LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-732038

Level set methods for detonation shock dynamics using high-order finite elements

V. Dobrev, F. Grogan, T. Kolev, R. Rieben, V. Tomov

May 26, 2017

Disclaimer

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

Level set methods for detonation shock dynamics using high-order finite elements

V. A. Dobrev[a], F. C. Grogan[b,c,*], T. V. Kolev[a], R. N. Rieben[c], V. Z. Tomov[a]

[a] *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551, United States*
[b] *Center for Computational Mathematics, Department of Mathematics, University of California at San Diego, La Jolla, CA 92093, United States*
[c] *Weapons and Complex Integration, Lawrence Livermore National Laboratory, Livermore, CA 94551, United States*

Abstract

Level set methods are a popular approach to modeling evolving interfaces. We present a level set advection solver in two and three dimensions using the discontinuous Galerkin method with high-order finite elements. During evolution, the level set function is reinitialized to a signed distance function to maintain accuracy. Our approach leads to stable front propagation and convergence on high-order, curved, unstructured meshes. The ability of the solver to implicitly track moving fronts lends itself to a number of applications; in particular, we highlight applications to high-explosive (HE) burn and detonation shock dynamics (DSD). We provide results for two- and three-dimensional benchmark problems as well as applications to DSD.

Keywords: level sets, advection, detonation shock dynamics (DSD), finite element methods, high-order methods

1. Introduction

Level set methods were introduced by Osher and Sethian in [1] to implicitly track dynamically evolving interfaces. The basis for these methods is the ability to embed an n-dimensional interface into an $n+1$-dimensional surface—called the level set function—and track the evolution of the interface by evolving the function instead. This approach arose as an alternative to tracking the interface through surface parametrization, where moving nodes, which can cross over themselves or separate into distinct regions, can lead to loss of accuracy and be difficult to implement algorithmically. In level set methods, the interface is represented as the zero-contour of the level set function, or the zero level set. Given the speed of the interface, it is then simple to describe movement of the front with a PDE-based advection equation [2, 3].

The simplicity of the foundation behind level set methods lends itself to a number of applications [2, 4, 5], including image processing [6], fluid dynamics [7], and crystal growth [8]. In this paper, our interest is in the level sets' ability to model high-explosive detonation fronts through detonation shock dynamics (DSD) [9, 10, 11]. DSD is a theory describing the evolution of multidimensional detonation shocks using the velocity and curvature of the front. The modeling of high-explosive (HE) energy release (or burn) typically falls into two categories: geometric methods in which the times of energy release are pre-calculated based on the HE detonation velocity and calculated distances from a given detonation point, and more sophisticated reactive chemistry models which directly compute the HE energy release based on solving reaction rate equations for the reactants and products [12, 13]. In this paper, we are concerned exclusively with geometric methods, and in particular, with DSD methods.

When using level sets for the purposes of DSD, given a level set function $\phi(\mathbf{x}, t)$, the burn front is represented by the contour $\phi = 0$. The area where $\phi < 0$ corresponds to the products of the reaction, while

*Corresponding author
Email address: fgrogan@ucsd.edu (F. C. Grogan)

the area with $\phi > 0$ corresponds to the unburned explosive. The goal of the DSD solver is to evolve (or propagate) the burn front $\phi = 0$ from some initial configuration to a final time where it has completely propagated through the region of interest. The resulting solution can be used to construct a so called *burn table*.

In our approach, we spatially discretize the described PDE using high-order finite elements. Since we wish to address problems which may have complex geometries and boundary conditions on unstructured meshes, using finite elements is more flexible over traditional finite difference methods. In particular, to address developing discontinuities in the dynamically evolving interface, we use the discontinuous Galerkin (DG) approach, which allows us to handle discontinuities across elements. This approach allows the propagating front to easily handle complicated geometries and curved, highly-unstructured 2D and 3D meshes. After spatial discretization, we use standard explicit time integrators (typically matching the temporal order of the time integrator to the spatial discretization order) to evolve the resulting ODE.

The paper is organized as follows. In the following section, we present the level set method as well as level set reinitialization, which is performed for stability purposes. In Section 3, we present the details of our finite element spatial discretization. Section 4 is a brief description of the multi-stage time integrator used for both level set advection and redistancing. In Section 5, we discuss the level set method for DSD and application to high-explosive burn. Numerical results are presented in Section 6, which include typical level-set benchmarks as well as more complex front propagation problems involving corner turning and shadow surfaces.

2. Level Set Method

Level set methods are centered around evolving a scalar function $\phi(\mathbf{x}, t)$ in time through advection:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{1}$$

where $\mathbf{u} = u\, \nabla\phi/|\nabla\phi|$ is the velocity of the front, which we denote as the contour $\phi = 0$. Note that when well-defined, $\nabla\phi/|\nabla\phi|$ is a unit vector field orthogonal to the level sets of ϕ. In this work, we will specify u as constant, but in general DSD applications $u = D_{CJ} - \alpha(\kappa)$, where D_{CJ} is the Chapman-Jouguet detonation velocity and $\alpha(\kappa)$ is a function of curvature. Here, the curvature is the divergence of the front normal,

$$\kappa = \nabla \cdot \mathbf{n_f} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right). \tag{2}$$

