LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-476935

Parallel Detonation Shock Dynamics Algorithm for Insensitive Munitions using ALE3D

D. Pfau, F. Najjar, B. McCandless, J. Yao, A. Nichols III

March 29, 2011

26th International Symposium of Ballistics
Miami, FL, United States
September 12, 2011 through September 16, 2011

Disclaimer

COVER SHEET

Title: Parallel Detonation Shock Dynamics Algorithm for Insensitive Munitions using ALE3D

Authors: D. Pfau[1]
F.M. Najjar[2]
J. Yao[2]
B. McCandless[2]
A. Nichols III[2]

Classification: UNCLASSIFIED

Distribution Statement: Public Release Unlimited Distribution

Article Number: 12085

[1]U.S. Army RDECOM-ARDEC, Picatinny, NJ 07806-5000, USA
[2]Lawrence Livermore National Laboratory (LLNL), B-Division, Livermore, CA 94551, USA

The Detonation Shock Dynamics (DSD) method consists of an asymptotic analysis of the reactive Euler flow equations and describes the evolution of a multi-dimensional curved detonation near the Chapman-Jouguet (CJ) condition. We have developed a parallel and scalable DSD modular library formulated with a narrow band approach and integrated in the multiphysics hydrodynamics code, ALE3D. Our goal is to develop a predictive DSD capability that computes the lighting times for ideal and non-ideal high-explosives (HE). The focus on insensitive munitions applications includes modeling damaged explosives, the design of smaller munitions or complex geometries, and the study of charges with less sensitive HE and larger critical diameters. The poster will describe the DSD approach and presents several examples relevant to the ballistics community.

INTRODUCTION AND BACKROUND

Current state of the art analysis pertaining Insensitive Munitions (IM) frequently involves desensitizing the main charge explosives. This makes the munitions less vulnerable to unexpected insults, such as impacts from fragments or bullets and nearby explosions. Further, munition safety has been significantly improved. However, a drawback of such desensitization is having high explosives (HE) that behave less ideally, and care must be taken in the design of the munition such that it will function reliably when required.

Detonation Shock Dynamics (DSD), as implemented into shock codes, is a computational approach that will allow highly accurate modeling of the HE reaction, without the high burden of computational expense that traditional reactive flow simulations suffer from (such as Ignition and Growth). With such tool, weapons designers can predict HE behavior in complex geometries and ensure their warheads will function as intended, both for safety as well as effectiveness. The present paper with its accompanying poster will deal with implementing the DSD techniques into ALE3D, the massively scalable multi physics code package developed and published by Lawrence Livermore National Laboratory (LLNL).

DSD THEORY

The first theory behind DSD is that the velocity of the detonation wave normal to the shock wave was only function of the curvature. Combined with an appropriate boundary condition describing the HE/confinement interface, DSD provides a complete description of detonation propagation [1-3]. This mathematical approach serves to keep track of sub-scale (compared to system-level length scale) physics in the reaction zone without needing to march element-by-element through the reaction zone with a reactive flow approach.

More recent experiments and simulations have showed that this approach is not always sufficient, especially with non-ideal explosives such as IM main charges. Work by Bdzil, et al., extended the theory to include higher order terms to better capture expansion, plus state-sensitive reaction rates [1]. Work has already been completed to parameterize DSD for traditional IM explosives (such as PBX9502 [2]), non-ideal materials (such as

Ammonium Nitrate-Fuel Oil [1]), and work is currently ongoing to parameterize the Army IM HE IMX-104.

ALE3D MULTIPHYSICS COMPUTATIONAL FRAMEWORK

The DSD capability was implemented as an external library in the multiphysics code, ALE3D[4]. ALE3D consists several physics modules including hydrodynamics, magneto-hydrodynamics, chemistry and thermal transport. The mathematical formulation is based on an Operator-Split method and invokes an arbitrary Lagrangian-Eulerian (ALE) approach in two and three-dimensional (3-D/2-D) Cartesian configuration as well as 2-D axisymmetric one. Simulations can run serially or on a massively parallel system with thousands of processing nodes.

