Modeling an Explosively Driven Fluid

New Mexico

Supercomputing Challenge

Final Report

Team Number 8

Albuquerque High School

Team Members

Carson Kent

Teacher

Joan Newsom

Executive Summary

One of the biggest threats American citizens and service members face abroad is attack via an improvised explosive device (IED). While, American soldiers have recently developed better methods and protocols for detecting these devices before they can detonate, they often still pose as a significant risk to the bomb squads that are tasked with neutralizing them. In response to this, experts are now looking at using shaped charges to drive jets of incompressible fluids, like water, into IEDs in order to break up the components of the bomb without initiating a dangerous secondary explosion. My objective was and, as this project will continue into next year, is to discover what types of liquid shaped charges will produce the longest, best-penetrating jet for breaking up shielded IEDs (i.e. IEDs that are placed in cars, storage containers, etc.). I researched and intend to implement a 3-dimensional, purely Lagrangian, finite element, hydrodynamic code in order to model the jets produced by various shapes of liquid charges. Specifically, my code will utilize a mid-point type integration scheme coupled with a predictor/corrector solution strategy in order to yield an explicit, iterative algorithm that will accurately modeling a fluid under explosive pressure. The entirely Lagrangian simulation environment of my code will significantly reduce the time required to compute this model as it eliminates the complex and cumbersome remapping step of contemporary arbitrary Lagrangian-Eulerian (ALE) codes. This may at first seem ill posed for the modeling of liquid shaped charges, due to the high deformation/shock physics that they involve, but recent experiments with fully-Lagrangian ballistic penetration environments that apply remeshing libraries have proven otherwise [2]. With the application of mesh healing and optimization packages like HealMesh, I will be able to effectively avoid extreme mesh distortion and thereby preserve a stable time step size in a purely Lagrangian simulation environment.

Statement of Problem

The objective of this project is to provide a computational test lab in which one can experiment with different shapes of liquid liners for directional charges. To this end, this project concerns itself with developing a computer code that is able to model explosively driven fluids with a high degree of accuracy.

Introduction

A hydrodynamic code requires an enormous breadth of complex ideas and disciplines to make it work. From advanced physics to numerical integration, hydrocodes are built on a framework that spans the academic efforts of many scientists and mathematicians over many years. Whole papers have been written upon a just single aspect of a hydrocode. As such, this paper cannot hope, in so few pages, to thoroughly and completely cover every single topic involved in the construction of my hydrocode. Instead, I will try to present basic explanations and sketch proofs of the most important concepts underlying my code. Undoubtedly, I will leave some minute details out, for which the reader may consult my sources.

Initial Principal of Lagrangian Hydrodynamics

Whether one is modeling a truss in a bridge, air moving over an airplane wing, or a fluid driven by an explosion there are certain identities that all materials, from a continuum perspective, must obey and by which we can model them [4]. These identities are provided by Langrangian continuum mechanics and are derived from the initial axiom that: given some domain Ω_0 that represents the material (reference) configuration of a continuum body in $\mathbb{R}^{n_d}$ (where n_d is the number of dimensions), there exists a diffeomorphism φ (i.e. a smooth, invertable map between manifolds) that

gives the current configuration of such body Ω at any time t. Even more strictly, φ can be defined in mathematical notation as

$$\begin{aligned} \boldsymbol{\varphi} : \Omega_0 &\to \Omega = \boldsymbol{\varphi}(\Omega_0) \ , \\ \boldsymbol{X} &\mapsto \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t) \ , \quad \forall \boldsymbol{X} \in \Omega_0, \ t \geq 0 \ , \end{aligned}$$

where $\boldsymbol{X}$ is the position vector for a fixed, material point in the reference configuration and $\boldsymbol{x}$ is the position vector for that same point in the current configuration. Physically however, the map φ is simply the deformation of a material body through time where the change in volume of the body due to the deformation is defined as

$$\begin{aligned} \boldsymbol{F} &= \nabla_{\boldsymbol{X}} \boldsymbol{\varphi} \ , \\ J &= \det(\boldsymbol{F}) \ , \end{aligned}$$

where J is the Jacobian determinant of the material gradient of φ [1].

Governing Equations

The langrangian map φ can be thought of as moving some control mass from one configuration/ volume to another. Therefore, Reynold's transport theorem

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_{\boldsymbol{x}} = \boldsymbol{\varphi}(\Omega_{\boldsymbol{X}})} \alpha = \int_{\Omega_{\boldsymbol{x}}} \left. \frac{\partial \alpha}{\partial t} \right|_{\boldsymbol{x}} + \int_{\partial \Omega_{\boldsymbol{x}}} \alpha \boldsymbol{v} \cdot \boldsymbol{n}_{\boldsymbol{x}}$$

can be applied in order to describe the time rate of change of the integral of some kinematic scalar variable α, where $\boldsymbol{v}$ is the velocity and $\boldsymbol{n}_{\boldsymbol{x}}$ is the unit vector normal to boundary of the current configuration. In combination with the master balance law:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{\boldsymbol{x}}=\boldsymbol{\varphi}(\Omega_{\boldsymbol{X}})}\alpha=\int_{\partial\Omega_{\boldsymbol{x}}}\boldsymbol{\gamma}\cdot\boldsymbol{n}_{\boldsymbol{x}}+\int_{\Omega_{\boldsymbol{x}}}\beta\ ,$$

