamic problem is made. The momentum balance in its typical forms can be manipulated and finally expressed for finite elements in a simple balance of forces as functions of displacement and time[2]:

$$Ma + f^{int}(d,t) = f^{ext}(d,t)$$

inertial forces + internal forces = external forces.

It is an intuitive concept: when a force acts on a body, it is going to manifest as both movement and deformation of the body. Deformation results in stresses, or internal force built up inside the body. Viewing structural problems in the light of conservation of momentum is important because it forces the mindful analyst to observe that static and dynamic problems are not two separate entities. Instead, a static problem is actually a special case in which an analyst has decided that inertial effects are so small that both time and mass can be neglected from the system for the sake of a simpler problem.

Many engineers try to avoid dynamic problems because highly dynamic situations get complicated and highly nonlinear very quickly. With this greater complexity comes a need for more detailed information, a tendency toward longer model solve times, and a higher likelihood that the model might fail to solve or that it might solve incorrectly. The best remedy to these issues is not to avoid the difficulties, but to leverage specialized and experienced modelers along with advanced software and hardware. Fortunately, REDD is able to supply all these assets.

Under the Hood of Dynamic FEA Software

Dynamic FEA software uses one of two methods to integrate differential equations in time: explicit and implicit. Explicit methods are a class of numerical integration algorithms that drive the solution forward in time by using information known in the present to calculate future values. Using the classic forward Euler method as an example,[3] position in the future is calculated by using the velocity known in the present:

$$x_{future} = x_{present} + \Delta t v_{present}.$$

The advantages of explicit methods are that they are computationally easy to solve and do not require iteration and convergence because they use values that are already known to step forward. There is also no need to build and then invert a large stiffness matrix, as is the case with static and implicit methods. The major disadvantage of explicit methods is that they are termed *conditionally stable*. This means that the solution can remain stable only if the time step is smaller than a critical value, known as the Courant–Friedrichs–Lewy condition.[2] For first-order structural finite elements, this critical time step is the characteristic length of the smallest element divided by the sound speed of the material within that element. If a single element of steel with a sound speed of 3,200 m/s is 1 mm across, then the maximum time step would be 0.3125 µs, requiring 3,200 time steps to solve 1 ms of simulation time!

Explicit methods trade time step size for ease of computation and are therefore best suited to events happening at small timescales on the order of a few milliseconds or fewer (crash, impact, blast). This means that explicit methods are usually confined to problems in which stress wave propagation is important and the area of interest is local to the applied loading. Additionally, since the solver does not require iteration and convergence, explicit methods are the best solution for solving problems that change suddenly and are highly nonlinear.

Implicit methods differ from explicit ones in that they drive the solution forward in time by using information from the future. This is exemplified by the classic backward Euler method,[3] in which position in the future is calculated using the velocity from the future also:

$$x_{future} = x_{present} + \Delta t v_{future}.$$

BOX 1. STATICS, DYNAMICS, LINEARITY, AND NONLINEARITY

In the field of continuum mechanics, a static problem is one in which the system does not change over time. A dynamic problem is therefore one in which the system does change over time. The broader implication is that in static finite element method (FEM) problems, the concept of time is neglected, along with mass, since inertial forces exist only in a system in which an object is moving.

Linearity and nonlinearity, in the context of FEM, refer to the mathematical relationship between the forces and the displacements in the system. A linear mechanics problem's output will scale proportionally with its input. Analysts and programmers care about this distinction mostly because linear problems are very easy for computers to solve, while problems with increasing nonlinearity generally require iterative calculations and special algorithms to solve, which increases the computer resources, the time, and the detail required for a correct solution, as well as the chances that the solver will fail to find a solution.

The glaring difference between this and the forward Euler method is that there are unknown terms from the future on both sides of the equation, which at the least requires a system of equations to solve (a matrix equation) and, because it is nonlinear, will require an iterative solution to solve. In FEA, this requires a large stiffness matrix to be built (a square matrix the size of the number of degrees of freedom in the system) and then factored to invert the matrix and solve for forces. In an iterative solution, this has to be done many times.

The consequence of the implicit solver is that it is more computationally expensive to compile and invert a large stiffness matrix and iterate toward a solution. This consequence can manifest itself in the form of long solving time, but it may also have repercussions derived from hardware or software license limitations. Large matrices require a lot of computer memory. If a user does not have sufficient computational resources or their available software licensing does not support the additional computational resources required, they may be restricted in terms of the size of the model they are able to run. The huge advantage, however, is that an implicit solver is termed *unconditionally stable*, meaning that the solution remains *mathematically* stable for virtually any time step size. However, exceedingly large time steps could skip over a potentially significant dynamic event in time or change the behavior of a material that is path dependent, so it is important to remain cognizant of the time step size, even if it is unconditionally stable. If an iterative solver has trouble converging on a solution, usually because of high nonlinearity, the best course of action is usually to take smaller time steps. In some cases, the required time steps for convergence can become so small that the value of the implicit solver is completely lost.