2.1. Reinitialization

During evolution, ϕ may drift from its initial value, creating steep or shallow gradients leading to inaccurate approximation of the spatial derivatives. For better accuracy, it is standard to reinitialize ϕ according to the following equation [14]:

$$\frac{\partial \phi}{\partial \tau} + \left(S(\phi) \frac{\nabla \phi}{|\nabla \phi|} \right) \cdot \nabla \phi = S(\phi), \quad \text{where } S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla \phi|^2 h(x)}}, \tag{3}$$

where $h(x)$ is a mesh size representing the resolution of the spatial discretization. Specifically, $h(x)$ is calculated by taking the smallest singular value of the Jacobian of the transformation from the perfect reference element at the center of said element. The above equation redistances the level set function to a signed distance function. Although this method is widely used throughout the literature, its exact implementation remains somewhat ad-hoc. When and for how many iterations to run Eq. (3) varies between algorithms, and can cause stability to become problem-dependent. We use a modified redistancing equation:

$$\frac{\partial \phi}{\partial \tau} + \left(S(\phi) \frac{\nabla \phi}{|\nabla \phi|} \right) \cdot \nabla \phi = S(\phi) + \mu(h(x)) \nabla \cdot \left[\left(1 - \frac{1}{|\nabla \phi|} \right) \nabla \phi \right], \tag{4}$$

where $S(\phi)$ is the same as in Eq. (3) and the additional right-hand side term, adapted from [15], provides artificial viscosity/diffusion for stability purposes. This term acts as a penalty: when $|\nabla\phi|$ strays far from 1 the term $(1 - 1/|\nabla\phi|)$ is a diffusion rate aiding the level set function with redistancing. The diffusion coefficient $\mu(h(x))$ is mesh-dependent, and as shown in Subsection 6.6, it results in stable propagation on complex meshes. In theory, the level set is reinitialized until Eq. (3) reaches a steady-state. However, during numerical evolution the interface will move during reinitialization, so care must be taken not to redistance too much [16].

3. Finite Element Discretization

We spatially discretize the level set equations using a discontinuous Galerkin method in space [17]. Inheriting features of the finite volume and finite element frameworks, DG methods are standard in numerically evolving level set equations as they can capture the natural discontinuities that develop during advection.

To derive the spatial discretization for a fixed t, we start from a splitting of the computational domain into a mesh with elements (zones) $\{z\}$ on which we introduce a piecewise polynomial discontinuous finite element space with basis functions $\psi_i(\mathbf{x}, t)$. This is a finite-dimensional subspace of L_2. Since we are interested in high-order methods, the basis functions can be mapped high-order polynomials, and the mesh elements are allowed to be curved (i.e. described by a high-order polynomial mapping).

We approximate the level set function by its expansion in the finite element basis,

$$\phi(\mathbf{x}, t) = \sum_{i=1}^{N} \boldsymbol{\phi}_i(t)\psi_i(\mathbf{x}, t),$$

where $\boldsymbol{\phi}(t)$ is the discrete vector of degrees of freedom in the finite element space. Starting from the weak form of the advection equation tested with a finite element basis function $\psi \in L_2$, we integrate by parts to obtain

$$\int_\Omega \frac{\partial\phi}{\partial t}\psi = \int_\Omega (\mathbf{u}\cdot\nabla\psi)\phi - \int_{\partial\Omega} \phi(\mathbf{u}\cdot\mathbf{n})\psi, \tag{5}$$

which can be expressed as

$$\mathbf{M}\frac{d\boldsymbol{\phi}}{dt} = \mathbf{K}\boldsymbol{\phi}, \tag{6}$$

where $\mathbf{M}$ is the mass matrix and $\mathbf{K}$ is the advection matrix for the finite element space:

$$\mathbf{M}_{ij} = \sum_z \int_z \psi_i \cdot \psi_j, \quad \mathbf{K}_{ij} = \sum_z \int_z (\mathbf{u}\cdot\nabla\psi_j)\psi_i - \sum_f \int_f (\mathbf{u}\cdot\mathbf{n})[\![\psi]\!]\phi_u. \tag{7}$$

Here, the integrals are over the zones z and faces f of the mesh, and we use the jump operator $[\![\psi]\!] := \psi^+ - \psi^-$, $\psi^\pm = \lim_{\epsilon\to 0^+} \psi(\mathbf{x} \pm \epsilon\mathbf{n}(\mathbf{x}))$, and the upwind operator

$$\phi_u := \begin{cases} \phi^+ & \text{if } \mathbf{u}\cdot\mathbf{n} > 0 \\ \phi^- & \text{if } \mathbf{u}\cdot\mathbf{n} < 0 \\ \phi & \text{if } \mathbf{u}\cdot\mathbf{n} = 0. \end{cases} \tag{8}$$

We perform a similar discretization of our reinitialization equation, Eq. (4). For simplicity, we rewrite Eq. (4) as

$$\frac{\partial\phi}{\partial\tau} + \hat{\mathbf{u}}\cdot\nabla\phi = S(\phi) + \nabla\cdot[c\nabla\phi], \tag{9}$$

where

$$\hat{\mathbf{u}} = \left(S(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right), \quad c = \mu(h(x))\left(1 - \frac{1}{|\nabla\phi|}\right). \tag{10}$$

The finite element discretization is then

$$\mathbf{M}\frac{d\boldsymbol{\phi}}{d\tau} = (\mathbf{K} + \mathbf{D})\boldsymbol{\phi} + \mathbf{f}, \tag{11}$$

where $\mathbf{M}$ and $\mathbf{K}$ are the mass and advection matrices from Eq. (6), and $\mathbf{D}$ and $\mathbf{f}$ are given by

$$\mathbf{D}_{ij} = \sum_z \int_z -c\nabla\psi_i \cdot \nabla\psi_j - \sum_f \int_f \{(c\nabla\phi)\cdot\mathbf{n}\}[\![\psi]\!] + \sigma[\![\phi]\!]\{(c\nabla\psi)\cdot\mathbf{n}\} + \kappa\left\{\frac{c}{h(x)}\right\}[\![\phi]\!][\![\psi]\!], \tag{12}$$

$$\mathbf{f}_i = \sum_z \int_z S(\phi)\psi_i, \tag{13}$$

where $\{\} =$ is the average. We choose $\sigma = -1$ and $\kappa > 0$ which yields a symmetric interior penalty Galerkin method [18].