The capabilities available in ALE3D allow solving the evolution equations either in a pure Lagrangian, pure Eulerian, or in an ALE manner. Within an Eulerian or ALE construct, a remap formalism is invoked, allowing the advection of the conserved variables on the moving mesh. In order to simulate hydrodynamic flow across shock fronts, it is necessary to add some form of numerical dissipation to the conservation equations. This was achieved through the use of artificial dissipation (typically referred to as artificial viscosity). Further, to control numerical instabilities due to the finite-element formulation, hour-glass mode control is invoked. Hence, the contributions of artificial viscosity and hour-glass mode control have been included. The governing equations are discretized using a finite-element method and are solved in a weak form. The spatial discretization is of hybrid form where the node coordinates and velocities *are node-centered* variables; while the density, pressure, internal energy, temperature, etc are *cell-centered* quantities. The basis function consists of bi-linear and piecewise constant for the velocities and pressure, respectively. The time integration approach follows a staggered explicit formulation where the velocity fields are computed in a staggered manner from the zonal state variables. Thus, the Lagrangian coordinates, accelerations, pressure, energy, and mass are centered in time at t^n; while the velocities, are centered in time at $t^{n-1/2}$, where n indicates the time step number. An operator-split scheme is invoked for the hydrodynamics and thermal solvers.

DSD IMPLEMENTATION

The DSD model currently implemented in ALE3D is based on the asymptotic analysis of the reactive Euler (inviscid flow) equations. It is easily handles arbitrarily connected quadrilateral (2-D) and hexahedral (3-D). The basic algorithm is based on a narrow band concept. Briefly described, the nodes in the computational mesh are set to one of four statuses: (i) inactive (nodes have a default distance), (ii) feasible (whose distance to the burn front is to be computed; the nodes are unlit neighbors of half-lit cells)), (iii) trusted (whose distance is defined to the surface. These are nodes of half-lit cells), and (iv) retired (node is lit and is deleted from the search list unless it is located within a search radius). The narrowband corresponds to a combination of feasible and trusted nodes. Figure 1 shows a schematic of the burn front and the various nodes in their light status. The burnfront and narrowband are integrated in times using a 3rd-order

Runge-Kutta scheme. Figure 2 describes schematically the time evolution of the burnfront with the narrowband as implemented currently.

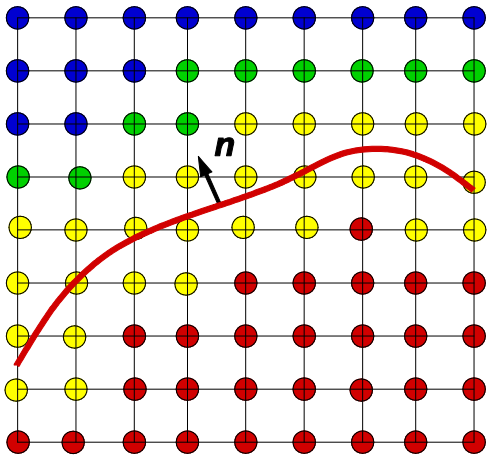
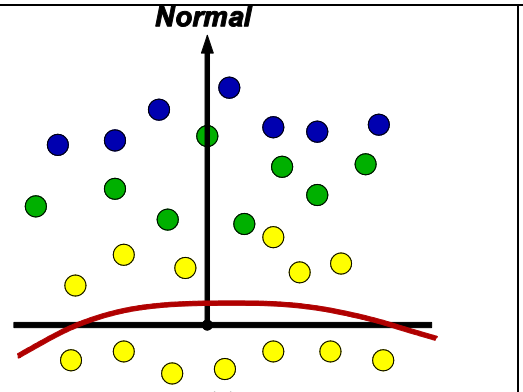
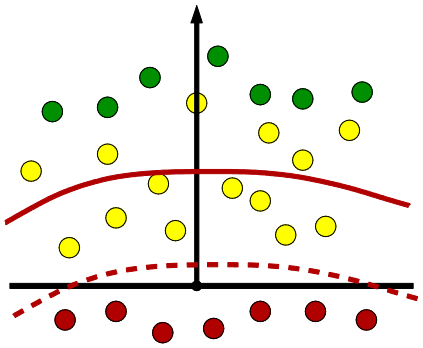