Implicit solvers trade ease of computation for a larger time step size and are therefore best suited to problems with moderate nonlinearity, moderate size, and relatively long simulation time on the order of tens of milliseconds or more. Problems that do not need to capture wave propagation, but rather the overall dynamic structural response of a system, fall into this category.

Another dividing line in dynamic FEA implementation is reference frame. What most imagine as classic FEA is from the Lagrangian perspective: element boundaries and nodes describe the deformation of a material. This is a very natural way to describe solid materials and their deformation. Unfortunately, under high deformation, elements can become overly stretched or even tangled; if this deformation does not crash the simulation, it both reduces the accuracy of the elements and drives the time step in explicit simulations to become smaller as the distance across some dimensions of the stretched elements becomes very small. Popular Lagrangian finite element codes include most of the industry-recognized names, such as ANSYS, NASTRAN, Abaqus, LS-DYNA, COMSOL, and many others.

The other perspective from which finite elements can be viewed is Eulerian. In this finite element scheme, a grid of elements and nodes is fixed in space while material moves through the fixed grid. This is a very natural way to describe fluid materials and how they flow. This description is convenient in terms of large deformation because the mesh cannot severely distort or tangle. The primary disadvantage is that Eulerian methods often require a very large number of elements since they must be defined at any point in which material might flow; it is also more difficult to visualize hard boundary lines between disparate materials or empty space. Another difficulty is that material history variables must be advected, or passed between elements, rather than being fully contained within an individual element over time, and this presents its own numerical challenges.[2] An example illustrating the differences between the Lagrangian and Eulerian modeling techniques is found in Figure 1, where the high-speed testing technique known as Taylor bar impact testing is simulated using both methods.

In an effort to combine the best of both worlds, a method called Arbitrary Lagrangian Eulerian (ALE) was developed.[4] An ALE solver allows a user-defined mix of Lagrangian and Eulerian descriptions. This method takes the form of a multistep process where a full

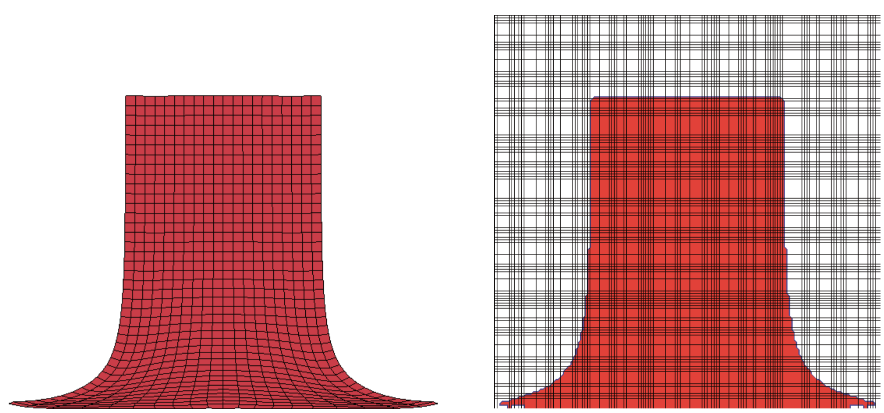


Figure 1. Comparison of Lagrangian and Eulerian methods for a 200-m/s Taylor impact test of a copper cylinder. The Lagrangian method (left) shows a clear deformed shape at the element boundaries and requires relatively few elements but displays severe element distortion in the highly deformed region, degrading accuracy and causing eventual stability issues. The Eulerian method (right) maintains a fixed element grid and thus maintains computational stability through extreme deformation, but at the cost of an unclear material boundary using over 50 times more elements that are 4 times smaller than their Lagrangian counterparts.

Lagrangian step is taken and then followed by a mesh remapping step where the mesh is pushed back an arbitrary amount toward its initial position, and material is appropriately advected through the mesh.[2] The user can go so far as to use an ALE code as purely Lagrangian or purely Eulerian if they wish.

Relevant to the discussion of dynamic FEA are the so-called meshless methods, in which elements are avoided completely in favor of particles so that large deformation or even fluid flow is possible in the Lagrangian description. An example often used at APL is smoothed particle hydrodynamics (SPH), in which particles interact with each other via kernel functions that describe continuum mechanics in similar ways to classic finite elements.[5] SPH is well suited to extremely fast problems involving very high deformation, like those involving explosives and hypervelocity impact. The advantage is a model capable of achieving massive deformation while maintaining a Lagrangian reference frame. However, a common disadvantage, similar to Eulerian methods, is that boundaries and continuity are more difficult to observe because of the particle representation. Advances in SPH post-processing have helped to relieve this problem.