4. Time Integration

Once we have obtained the semi-discrete form Eqs. (6) and (11), we integrate both sets of equations using an explicit Runge-Kutta operator usually of order corresponding to the order of the finite elements space. For example, 2nd order elements implies a time discretization using two-stage Runge-Kutta for the advection steps:

$$\phi^{n+\frac{1}{2}} = \phi^n + \frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{K}\phi^n, \tag{14}$$

$$\phi^{n+1} = \phi^n + \Delta t\mathbf{M}^{-1}\mathbf{K}\phi^{n+\frac{1}{2}}, \tag{15}$$

as well as the reinitialization steps:

$$\phi^{n+\frac{1}{2}} = \phi^n + \frac{\Delta t}{2}\mathbf{M}^{-1}\left[(\mathbf{K}+\mathbf{D})\phi^n + \mathbf{f}\right], \tag{16}$$

$$\phi^{n+1} = \phi^n + \Delta t\mathbf{M}^{-1}\left[(\mathbf{K}+\mathbf{D})\phi^{n+\frac{1}{2}} + \mathbf{f}\right]. \tag{17}$$

We reinitialize after every five iterations of the advection equation. In an effort to address consistency with how long to reinitialize, we use variable redistancing. Using the fact that a signed distance function stipulates $|\nabla\phi| = 1$, we reinitialize until $|\nabla\phi|$ is within an acceptable order of 1, or the maximum number of iterations has been reached. This constraint prevents the level set from over-reinitializing. The detailed algorithm is presented in Alg. 1.

To illustrate the redistancing portion of the algorithm which solves Eq. (4), we briefly consider the test problem described in [19]. Specifically, we start with a distorted level set function:

$$\phi = ((x-1)^2 + (y-1)^2 + 0.1)(\sqrt{x^2+y^2} - 1), \tag{18}$$

which has steep gradients that will affect spatial accuracy if not corrected. Eq. (4) should reinitialize this distorted level set into the proper signed distance function

$$\phi = \sqrt{x^2+y^2} - 1. \tag{19}$$

Results from application of the redistance solver using quadratic (Q_2) elements on a $[0,1]\times[0,1]$ structured square mesh are shown in Fig. 1.

5. Application to DSD and High-Explosive Burn

Using level sets to model HE burn relies on the idea that the radius of the shock curvature is large compared to the one-dimensional, steady, reaction zone length. This relatively thin reaction zone can then be viewed as a front separating burnt from unburned explosive [20]. Using the language of level sets, we denote the zero-contour of ϕ as our detonation front. The material in front of $\phi = 0$, i.e. $\phi > 0$, are the reactants, while $\phi < 0$ designates the resulting products. The velocity coefficient used is the normal shock

Algorithm 1 Level set solver

$t \leftarrow 0$
$iter \leftarrow 0$
while $t \leq t_{final}$ **do**
  $\phi^{n+\frac{1}{2}} \leftarrow \phi^n + \frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{K}\phi^n$
  $\phi^{n+1} \leftarrow \phi^n + \Delta t\mathbf{M}^{-1}\mathbf{K}\phi^{n+\frac{1}{2}}$
  $iter \leftarrow iter + 1$
  if $iter \bmod 5 = 0$ **then**
    $\tau \leftarrow 0$
    while $\tau \leq \tau_{max}$ **or** $0.9 \leq |\nabla\phi| \leq 1.1$ **do**
      $\phi^{n+\frac{1}{2}} \leftarrow \phi^n + \frac{\Delta t}{2}\mathbf{M}^{-1}\left[(\mathbf{K}+\mathbf{D})\phi^n + \mathbf{f}\right]$
      $\phi^{n+1} \leftarrow \phi^n + \Delta t\mathbf{M}^{-1}\left[(\mathbf{K}+\mathbf{D})\phi^{n+\frac{1}{2}} + \mathbf{f}\right]$
      $\tau \leftarrow \tau + \Delta\tau$
    end while
  end if
  $t \leftarrow t + \Delta t$
end while
return ϕ

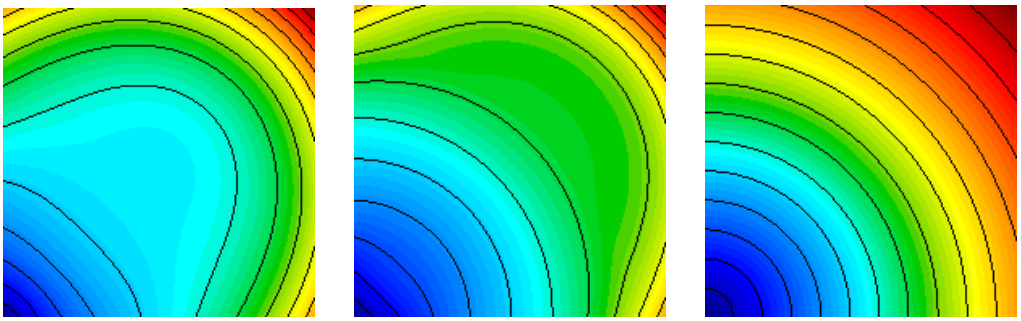


Figure 1: An initially distorted level set function (left) is reinitialized to a proper signed distance function (right) using Q_2 finite elements.

velocity $u = D_{CJ} - \alpha(\kappa)$, where D_{CJ} is the Chapman-Jouguet detonation velocity, or the normal speed of the detonation, and $\alpha(\kappa)$ is a function of curvature. As previously mentioned, we consider only the simple case of $\alpha = 0$ in this work.