Figure 1: Schematic of the DSD node status with burn front (shown as a red curve). Blue: inactive; Green: feasible status; Yellow: trusted status; Red: retired status. The narrowband corresponds to the yellow and green nodes.



(a)



(b)

Figure 2: Schematic of advancing the burnfront and updating the narrowband. (a) burn front location and status of nodes at t^n; (b) burnfront evolves in time at t^{n+1} with updated node statuses. Red curve is burnfront (dashed line is previous state of the burnfront)

The overall flowchart of the DSD algorithm is schematically presented in Figure 3 showing the time evolution process. Currently a boundary treatment of the HE-inert interface is in development and this is shown as a red box in the figure.

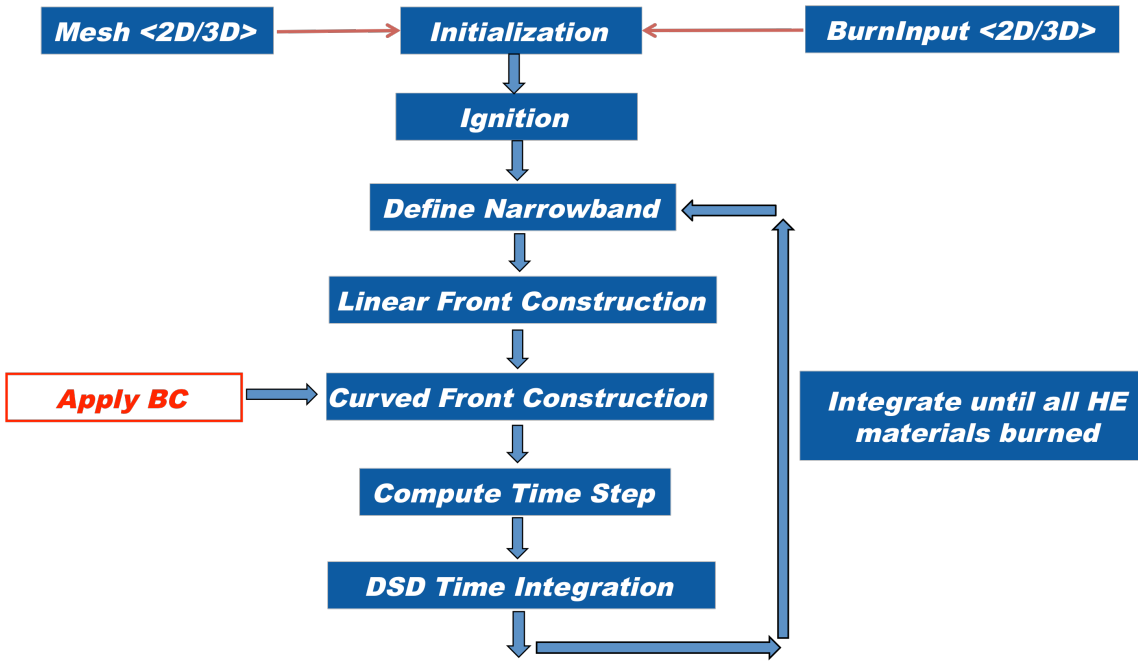
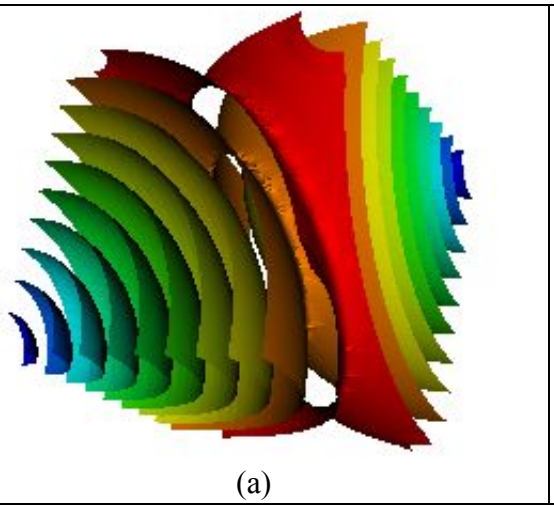
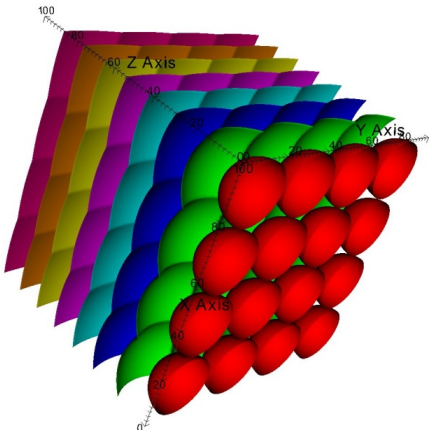