BOX 2. CURRENT SOFTWARE CAPABILITIES USED IN APL'S REDD

With all of the different types of solvers, length scales, timescales, and problem sizes, it isn't surprising that a single software package would be inadequate to handle all of them. For this reason, REDD employs an entire suite of software applications whose combined capabilities allow researchers to tackle a broad range of problems. The following is a list of the most commonly used software packages in REDD:

- ANSYS LS-DYNA: A finite element solver whose progenitor, DYNA3D, was created at Lawrence Livermore National Laboratory (LLNL) in 1976. The software is capable of both explicit and implicit nonlinear dynamics and has Lagrangian, ALE, and SPH modules. It is most widely associated with Lagrangian nonlinear explicit dynamics simulations.
- IMPETUS: A finite element solver with similar capabilities to LS-DYNA, but without the ALE or implicit capabilities. Its major difference from LS-DYNA is that it was written to be run on graphics processing units (GPUs) rather than standard central processing unit (CPU) machines. Its GPU capabilities allow for very large models and extreme deformation of Lagrangian elements without computational failure.
- ALE3D: LLNL's modern ALE finite element solver available to the Department of Defense and associated contractors for work related to national defense.[6] Its main use is as an explicit hydrodynamics code, but it also has multiphysics, implicit, and SPH capabilities as well.
- Abaqus FEA: A finite element solver capable of both explicit and implicit nonlinear dynamics with Lagrangian, ALE, and SPH modules. The software has a long legacy reaching back to 1978, is very well known, and is most widely associated with Lagrangian nonlinear implicit simulations.

CASE STUDIES

Simulation of the Warrior Injury Assessment Manikin (WIAMan)

Improvised explosive devices (IEDs) quickly became a problem in the Iraq and Afghanistan wars of the 2000s when they introduced a new loading mechanism that vehicles had to protect against—underbody blast (UBB). To better protect the warfighter in such scenarios, the Army used the Hybrid-III crash test dummy to understand the human response to the UBB loading scenario. Because the Hybrid-III was designed by the automotive industry for frontal crash testing, it was discovered to be an inadequate surrogate for the human in UBB conditions. Its physical response did not match that of a human, and it was not sufficiently durable in UBB conditions. The Army decided to develop a crash test dummy specifically designed for UBB, called the Warrior Injury Assessment Manikin (WIAMan). The WIAMan needed to withstand the severe loading of UBB while still exhibiting the physical response of a human. On top of this challenge, the Army needed a solution fast, but crash test dummies are often developed over the course of decades with necessary refinement to their predictions of human injury. Since the Army did not have decades to develop the WIAMan, dynamic FEA was used to help speed up the design process.

With nearly a decade of expertise in both biomechanics and LS-DYNA modeling, REDD's Biomechanics and Injury Mitigation Systems (BIMS) program modeling team developed an LS-DYNA model of the manikin (the WIAMan FEM) and validated it against the response of the physical test device.[7,8] This development effort was performed in conjunction with government engineers from the WIAMan Project Office, the Army Research Laboratory, and the Combat Capabilities Development Command (DEVCOM) Analysis Center (DAC). The WIAMan FEM team had particular expertise in validating large-scale dynamic models of this nature. Further, the team was composed of a mix of mechanical engineers and biomechanical engineers who brought unique perspectives from both the mechanical design side and the human response relationship side. For this kind of modeling, software solutions must offer a number of capabilities: a vast library of nonlinear material models, the ability to simulate short-duration events (~100 ms) using explicit FEA, the ability to solve very large models, and a long history of use for blast simulation. Inherent in blast simulations is the need for very large deformations and motion over a short period of time, which requires that stress wave propagation is tracked appropriately. This, combined with hundreds of parts coming into contact and the use of advanced material models for both polymers and metals, demands the use of an explicit dynamic nonlinear FEA solver.

Model development was no small undertaking: on the order of 1,000 parts had to be modeled for the WIAMan (Figure 2). The team took a hierarchical approach to validation, where material characterization fed into component and subsystem models that were ultimately assembled into the whole-body model. Subsystem models were validated against test data in addition to the full system-level validation of the whole-body model. The current whole-body model typically runs on a high-performance computing cluster on 60–100 CPU cores and can simulate 50 ms in roughly 6 h.

The validated model was used in several ways to enhance the physical test device and the understanding of human injury in UBB. A large-scale design-of-experiments study was conducted using about 50 simulations to help the WIAMan designers understand which design aspects would have the greatest effect on the biofidelity (WIAMan's ability to respond like a human). This study revealed that while some design parameters yielded minor improvements to the biofidelity scores in close proximity to the changed parameter, they were often at the detriment of other scores, with a net-zero gain. The WIAMan test device is now in the production phase, but the WIAMan FEM will live on as a long-term complement to the physical test device. While the physical test device was being outfitted with injury prediction capabilities through the research of a collection of universities and APL, the WIAMan FEM team was concurrently developing the same capabilities for the model. This will give Army engineers an additional tool for predicting human injury in vertical loading scenarios to complement expensive blast tests. Ultimately, the WIAMan FEM affords two benefits: it is a design tool for rapidly developing a physical device and is a stand-alone injury prediction capability for the Army. The WIAMan FEM is owned and managed by DAC.