DSD also takes into account additional boundary conditions, namely

$$\cos\theta = \mathbf{n_b} \cdot \mathbf{n_s}, \tag{20}$$

where θ is an experimentally-calculated angle, $\mathbf{n_b}$ is the boundary normal, and $\mathbf{n_s}$ is the local shock normal. When using level sets, we define the shock normal as

$$\mathbf{n_s} = \frac{\nabla\phi}{|\nabla\phi|}\,. \tag{21}$$

In this work, we limit our consideration to the case $\mathbf{n_b} \cdot \mathbf{n_s} = 0$; however the approach we describe is readily extendable to more general settings.

We can integrate DSD results with hydrodynamics simulations through creation of a burn table. Its implementation is straightforward. The solution of our advection equation is $\phi(\vec{x}, t)$. With this, we can locate the burn front at any time by finding the zero contour of the level set, $\phi = 0$. Recording the time the

burn front arrives at particular points in the mesh gives us our burn table. We can do this simply by seeing if ϕ has changed sign at a node after a time step has been taken. Note that in doing this, we assume the velocity coefficient u is greater than zero, as this means the shock will cross a node only once.

6. Numerical Results

In this section we present several numerical results. The level set algorithm with modified reinitialization was implemented using the MFEM finite element software library [21]. Our choice of numerical tests is similar to those presented in [22]. The first two results are based on using a prescribed velocity field in order to test the stability and accuracy of the basic advection equation, Eq. (1). The remaining tests consider the more general case where the velocity of the front is determined by its time evolving surface normal, which requires use of the full level set evolution plus redistance solver.

6.1. Advection Only Test: Sheared Circle

Originating in [23], a circle is deformed according to the velocity field

$$\mathbf{u} = -\cos\left(\pi\frac{t}{8}\right)\left[-\sin(\pi x)\cos(\pi y), \sin(\pi y)\cos(\pi x)\right]. \tag{22}$$

The circle deforms in one direction until $t = 4$, and then rotates in the other direction until $t = 8$. At $t = 8$ the circle should have the same shape and location as the initial circle. We use cubic Q_3 finite elements on a $[0, 1] \times [0, 1]$ structured square mesh with 4,096 elements. The circle itself has radius 0.2 and is initially centered at $(0.5, 0.75)$, i.e. the level set function is

$$\phi = \sqrt{(x - 0.5)^2 + (y - 0.75)^2} - 0.2. \tag{23}$$

Results are in Fig. 2, showing the circle returning to its original location and shape.

6.2. Advection Only Test: Zalesak's Disk

This problem, proposed in [24], rotates a notched circle with velocity

$$\mathbf{u} = [-\pi y, \pi x]. \tag{24}$$

We use cubic Q_3 finite elements on a $[-1, 1] \times [-1, 1]$ structured square mesh with 9,216 elements. The notch's corners should not deform as the circle rotates and should remain sharp. Smearing of the corners may be seen with certain finite difference methods if AMR is not used (see [25] for an example). Results are in Fig. 3. The solution exhibits numerical oscillations at the circle interface, which is expected behavior for the use of high-order elements without any special (non-linear) flux correction [26].

6.3. Expanding Circle Test

In the remaining problems we use a velocity based on the surface normal $\nabla\phi/|\nabla\phi|$, with a constant magnitude u.

This simple problem consists of tracking the expansion of an initial circular level set with a starting radius of r_0, expanding outward with constant speed $u = 1$. The analytic solution to the problem for any given time t is simply an expanded circle of radius $r_0 + ut$. In this example we consider two different mesh types:

- a simple Cartesian mesh of $[0, 1] \times [0, 1]$
- an more complex unstructured, quadratically curved (Q_2) mesh