where $\boldsymbol{\gamma}$ is the flux of α and β is to source/sink term, these two relations yield a system of equations or conservation laws [3] that, after simplification and localization, read

$$\begin{aligned}
\rho\, J &= \rho_0\ , \\
\rho\,\dot{\boldsymbol{v}} &= \rho\ \boldsymbol{g}+\nabla_{\boldsymbol{x}}\cdot\boldsymbol{\sigma}\ , \\
\rho\,\dot{E} &= \rho\ \boldsymbol{g}\cdot\boldsymbol{v}+\rho\ r+\nabla_{\boldsymbol{x}}\cdot\left(\boldsymbol{\sigma}^{T}\boldsymbol{v}+\boldsymbol{q}\right)\ , \\
\dot{\boldsymbol{u}} &= \boldsymbol{v}\ .
\end{aligned}$$

Here, ρ_0 is the reference density, ρ is the current density, $\boldsymbol{g}$ is the body force, $\boldsymbol{v}$ is the velocity, $\boldsymbol{\sigma}$ is the Cauchy stress (a symmetric tensor), E is the total energy where $E=\epsilon+\boldsymbol{v}\cdot\boldsymbol{v}/2$, ϵ is the internal energy, r is the energy source term, and $\boldsymbol{q}$ is the heat flux [1]. (It is important to note that body forces will be neglected in the actual Lagrangian computation of my code because they are very small when compared to the forces exerted on the boundaries of the material that I am modeling.)

The above equations can provide a system by which I can obtain a solution for the position and velocity of an explosively-driven liquid, but first they must first be closed with two additional equations. In short, these equations must give my mathematical algorithm a method of finding the Cauchy stress tensor in terms more the more clearly variables of density and internal energy. Therefore, these equations take the form of a constitutive equation

$$\boldsymbol{\sigma}\ =\ -p\boldsymbol{I}_{n_d\times n_d}$$

that is dependent upon the state/class of the body, where p is the pressure and $\boldsymbol{I}_{n_d \times n_d}$ is the identity matrix, and an equation of state

$$p = \hat{p}(\rho, \epsilon)$$

which is specifically dependent upon the material being modeled [4]. For the purposes of this code, I will only consider the above mentioned the fluid constitutive equation because I am only modeling liquids.

Numerical Formulation

Due to the fact that a computer has finite amount of memory, the above equations must be discretized in both space and time in order to form a problem that can be evaluated computationally. My code will leverage finite element approximations for the spatial dicretizations (because of their high degree of resolution) and use an iterative predictor/corrector strategy to approximate the kinematic and thermodynamic variables in time. This choice of algorithm leads to the following computational derivation of the above mentioned conservation laws:

$$\textit{Find } \boldsymbol{v} \in \mathcal{S}_{\boldsymbol{\kappa}}^h, \textit{ such that, } \forall \boldsymbol{\psi}_{\boldsymbol{\kappa}}^h \in \mathcal{V}^h,$$

$$\int_{\Omega_0} \boldsymbol{\psi}_{\boldsymbol{\kappa}}^h \cdot \rho_0 \left(\boldsymbol{v}_{n+1} - \boldsymbol{v}_n \right) \, \mathrm{d}\Omega_0$$
$$+\Delta t \int_{\Omega_{n+1/2}} (\nabla_{\boldsymbol{x}} \boldsymbol{\psi}_{\boldsymbol{\kappa}}^h)_{n+1/2} : \tilde{\boldsymbol{\sigma}}_{n+1/2} \, \mathrm{d}\Omega - \Delta t \int_{\Gamma_{n+1/2}^h} \boldsymbol{\psi}_{\boldsymbol{\kappa}}^h \cdot \boldsymbol{t}_{n+1/2} \, \mathrm{d}\Gamma = 0$$

$$\textit{Find } \epsilon \in \mathcal{S}_{\gamma}^h, \textit{ such that, } \forall \psi_{\gamma}^h \in \mathcal{V}^h,$$

$$\int_{\Omega_0} \psi_{\gamma}^h \rho_0 \left(\epsilon_{n+1} - \epsilon_n \right) \, \mathrm{d}\Omega_0 - \Delta t \int_{\Omega_{n+1/2}} \psi_{\gamma}^h \, (\nabla_{\boldsymbol{x}} \boldsymbol{v})_{n+1/2} : \tilde{\boldsymbol{\sigma}}_{n+1/2} \, \mathrm{d}\Omega = 0$$

$$\textit{Find } \rho \in \mathcal{S}_{\gamma}^h, \textit{ such that, } \forall \psi_{\gamma}^h \in \mathcal{V}^h,$$