Simulation of the Swaging Process on an Elastomer Hose

For another project, the design team requested simulation to help them design a metal collar to be swaged on to the end of an elastomer (rubber-like) hose. The swaging process involved drawing a mandrel through the inner collar part while holding the outer collar part fixed with the rubber hose sandwiched between the two parts. The drawn mandrel then forced the inner collar part outward, plastically deforming the metal and clamping the rubber tightly between the inner and outer collars.

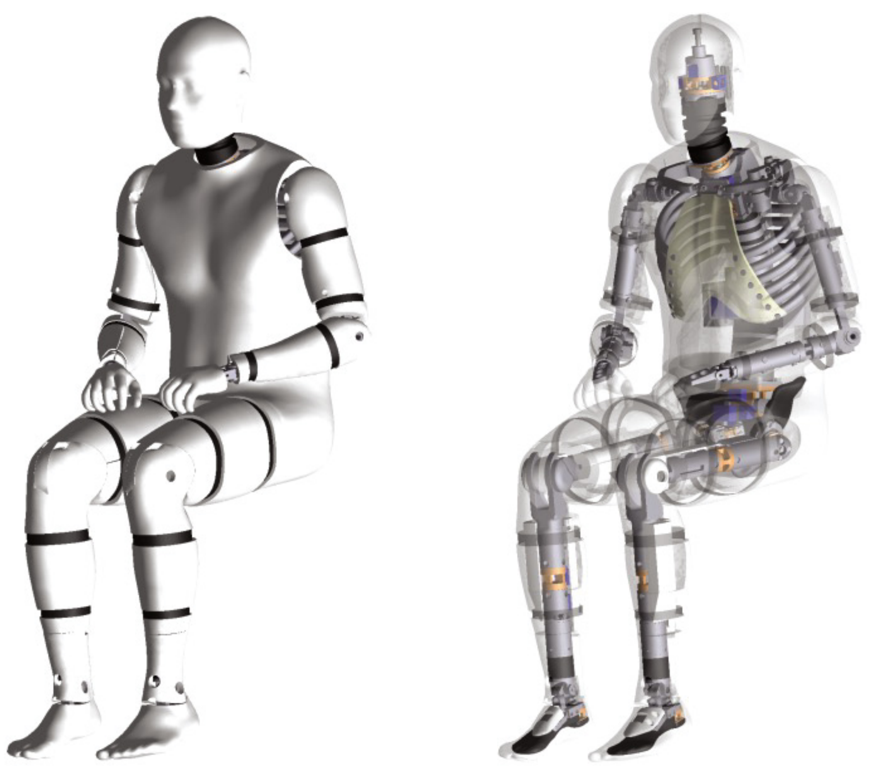


Figure 2. The whole-body WIAMan FEM. The model, with flesh (left) and interior parts exposed (right), was developed using LS-DYNA. It can simulate the response of the physical test device and predict the likelihood of human injury in the dynamic UBB environment.

Modeling this process required advanced material models for both the titanium metal collar and the rubber tubing. Because the metal was plastically deformed, the material model had to capture this phenomenon accurately. Titanium, a commonly used structural metal, is just as commonly used in FEA and is well characterized in the literature, so the team easily found and applied a Johnson–Cook plasticity material model for the swaging simulations.

The hose underwent very rapid, large deformation during the swaging process. Although elastomers have a reputation for springing back to shape, they generally have varying degrees of nonlinear, plastic, and time-dependent properties, so proper material model selection and intelligently designed material testing is required in order to calibrate the material models.[1] In addition to the material modeling, complexity is added to the simulation since the swaging process and the nature of the parts involved require multiple time scales, small mesh sizes, and accurate contact interfaces between the parts. This complexity created a difficult choice between explicit and implicit methods: the nonlinearity of the problem due to the materials and contacts made it a good candidate for explicit methods, while the mesh size and timescales on the order of seconds made the problem a good candidate for implicit methods. The team used Abaqus, which offers a robust library of material models for elastomers and the ability to use both explicit and implicit methods, to carry out the simulation.

The first step was to test the elastomer materials and calibrate them to a representative material model. The elastomer material needed to respond correctly to large deformations at a relatively high rate, and then relax appropriately after the swaging process was finished. This behavior was captured with two tests: a standard ASTM D395 compression set test[9] and a customized cyclic tension test with stress relaxation holds.[10] The compression set test, sometimes called "permanent set," compresses rubber samples to a defined strain and holds them for a long period of time before releasing them. The residual strain in the compression samples is then measured after samples have rested for an appropriate period of time. The custom cyclic tension test consisted of five tensile iterations in which the sample was pulled to a finite strain, held at this strain for 10 s to allow observation of stress relaxation, and then released back to zero strain. With each new cycle, the maximum strain was increased from an initial 5% strain to a final 60% strain to capture nonlinear viscous effects in the stress relaxation. With these test data in hand, a polymer material fitting software was used to calibrate an advanced polymer material model in Abaqus, called the Parallel Rheological Framework (PRF) model, using three viscoplastic networks and Mullins damage to achieve a nearly perfect fit to the experimental data (Figure 3).