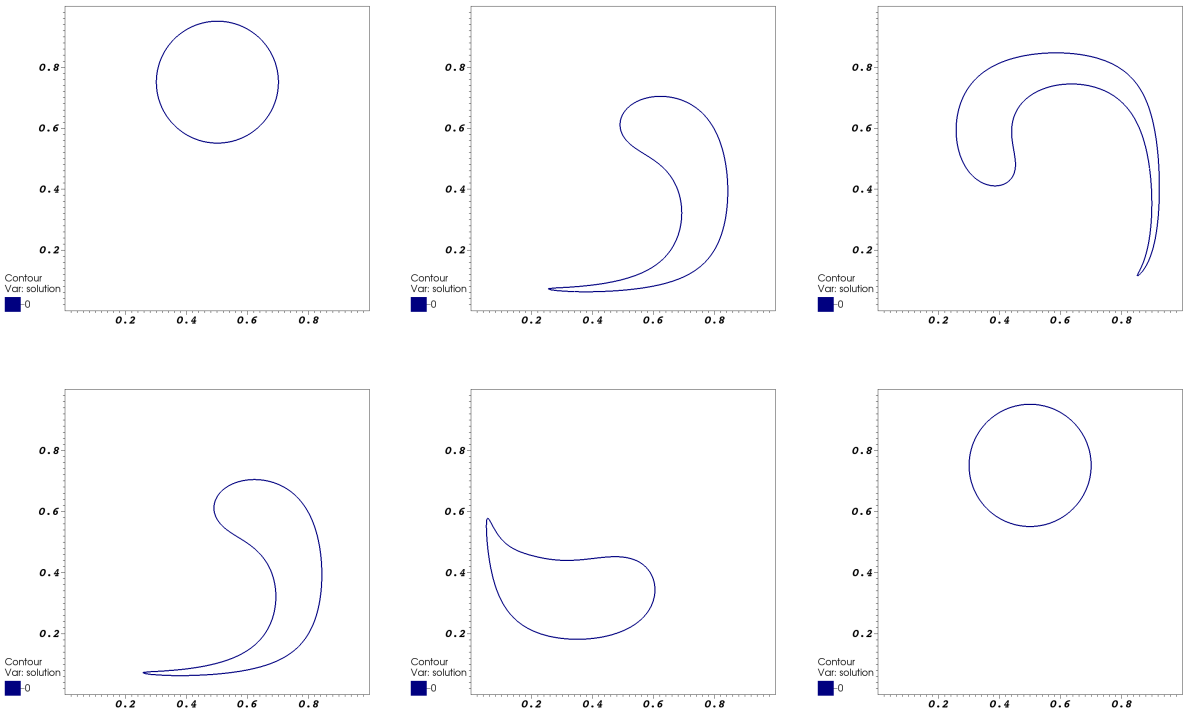


Figure 2: Deformation of circle from $t = 0$ (top left) to $t = 8$ (bottom right). In the third image, at $t = 4$, the circle reverses direction.

We show mesh convergence results for the expanding circle initialized according to the level set

$$\phi = \sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.1 \tag{25}$$

on the simple Cartesian mesh in Fig. 4 using both h and p refinement. The convergence study, which plots the L_2 error between the exact and calculated numerical solution, i.e. $\|\phi_{exact} - \phi_{num}\|_2$, is performed using both the classic redistancing method and the modified redistancing method with coefficient $\mu = 0.00025/p$, where p is the order of elements used. Convergence rates are provided in Tables 1 and 2. Note that the classic redistancing method achieves for the most part, the expected high-order convergence rates; although the $p = 4$ case is not reaching $4th$ order. The modified redistancing method exhibits some degree of high-order convergence until the error reaches a *floor* which is due to the addition of numerical diffusion characterized by the coefficient $\mu = 0.00025/p$.

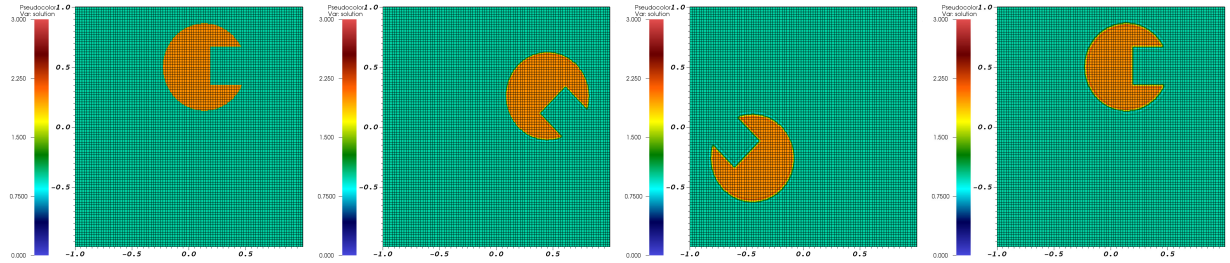
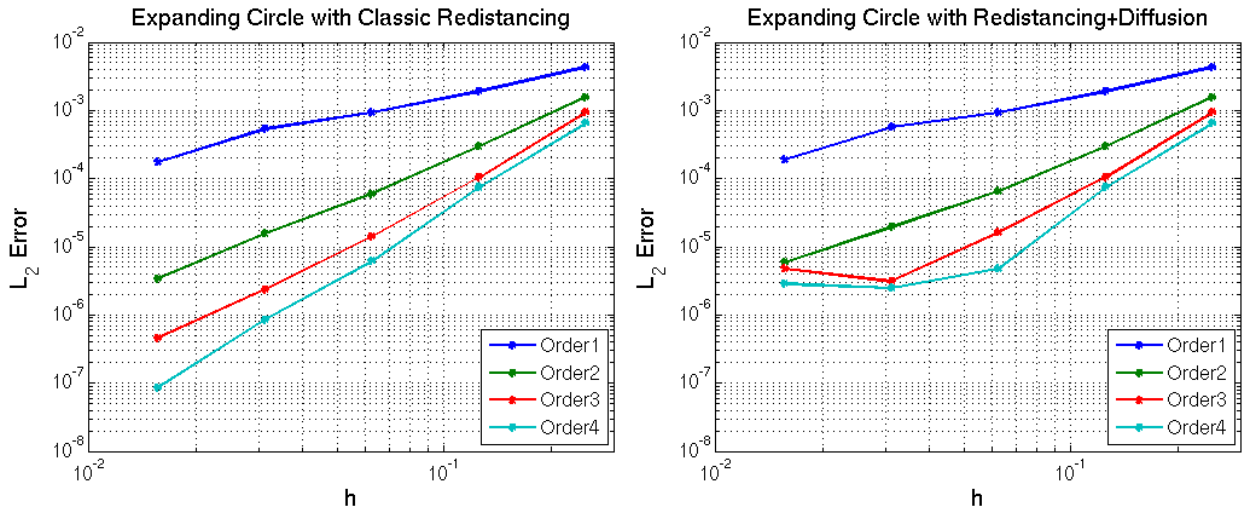


Figure 3: One clockwise rotation of notched disk (left to right). The notch maintains its integrity and does not deform as the disk rotates.