$$\int_{\Omega_0} \psi_{\gamma}^h \rho_0 \, \mathrm{d}\Omega_0 = \int_{\Omega_0} \psi_{\gamma}^h \rho J \, \mathrm{d}\Omega_0 = \int_{\Omega} \psi_{\gamma}^h \rho \, \mathrm{d}\Omega$$

where $\mathcal{S}_{\kappa}^{h}$, $\mathcal{V}_{\kappa}^{h}$, $\mathcal{S}_{\gamma}^{h}$, and $\mathcal{V}_{\gamma}^{h}$ are the kinematic and thermodynamic trial and test function spaces, respectively, that give piece-wise linear approximations for kinematic variables and piecewise constant approximations for thermodynamic variables. Here, subscripts are used to indicate a certain variable at either half the time step or the next time and the symbol $\tilde{\sigma}$ indicates the algorithmic stress tensor, including both thermodynamic pressure and artificial viscosity to deal with the Gibbs phenomenon resulting from shocks. These equations are then iterated, in local matrix form using an Adams-Bashforth scheme, until second order accuracy is achieved and the simulation is advanced [1].

Mesh Optimization

As stated before, the posed computational problem involves high deformation. Consequently, the Lagrangian mesh cannot simply be advanced through time because its distortion will reduce the time step of the computational solution to impractical sizes. For it to be accurate, my hydrocode must have some instrument to account for and eliminate deformation-induced distortions within a Lagrangian mesh. To this end, it will employ the software library HealMesh.

HealMesh is a software package that provides functionality for mesh healing and optimization by implementing a suite of hillclimbing methods. This means that each method of HealMesh sweeps over the Lagrangian mesh and investigates local changes; if a proposed change improves the quality of the mesh then it is accepted. These local changes are divided into two categories: geometric transformations/moving nodes, and topological transformations/changing connectivities [2].

Before discussing the geometric and topological optimization however, it is necessary to specify the mesh quality metrics in HealMesh that my code will use. Metrics give the creiteria and mathematical basis for mesh optimization and, from the results of previous codes using HealMesh, I have decided to

use the condition number metric and the mean ratio metric. These metrics are algebraic, meaning that they are functions of the Jacobian matrix S of the map φ , and are given by

$$\kappa(S) = \frac{|S|}{|S^{-1}|} \qquad \eta(S) = \frac{|S|^2}{3\det(S)^{2/3}},$$

where $\kappa(S)$ is the condition number metric and $\eta(S)$ is the mean ration metric. The matrix S becomes singular as the element volume vanishes and so these metrics can be thought of as measuring distance from singular matrices. The advantage to these metrics is that they are sensitive to all types of distortions within the mesh and have continuous derivatives. This allows optimizations to utilize gradient methods(detailed below) and operate with a simple set of only two metrics [2].

The geometric optimization methods in HealMesh first sweep over all interior movable nodes that have neighboring elements with poor quality and assemble a complex of the connectivity of all the vertices, edges and faces of the elements surrounding those nodes. This complex is then broken down into simplicies, subsets, for which the ℓ_2 norms of the quality metrics are optimize, equalizing the connectivities for distorted nodes. Because of my previous choice of metrics that are differentiable, the far more efficient, quasi-Newton, Broyden–Fletcher–Goldfarb–Shanno (BFGS) method can be applied to optimize the ℓ_2 norm— decreasing the time it takes a geometric mesh optimization method to run [2].

For topological optimizations, HealMesh sweeps over the mesh in the same hill-climbing method used for geometric optimization and adds or removes edges and faces in m-n flips (replaces m elements with n elements). These m-n flips produce a new local set of elements for which new metrics are determined. If these new metrics are an improvement then the changes are adopted. Furthermore, if an element is so severely distorted that a simple flip cannot markedly improve its condition number or mean ratio metrics, then the element can be removed by composite sequences of flips such as 2-3, 3-2, 2-2, and 4-4 [2].

Conclusion

Having to teach myself all the afore mentioned material, I failed to actually write a successful hydrodynamic code this year. I did, however, develop a plan, that I will implement next year, for an accurate and fast algorithm that will solve my computational problem. My efforts this year didn't result in a hydrocode, but they gave me the tools to construct one next year.

References

[1] G. Scovazzi, E. Love, and M. Shashkov. A Multi-Scale Q1/P0 Approach to Lagrangian Shock Hydrodynamics. Technical report, Sandia National Laboratories, Albuquerque, NM, March 2007. Available at: http://www.cs.sandia.gov/~gscovaz/index.html.

[2] S. Mauch, L. Noels, Z. Zhao, and R Radovitzky. Lagrangian Simulation of Penetration Environments Via Mesh Healing and Adaptive Optimization. *Transformational Army Science and Technology: Proceedings of the 25th US Army Science Conference*, Tech Science Press: Norcross, GA, 2008.

[3] G. Scovazzi and T. Hughes. Lecture Notes on Continuum Mechanics on Arbitrary Moving Domains. Technical report, Sandia National Laboratories, Albuquerque, NM, November 2007. Available at: http://www.cs.sandia.gov/~gscovaz/index.html.

[4] C.M. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics*. Springer: New York, NY, 1999.