Along with the PRF models built for the rubber layers, orthotropic material properties were introduced and assigned to "skins" (shell layers) within the rubber layers to capture the additional radial stiffness of the hose imparted by the layers of wrapped nylon cord reinforcement. These properties were calculated by applying a series of stiffness matrix transformations to test and published data to appropriately capture the micromechanics of this specific material system. These material properties were assigned to their respective rubber hose and reinforcement and applied to an axisymmetric geometric representation of the configuration-built Abaqus CAE. A rigid surface representing the mandrel was aligned axially and prescribed a constant velocity so that it made contact with the inner swage fitting surface. The mandrel caused the inner swage fitting to plastically deform and crimp the inner hose material.

The team used the Abaqus explicit solver to solve the mechanics of the mandrel swaging process, and then imported results into the Abaqus implicit solver to investigate the coupling's springback behavior. A springback analysis (implicit) allows for efficient assessment of the effects of stored elastic energy on a part that has been plastically deformed, providing an "equilibrium" state for the swage fitting so that its final deformed dimensions can be realized. Once the amount of springback was determined, tie constraints were applied to the inner surface of the coupling and outer surface of the hose, thereby fixing the hose within the coupling. This approximated the epoxy that is injected into the coupling fitting before the fitting is inserted and the hose is swaged. Once these constraints had been applied, a vertical load was applied to the top of the swage fitting to assess the stresses and strains within the various hose materials as part of a preliminary analysis of the pull test qualification required for the final system. By combining these Abaqus analyses and the test data generated from APL's tear test experiments, the team investigated several criteria as candidates for predicting the onset of material failure.

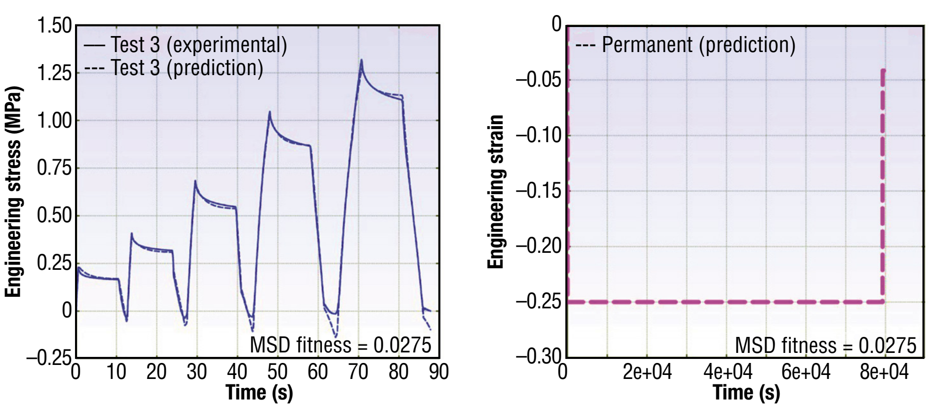


Figure 3. Swaging simulation test data. Abaqus PRF material model calibration to cyclic tensile test data (left) and permanent set data (right). The permanent set graph does not show experimental data, but the predicted permanent strain value of 4% matches the experimentally measured value.

Modeling contact interactions between very stiff and very compliant materials is one of the toughest challenges encountered in FEM. Each solver's design of its elements is proprietary, and various material model options are available for use in the modeling of hyperelastic materials, so element failures and errors are common. In the case of this analysis, the Abaqus explicit solver offered robust axisymmetric element formulations and a built-in ability to export explicit solutions to the implicit solver for a restart analysis, which facilitated very efficient sequential, multistep analyses where use of the explicit solver was not necessarily required.

Simulation of Bullet Impact on Armor

For another project, a REDD team was asked to provide whatever information it could on the rear face of personal body armor after it was struck by a rifle bullet—and the team was asked to do it on a short schedule of just 40 hours. The colloquial term for this phenomenon is *backface deformation* (BFD) of armor. Understanding BFD is important because, while armor may protect wearers from some ballistic threats penetrating into their bodies, the large energy of the impact is still a problem when it transfers through the armor and into the wearer, causing injury. The goal of the project was to develop a new system for measuring BFD, including selecting a material that could rest against an unbacked armor system and provide an accurate and consistent visual representation of the BFD. Because of the short schedule and the need for some very advanced modeling, the solver had to offer a fast and easy modeling process.

The armor system consists of a hard armor plate, which is represented in the model as a layer of ceramic armor in front of several layers of a thermoplastic composite (Figure 4). In practice, several layers of Kevlar are the next line of defense behind the armor plate, and these layers are also represented in the model. Finally, the deformation measurement material, which was originally specified to be a rubber material, backs the armor system. A rifle bullet traveling at several hundred meters per second impacts the hard armor plate and transfers its energy through the system, resulting in deformation, which is tracked and measured in the model.