(a) Classic redistancing

(b) Modified redistancing with $\mu = 0.00025/p$

Figure 4: L_2 error over entire mesh for expanding circle on structured square.

	h=1/8	1/16	1/32	1/64
Order 1	1.164824	1.036367	0.789979	1.627847
2	2.392793	2.328633	1.921959	2.194530
3	3.176047	2.887602	2.574728	2.376827
4	3.129076	3.591215	2.857416	3.307462

Table 1: Convergence rates with classic redistancing

	h=1/8	1/16	1/32	1/64
Order 1	1.164824	1.036367	0.703796	1.597087
2	2.396982	2.179868	1.734678	1.722894
3	3.168031	2.677736	2.394737	-0.624212
4	3.124182	3.961253	0.950363	-0.205903

Table 2: Convergence rates with modified redistancing ($\mu = 0.00025/p$)

In Fig. 5 we show results of the initial level set $\phi = \sqrt{x^2 + y^2} - 0.1$ propagating outwards using Q_2

(quadratic) finite elements on the unstructured mesh. The algorithm is able to preserve the symmetry of the initial circular level set even on a highly unstructured, curved mesh.

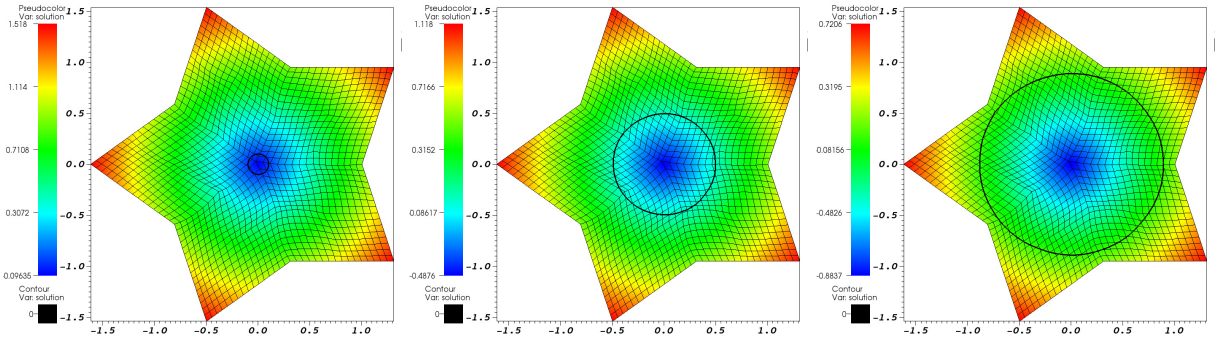


Figure 5: A circle, initialized at the star's center, stably propagated outwards (left to right).

6.4. Rate Stick

The rate stick experiment simulates a traveling high-explosive detonation front [9, 20]. Using Q_2 (quadratic) elements, we initialize the interface according to

$$\phi = \sqrt{x^2 + (y - 2.5)^2} - 0.5 \tag{26}$$

on a structured $[0, 25] \times [0, 5]$ stick with 500 elements. Once the front is propagated to the end of the stick from left to right, lighting times for hydrodynamics simulations can easily be calculated at some point $\hat{\mathbf{x}}$ and time t using $T_{light} = \phi(\hat{\mathbf{x}}, t)/u$. The front towards the end of its run is shown in Fig. 6.

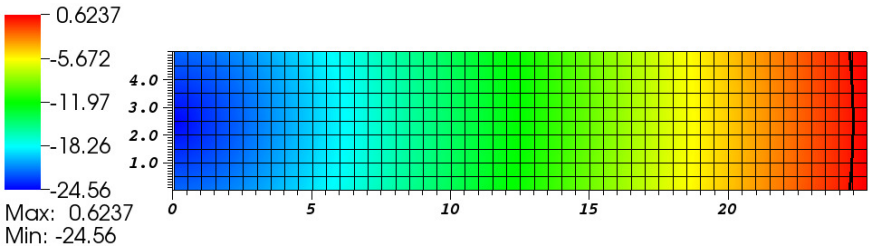


Figure 6: A detonation front (in black) is initialized as a small circle on the left edge and propagated to the end of the stick.

6.5. 3D–Expanding Sphere and Merging Spheres

Expanding Sphere [22]: We expand a sphere using Q_2 (quadratic) elements on a structured $[0, 1] \times [0, 1] \times [0, 1]$ cube with 4,096 elements. The initial level set function ϕ is given by

$$\phi = \sqrt{(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2} - 0.2. \tag{27}$$

This problem represents propagation of an idealized detonation front. Results are in Fig. 7.

Merging Spheres [22]: In this problem we initialize two spheres at two corners of a cube according to the equation

$$\phi = \begin{cases} \sqrt{x^2 + y^2 + z^2} - 0.1, & \text{if } x \leq 0.5 \\ \sqrt{(x - 1)^2 + y^2 + z^2} - 0.1, & \text{if } x > 0.5. \end{cases} \tag{28}$$

Similar to the propagation of 2D circles, the analytic solution to the problem for any given time t is simply an expanded sphere of radius $0.1 + ut$, where $u = 1$, centered at $[0, 0, 0]$ for points satisfying $x \leq 0.5$, and an expanded sphere of radius $0.1 + ut$ centered at $[1, 0, 0]$ for points satisfying $x > 0.5$. Like the expanding sphere, the merging spheres represents the propagation and subsequent collision of two idealized detonation fronts. This problem introduces an added complication, as certain finite difference methods cannot handle the discontinuous derivative along the merging fronts. Order two elements are used on the same mesh as the expanding sphere. We see smooth merging in Fig. 7.