Figure 3: Flowchart of the DSD algorithm with its current implementation in ALE3D. The red box showcases a future capability where a boundary condition treatment is being included.

Figure 4 provides two representative examples of the lighting times computed with various detonation points, showcasing the DSD capability.



(a)



(b)

Figure 4: Examples of lighting times computed with the DSD algorithm developed for ALE3D for two representative configurations: (a) dual detonation points in a cube; (b) 16 detonation points in a rectangular box.

DSD VERIFICATION CASES

Several verification cases have been performed to evaluate the accuracy and convergence of the developed modular DSD algorithm. These cases consist of 2-D and 3-D configurations with known analytical solutions. The first 2-D test case corresponds to the one proposed by Aslam *et al.* [5]. The computational domain is a square domain extending $[0,1]^2$, and the detonation point is located at (0,0) with a detonation radius of 0.2. The D_n-κ relation is based on a linear relation as follows: $D_n = D_{CJ}(1-\beta*\kappa)$; with $D_{CJ}=1$ and $\beta=0.1$. Figure 5 plots the L1 error norm versus the mesh resolution, comparing Aslam's results with the current ones. It is seen that both algorithms the error slope is second-order accurate. However, the current DSD algorithm leads to an error that is an one-order of magnitude smaller than the one proposed by Aslam *et al.*[5]

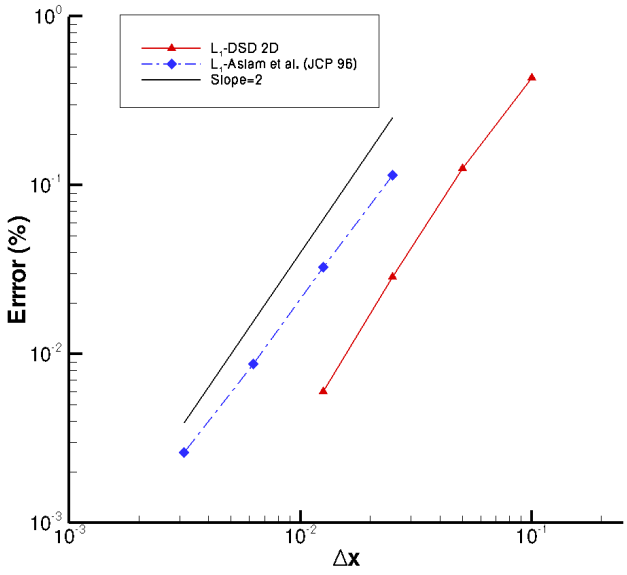


Figure 5: Error vs mesh size for Aslam's 2-D test case: blue line corresponds to Aslam's value and red line represents the current DSD algorithm in ALE3D. The black line is a slope=2 line.