The bullet is modeled using IMPETUS's advanced SPH package to capture its extreme deformation. The materials included in the bullet are Johnson–Cook plasticity models of lead and brass that are referenced to experimental work. The ceramic layer is represented by SPH and a Johnson–Holmquist material model lifted from the literature, specifically intended to characterize ceramic material strength and failure under impact loads. The thermoplastic is represented in several bonded layers by another tuned model for Dyneema, an ultra-high-molecular-weight polyethylene composite that is often used in armor panels.[11] The Kevlar is modeled with layers of an orthotropic fabric material model with typical stiffness and strength of Kevlar KM2 fabric. The initial trial material model for the backing material was a Bergström–Boyce elastomer model of moderately soft rubber that had been tuned in a previous project for air cannon impacts. High-order elements were used to ensure good deformation and good performance for high-aspect-ratio elements.

For partial validation, since full experimental data were not available in the time allowed for the simulations, the team was able to confirm that the simulated hard armor BFD depth matched average observations of APL-conducted experiments on unbacked hard armor plates. The simulated ceramic layer is fully penetrated by the bullet and stopped by the composite layer, which is also corroborated by the experiments.

Initial simulations that included the elastomer backing layer showed that it separated from the Kevlar backing layer and deformed to a much greater depth, very local to the bullet impact area. Because the intent was to observe the backing layer as a representative of armor BFD, extremely different deformation characteristics were not preferred. The behavior of the simulated rubber is reasonable, given that its material properties are drastically different from both the hard armor and the Kevlar layer. Since it is so soft compared with both the Kevlar and the hard armor, stress waves cannot travel nearly as quickly in the soft material, which forces material local to the impact to absorb and respond to all the impact energy, rather than spread it out further into more mass as stiffer materials tend to do. When testing this observation, the team was able to show that a theoretical soft material reinforced with directional strands of a stiffer material (which is the same idea as a tire using nylon fibers to reinforce the rubber directionally) would respond in a way that was more suitable to the designers' goals.

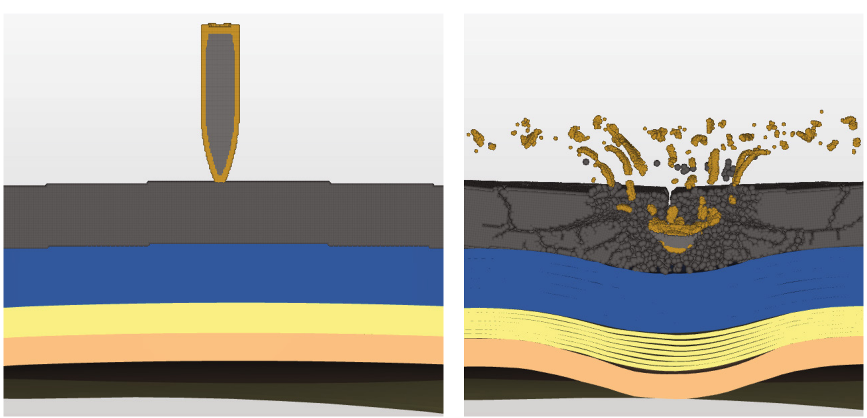


Figure 4. Simulation of a bullet strike on a layered armor system. The modeled system consists of (from top to bottom) ceramic, thermoplastic, Kevlar, and an elastomer backing material. The bullet and ceramic are modeled using the SPH method to capture the extreme deformation of the bullet, as well as the cracking and crushing mechanisms of the ceramic failure.

This quick evaluation of the impact event steered the experimental team toward a few subsets of material choices that would help them accomplish their ultimate goals. This also helped to guide the design of the clamping fixture that secured the backing material to the armor system. If the designers had not consulted the modeling team, a costly series of design iterations involving prototyping and experiment would have been required. This is a prime example of how physics simulation can drastically shorten the design process, even without a full suite of materials testing and validation efforts. While extra tests are certainly preferred or required in many cases where simulated response needs to be exact, when guidance and an understanding of how a system is likely to respond are needed, models such as the one presented here are still highly valuable.

Simulation of Shaped Charge Jet Generation and Fragmentation

Several APL programs are exploring the effects of shaped charges and warhead fragmentation, but much of this work cannot be discussed in open forums. In the interest of presenting an example of this work, a generic shaped charge warhead, with no link to current APL programs, is discussed.

Shaped charge jet formation and warhead fragmentation occur on the microsecond timescale. This timescale, along with the massive deformation of the materials, tends to place these phenomena fully in the camp of explicit FEA methods. In our experience, ALE and SPH methods accurately simulate explosive detonation, metal fragmentation and deformation, and ceramic cracking for a variety of problems.