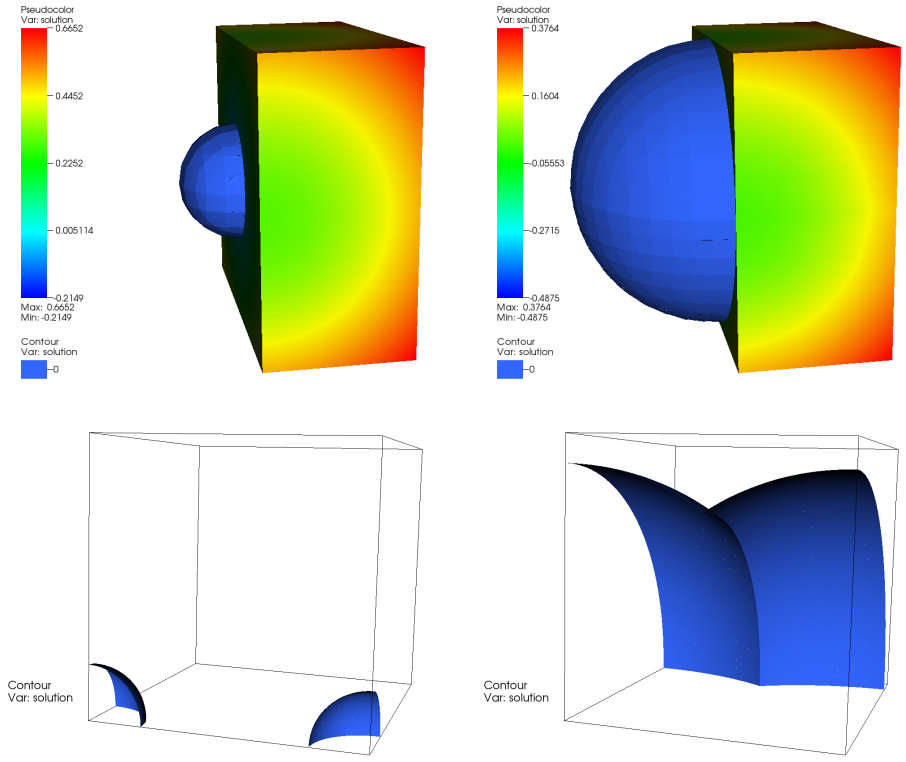


Figure 7: (top) Simulation of an idealized detonation front with an expanding sphere. (bottom) Simulation of merging detonation fronts with two expanding spheres.

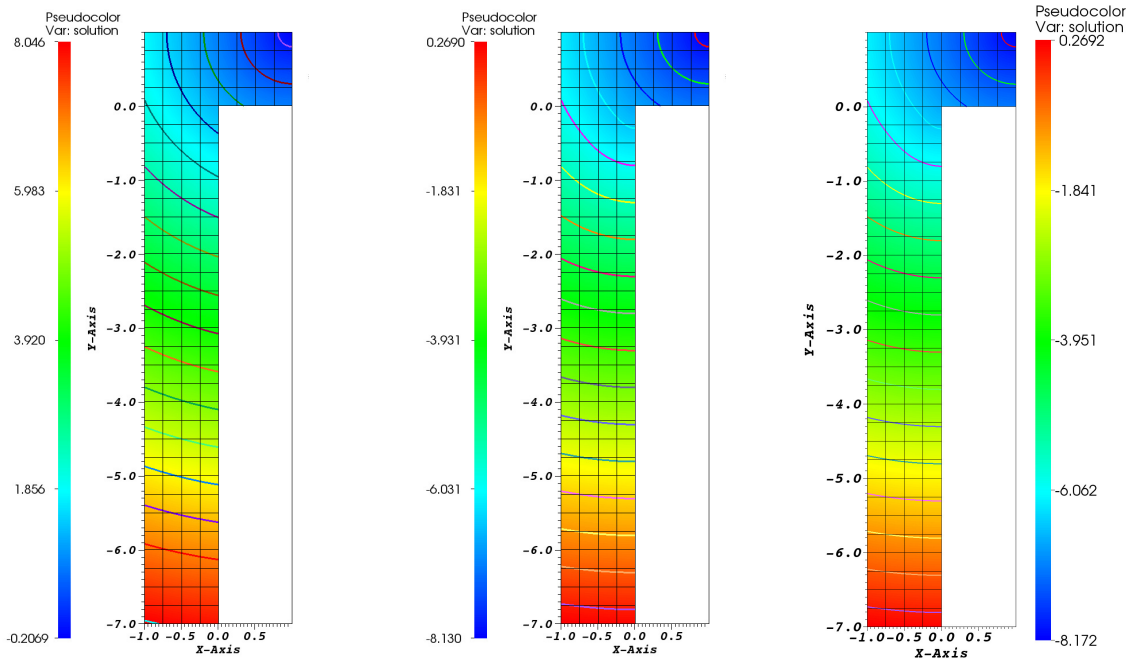
6.6. Corner Turning

The next two results test the ability of our solver to handle propagation around sharp and smooth corners.

This problem initializes a circle using

$$\phi = \sqrt{(x-1)^2 + (y-1)^2} - 0.2, \tag{29}$$

using Q_4 (4th order) finite elements on the upper right corner of an $[-1, 1] \times [-7, 1]$ L-shape mesh with 144 elements. The contours of the level set interface should properly turn the corner and form a ninety-degree angle with the boundary [27]. On this mesh, classic reinitialization may have stability issues depending on order and refinement level. We use a modified reinitialization with $\mu = 0.0005$, which we see in Fig. 8 yields stable evolution of the front.



(a) Initialized level set function (b) Final numerical solution (c) Final exact solution

Figure 8: (a) Initialization and propagation of front on an L-shape mesh. (b) Reinitialization correctly adapts and propagates level set contours around the sharp corner and compares well to final exact solution (c).

6.7. Propagation Around Circular Hole

Similar to [28], this problem, using Q_1 (linear) elements, initializes ϕ given by

$$\phi = \sqrt{(x-0.5)^2 + y^2} - 0.1, \tag{30}$$

which is centered on the bottom edge of a $[0,1] \times [0,1]$ NURBS square mesh with 4,096 elements with a hole of radius 0.2 in the middle. The contours of the level set interface should properly address the hole and exhibit appropriate bending after passing through the hole—thus maintaining the property of a signed distance function. We see our method yields stable and accurate approximation of the resulting level set field. Results are in Fig. 9.