A 3-D testcase is investigated next where the computational domain consists of a cubedomain extending $[-20,20]^3$, and the detonation point is located at (0,0,0) with a detonation radius of 10. The D_n-κ relation is based on a linear relation as follows: $D_n = D_{CJ}(1-\beta*\kappa)$; with $D_{CJ}=1$ and $\beta=0.2$. Table I summarizes the errors and CPU times for various mesh sizes (a weak scaling analysis is performed where the amount of work is fixed per processor). It is seen that the L1 error norm has a slope of 1.5; while good parallel scalability is achieve. Further, it is noted that the DSD computed the lighting time on a mesh size of 33M elements, running on 512 processors.

TABLE I. L1 ERRORS AND CPU TIMES FOR 3-D DSD TEST CASE

Number of Processors	Mesh Size	L1 Error (%)	CPU Time (s)
1	10^3	1.57	1.2
8	20^3	0.84	3.2
16	40^3	0.13	15.7
32	80^3 [0.5M]	0.064	103
64	160^3 [4M]	0.026	831
512	320^3 [33M]	0.010	3092

DSD PARALLEL PERFORMANCE

An initial parallel algorithm has been developed that works in 2D and 3D computational space for arbitrarily connected quadrilateral or hexahedral meshes. The DSD algorithm, presented in the previous section, invokes the same domain decomposition used by the hydrodynamics and other physics packages in ALE3D. Such decomposition is scalable in memory, requiring a minimum amount of data communication. However, the workload, in general, is not balanced for DSD: as the burn front propagates through the material, some processors will be busy, while others will be idle. Nevertheless, there are significant speedups in DSD burn algorithm when run in parallel, with the total DSD processing time remaining small compared to the time spent running a simulation in ALE3D.

A strong scaling study has been conducted where four mesh configurations are evaluated: a small 2D mesh (1M zones), a large 2D mesh (11M zones), a small 3D mesh (1M zones), and a large 3D mesh (11M zones). These problems were setup as either a square or a cube computational domain, with a detonation point in the center. The study ran from 1 processor to 1024 processors. Figure 6 summarizes the parallel performance of the four configurations studied. As expected, the larger meshes demonstrated better scaling than the smaller meshes, and the 3D configuration demonstrated better scaling than the 2D configuration.

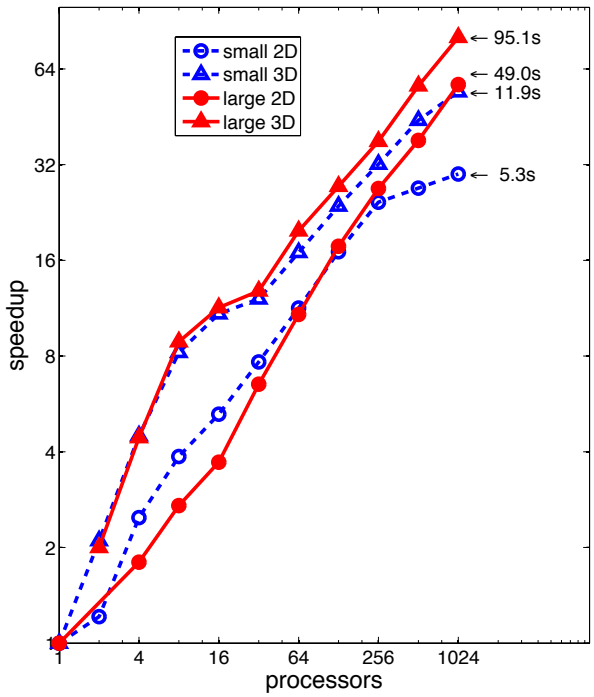
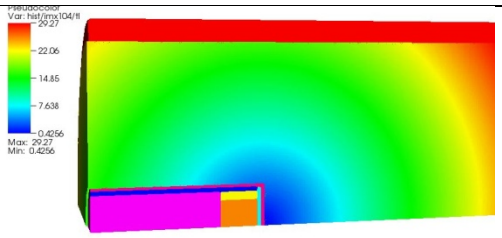


Figure 6: Strong scalability study for DSD algorithm in ALE3D for up to 1024 processors.