The case study presented here and shown in Figure 5 is of a basic shaped charge jet penetrator that uses a waveshaper to reduce its footprint. A basic shaped charge is a high explosive shaped to direct and concentrate its energy in a particular direction. Colloquially in the discussion of warheads, the term *shaped charge* generally indicates an explosive whose detonation collapses a liner metal into an extremely high-velocity jet that is used as an armor penetrator. Rather than relying on the strength or hardness of a penetrator to defeat armor, shaped charge jets rely almost purely on kinetic energy and momentum driven by extremely high velocities.[12] The basic shaped charge can be described as a cylinder of high explosive into which a cone of liner material is pressed. When the charge is ignited at the opposite side from the liner, the detonation wave advances through the explosive toward the liner, impacting the tip of the cone first and gradually enveloping the rest of the liner. The explosive pressures cause the liner to collapse on its symmetric axis, resulting in the liner accelerating and stretching into a very fast and thin jet.[11,12]

Almost any material or geometric aspect of the full charge can affect the resulting jet's performance and shape. One way to increase jet performance while using less explosive is to change how the detonation wave reaches the liner via a waveshaper.[13] A waveshaper is an inert material that is placed between the ignition point of the explosive and the liner such that the detonation wave must travel around or through this material to detonate the explosive material on the other side. This shapes the detonation wave before it reaches the liner. In the example described here, a syntactic foam disk is embedded in the explosive charge such that the detonation wave is impeded from advancing directly into the liner tip, and instead must travel around the outside of the waveshaper, creating a shock wave that converges on the tip from the outside of the cylinder to the central axis. The model uses a polymer-bonded explosive, modeled with a tuned Jones–Wilkins–Lee explosive equation of state (EOS), to accelerate a metal liner modeled with a Steinberg–Guinan material strength law and Mie–Grüneisen EOS specifically tuned for very high velocities, temperatures, and pressures. The waveshaper is modeled with pore-compaction strength law and EOS. For completeness, the warhead liner jet is shown penetrating steel.

Placing a case around the sides of the explosive can also affect the jet, but it can also serve as additional fragments for the explosive. By applying appropriate material strength, EOS, and damage laws, the fragmentation characteristics of the case can also be modeled and tracked in the software. In the case of this model,

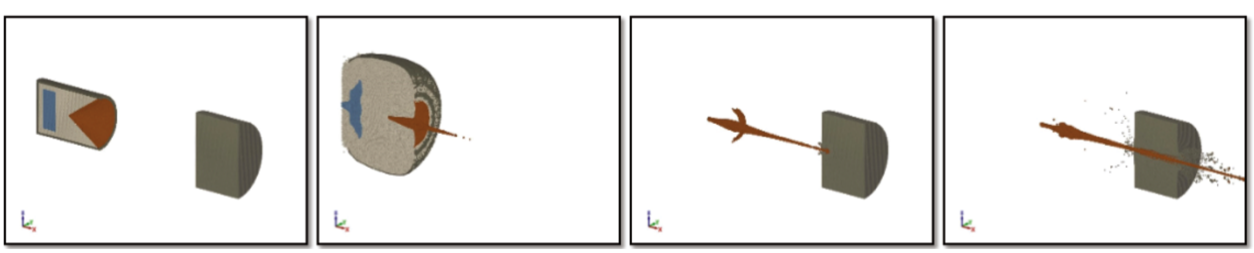


Figure 5. Simulation of a generic shaped charge generating a jet to penetrate steel. The detonation wave begins at the rear of the warhead, traveling through the explosive (white) and around the waveshaper (blue). As the explosive detonates, it converts to rapidly expanding detonation products, both fragmenting the metal body and forming the liner into a jet. The jet strikes and easily penetrates the cylinder of steel because of its extremely concentrated kinetic energy. The simulation is built for efficiency, turning parts on and off as needed throughout the simulation.

Johnson–Cook strength and damage laws are combined with a Mie–Grüneisen EOS to appropriately fracture the steel material into fragments. A built-in fragment tracker catalogs unique free bodies of all nonexplosive parts over the course of the simulation such that individual fragments can be identified and even exported with information, including their material, mass, velocity, size, and spin properties. This information is easily converted to standard ZDATA fragment files or paired directly with in-house codes such as the Ray-tracing Endgame Computational Tool (RECT) for future fragment dispersal and lethality assessments.

APL's ability to quickly analyze the detonation process, the effects of the explosion on both near and far-field material, and the fragmentation of materials—all while avoiding an extremely large Eulerian element grid or tangled Lagrangian elements—has significantly boosted its ability to support multiple programs. Future work in this field could contribute to warfighter injury mitigation, explosive ordnance disposal, future conventional weapons advancement, novel armor concepts, and sympathetic detonation assessment.

CONCLUSION

For over a decade, REDD staff members have been amassing experience and making critical contributions to projects across APL using nonlinear dynamic FEA. Along with the experience, software capabilities have also grown, both in product updates to existing solvers and the advent of new products. All of this, combined with powerful computing clusters, gives REDD a robust and advanced set of tools to solve difficult problems across the entire Lab. By solving nonlinear and dynamic problems in their natural state, instead of resorting to the old practice of oversimplifying or converting from dynamics to statics, we can significantly reduce over-engineered products, solve harder problems more accurately, and make critical contributions to our national defense.