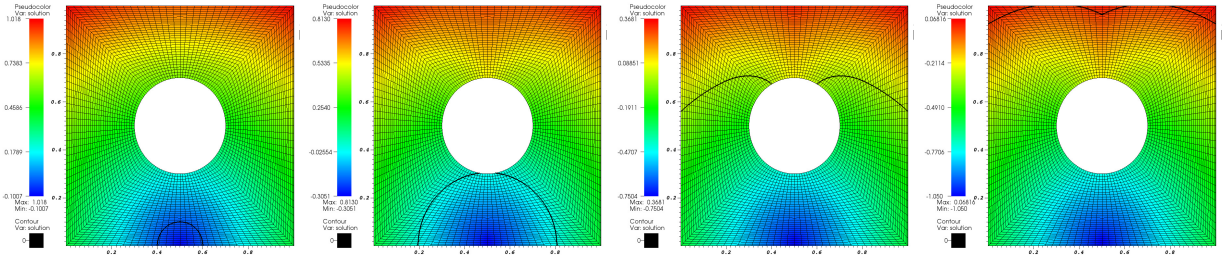


Figure 9: Propagation of a circular front around a hole of radius 0.2 (left to right).

7. Conclusion

We propose a level set approach using high-order finite elements that leads to stable front propagation in 2D and 3D on high-order, curved, unstructured meshes. To maintain accuracy during spatial discretization, the level set is periodically reinitialized using an amalgamation of the classic redistancing equation and a diffusion penalty term. This provides stability to front evolution. Applications of this approach include detonation shock dynamics (DSD) for accurately computing the propagation of a high-explosive burn front. Numerical results show stable evolution of the advection equation, proper redistancing using our reinitialization equation, and the ability to successfully navigate geometric complexities such as corners and holes.

References

[1] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1988) 12–49.

[2] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer New York, 2003.

[3] J. A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.

[4] S. Osher, R. P. Fedkiw, Level set methods: An overview and some recent results, Journal of Computational Physics 169 (2001) 463–502.

[5] J. A. Sethian, Theory, algorithms, and applications of level set methods for propagating interfaces, Acta Numerica 5 (1996) 309.

[6] S. Osher, R. Tsai, Level set methods and their applications in image science, Communications in Mathematical Sciences 1 (2003) 1–20.

[7] J. D. Anderson, Computational Fluid Dynamics, McGraw-Hill Education - Europe, 1995.

[8] S. Chen, B. Merriman, M. Kang, R. E. Caflisch, C. Ratsch, L.-T. Cheng, M. Gyure, R. P. Fedkiw, C. Anderson, S. Osher, A level set method for thin film epitaxial growth, Journal of Computational Physics 167 (2001) 475–500.

[9] T. D. Aslam, J. B. Bdzil, D. Stewart, Level set methods applied to modeling detonation shock dynamics, Journal of Computational Physics 126 (1996) 390–409.

[10] J. Bdzil, D. Stewart, T. Jackson, Program burn algorithms based on detonation shock dynamics: Discrete approximations of detonation flows with discontinuous front models, Journal of Computational Physics 174 (2001) 870–902.

[11] J. B. Bdzil, D. S. Stewart, Theory of detonation shock dynamics, in: Shock Waves Science and Technology Library, Vol. 6, Springer, 2011, pp. 373–453.

[12] S. G. Cochran, J. Chan, Shock Initiation and Detonation Models in One and Two Dimensions, Phys. Fluids 23 (1980) 2362–2372.

[13] E. L. Lee, C. M. Tarver, Phenomenological Model of Shock Initiation in Heterogeneous Explosives.

[14] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1994) 146–159.

[15] C. Li, C. Xu, C. Gui, M. Fox, Level set evolution without re-initialization: A new variational formulation, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05), Institute of Electrical and Electronics Engineers (IEEE).

[16] J. A. Sethian, P. Smereka, Level Set Methods for Fluid Interfaces, Annual Review of Fluid Mechanics 35 (1) (2003) 341–372.

[17] W. Reed, T. Hill, Triangular mesh methods for the neutron transport equation, Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM (1973).

[18] M. F. Wheeler, An elliptic collocation-finite element method with interior penalties, SIAM Journal on Numerical Analysis 15 (1978) 152–161.

[19] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Computers & Fluids 27 (1998) 663–680.

[20] J. B. Bdzil, T. D.Aslam, R. A. Catanach, L. G. Hill, M. Short, DSD Front Models: Nonideal explosive detonation in ANFO, in: Twelfth International Symposium on Detonation, San Diego, California, USA, 2002.

[21] MFEM: Modular finite element methods, `mfem.org`.

[22] N. R. Morgan, J. I. Waltz, A 3D level set approach for evolving fronts on tetrahedral meshes with adaptive mesh refinement, Journal of Computational Physics.

[23] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112–152.

[24] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of Computational Physics 31 (1979) 335–362.

[25] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A Hybrid Particle Level Set Method for Improved Interface Capturing, Journal of Computational Physics 183 (1) (2002) 83–116.

[26] R. Anderson, V. Dobrev, T. Kolev, R. Rieben, Monotonicity in high-order curvilinear finite element arbitrary Lagrangian-eulerian remap, International Journal of Numerical Methods in Fluids 77 (5) (2014) 249–273.

[27] M. Arienti, A Numerical and Analytical Study of Detonation Diffraction, Ph.D. thesis, California Institute of Technology (2002).

[28] S. Yoo, D. S. Stewart, A hybrid level set method for modelling detonation and combustion problems in complex geometries, Combustion Theory and Modelling 9 (2005) 219–254.