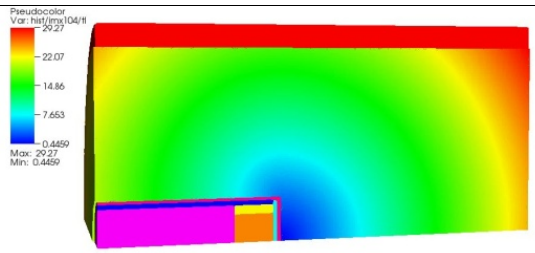
BENEFIT TO THE BALLISTICS COMMUNITY

The benefit to the munitions modeling community is accuracy in our models at a lower computational cost than alternatives. As mentioned previously, if designers and engineers can attain the accuracy of reactive flow at the computational cost that is closer to line-of-sight lighting, this will benefit all stakeholders. Additionally, DSD can capture wave propagation in IM explosives which may not behave the same as traditional explosives.

Having accurate answers for non-ideal explosives is becoming of greater concern. Figure 7 uses a notional cylindrical configuration with a booster pellet to illustrate the difference between a 2D burn and DSD burn. If one were to use a traditional lighting routine which has been calibrated from traditional methods (e.g. cylinder expansion tests), the predicted wall velocities will be quite different, especially where the detonation wave wraps around corners at the corner of the booster. DSD is also expected at a future point to be able to predict the extinguishing of detonation reactions due to phenomena such as corner turning.



(a)



(b)

Figure 7: 2D lighting time contours for a booster using two approaches (a) line-of-sight burn; (b) DSD burn. The burn front proceeds radially away from the booster. Note the difference in lighting times, especially in the aft region of the booster.

SUMMARY

Having DSD implemented into a production code like ALE3D affords weapons analysts a much better tool in predicting detonations through their warheads and other energetic devices. This capability will become more and more important as the weapons community moves towards HE fills with non-ideal behavior and critical diameters approaching the size of their munition.

ACKNOWLEDGMENTS

The work described in this paper was done in part to support the High Performance Computing Modernization Office IM Multiscale Reactive Modeling Institute (IM MSRM) with leveraged funds from the Armament Research Development and Engineering Center (ARDEC) Tech Base.

LLNL Distribution Statement: This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

REFERENCES

[1] Bdzil, J.B., Aslam, T.D., Catanach, R.A., Hill, L.G., Short, M., "DSD Front Models: Nonideal Explosive Detonation in ANFO," Proceedings of the 21st International Ballistics Symposium, 2002.

[2]Aslam, T.D., Bdzil, J.B., Hill, L.G., "Extensions to DSD Theory: Analysis of PBX9502 Rate Stick Data," Proceedings of the 11th International Detonation Symposium, 1998.

[3]Short, M., Bdzil, J.B., "Propagation laws for Steady Curved Detonations with Chain-Branching Kinetics," *Journal of Fluid Mechanics*, 2002.

[4] Nichols A. L. III, Editor, "User's Manual for ALE3D, An Arbitrary Lagrange/Eulerian 2D and 3D Code System", Lawrence Livermore National Laboratory, LLNL-SM-433954, 2010.

[5] Aslam, T.D., Bdzil, J.B, and Stewart, D.S. , "Level Set Methods Applied to Modeling Detonation Shock Dynamics", J. Comp. Physics, Vol. 126, pp. 390-409, 1996.