REFERENCES

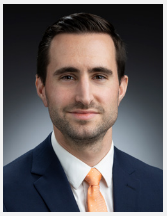
[1]D. J. Benson, "Computational methods in Lagrangian and Eulerian hydrocodes," University of California, San Diego, 1990.
[2]T. Belytschko, W. K. Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*, Chichester: Wiley, 2006.
[3]J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Chichester: Wiley, 2016.
[4]C. W. Hirt, A. A. Amsden, and J. L. Cook, "An arbitrary Lagrangian-Eulerian computing method for all flow speeds," *J. Comput. Phys.*, vol. 14, no. 3, pp. 227–253, 1974.
[5]R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: theory and application to non-spherical stars," *Monthly Notices Roy. Astron. Soc.*, vol. 181, pp. 375–389, 1977.
[6]C. Noble, A. Anderson, N. Barton, J. Bramwell, A. Capps, et al., "ALE3D: An arbitrary Lagrangian-Eulerian multi-physics code." LLNL-TR-732040. Livermore, CA: Lawrence Livermore National Laboratory, 2017. https://doi.org/10.2172/1361589.
[7]M. P. Boyle, A. M. Lennon, N. A. Vavalle, M. T. Shanaman, C. W. Lomicka, and C. O. Pyles, "Optimizing the biofidelity of the Warrior Injury Assessment Manikin through design of experiments," in *15th Int. LS-DYNA Users Conf.*, 2018.
[8]A. M. Baker, Z. Hostetler, N. A. Vavalle, R. S. Armiger, R. Coates, and F. S. Gayzik, "Hierarchical Validation prevents over-fitting of the neck material model for an anthropomorphic test device used in underbody blast scenarios," *J. Biomech. Eng.*, vol. 143, no. 1, pp. 1–11, 2021.
[9]ASTM International, "Standard test methods for rubber property – compression set," West Conshohocken: ASTM International, 2018.
[10]J. Bergstrom, "Smart mechanical testing of polymers," PolymerFEM, May 30, 2020. polymerfem.com/smart-mechanical-testing-of-polymers/.
[11]W. Walters, "Introduction to shaped charges," Aberdeen, MD: Army Research Laboratory, 2007.
[12]C. A. Weickert, "Demolitions," in *Explosive Effects and Applications*, J. A. Zukas and W. P. Walters, Eds., New York: Springer, 1997, pp. 381–420.
[13]H.-g. Guo, Y.-f. Zheng, L. Tang, Q.-b. Yu, C. Ge, and H.-f. Wang, "Effect of wave shaper on reactive materials jet formation and its penetration performance," *Defence Technol.*, vol. 15, no. 4, pp. 495–505, 2019.

Matthew T. Shanaman, Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Matthew T. Shanaman is a senior engineer and analyst in APL's Research and Exploratory Development Department. He has a BS in mechanical engineering from Drexel University and a master of mechanical engineering from Johns Hopkins University. Matt is an expert in nonlinear explicit dynamic finite element modeling (e.g., blast, impact, penetration) and LS-DYNA and IMPETUS dynamic finite element software packages. He has extensive experience in computational modeling and analysis of dynamic mechanical systems in the fields of biomechanics, aerospace, personal protection, additive manufacturing, topology optimization, ballistic penetration, and blast modeling. His email address is matthew.shanaman@jhuapl.edu.

Nicholas A. Vavalle, Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Nicholas A. Vavalle is a project manager in APL's Research and Exploratory Development Department. He has a BS in biomedical engineering from the University of Rochester and an MS and a PhD in biomedical engineering from Wake Forest University. Nicholas has expertise in computational models for predicting human injury, including the development and validation of such models. He led development of models that have been used to predict and prevent injury in underbody blast, blast overpressure, and nonlethal weapon loading, with particular interest in fracture and head injury. His past experience was in applying models to understanding and preventing injury in car crashes. He is the lead or contributing author on more than 40 conference and journal papers related to injury biomechanics, the development of human computational models, and validation techniques for human models. His email address is nicholas.vavalle@jhuapl.edu.



Michael A. Lapera, Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Michael A. Lapera is a computational engineer in APL's Research and Exploratory Development Department. He has a BS in mechanical engineering from Virginia Polytechnic Institute and State University (Virginia Tech) and an MS in mechanical engineering from the University of Rhode Island. Mike has experience in implicit and explicit finite element analysis (using Abaqus, LS-DYNA, and IMPETUS); materials characterization; micromechanics; microelectronics packaging thermal and structural analysis; vibration analysis; application of machine learning to mechanical engineering systems; diffusion modeling; viscoplasticity; composite material system design and analysis; and hydraulic control system design. His email address is michael.lapera@jhuapl.edu.