JOURNAL OF COMPUTATIONAL PHYSICS **126,** 390–409 (1996)
ARTICLE NO. 0145

Level Set Methods Applied to Modeling Detonation Shock Dynamics

TARIQ D. ASLAM,* JOHN B. BDZIL,† AND D. SCOTT STEWART*,[1]

University of Illinois, Urbana, Illinois 61801 and †Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Received June 26, 1995; revised November 3, 1995

We give an extension of the level set formulation of Osher and Sethian, which describes the dynamics of surfaces that propagate under the influence of their own curvature. We consider an extension of their original algorithms for finite domains that includes boundary conditions. We discuss this extension in the context of a specific application that comes from the theory of detonation shock dynamics (DSD). We give an outline of the theory of DSD which includes the formulation of the boundary conditions that comprise the engineering model. We give the formulation of the level set method, as applied to our application with finite boundary conditions. We develop a numerical method to implement arbitrarily complex 2D boundary conditions and give a few representative calculations. We also discuss the dynamics of level curve motion and point out restrictions that arise when applying boundary conditions. © 1996 Academic Press, Inc.

1. INTRODUCTION

1.1. Detonation Shock Dynamics (DSD)

Detonation shock dynamics (DSD) is an asymptotic theory that describes the evolution of a multi-dimensional, curved, near-Chapman–Jouguet (CJ) detonation shock in terms of an intrinsic evolution equation for the shock surface. A complete mathematical model of detonation [14], consists of the compressible Euler equations, an equation of state with a reaction progress variable and a reaction-rate law. These equations admit a one-dimensional (1D), steady traveling wave solution that corresponds to a detonation with a distributed, finite width reaction zone. The structure calculation for this zone consists of a system of ordinary differential equations (ODEs) that contain a critical point within the zone. These, together with the shock conditions, serve to define the normal speed of the detonation, D_{CJ}. The CJ detonation is the detonation whose speed corresponds to a sonic state at the end of the reaction-zone. The steady solution is called the ZND solution.

The shock evolution equations of DSD-theory are derived from an asymptotic theory that assumes the curved shock has a large radius of curvature compared to the characteristic, 1D reaction-zone length and that the important dynamic time scale is slow, compared to the transit time for particles through the reaction zone [1, 8]. The simplest form of the intrinsic surface evolution equation, derived from DSD-theory, obtains a relation for the normal detonation shock velocity, D_n, as a function of the local total shock curvature, $\kappa = \kappa_1 + \kappa_2$ (the sum of the principle curvatures and twice the mean curvature).

In our notation, the shock normal is chosen to point in the direction of the unreacted explosive and the curvature, κ, is defined to be positive when the shock is convex. We call this intrinsic (material dependent) relation between D_n and κ, the $D_n - \kappa$ relation. Physically, positive curvature corresponds to a *diverging* detonation in which the shock is convex shape, and D_n is below the plane CJ value, D_{CJ}, for $\kappa > 0$. When the curvature has the opposite sign, $\kappa < 0$, the shock has a concave shape and D_n lies above D_{CJ}. The physical justification for modeling the shock dynamics in such a simple way is as follows.

In the streamwise direction, the reaction zone that supports the detonation resembles the classical ZND structure. Although the reaction zone is not strictly steady for multi-dimensional detonation, it continues to have the property that the shock is only influenced by the subsonic region between the sonic curve and the detonation shock curve. This insulation of the shock from the vast region that follows the reaction zone leads, in the limit of weak shock curvature (measured relative to the distance from the shock to the sonic curve), to the result that the normal detonation speed D_n is a function of the shock curvature κ (under the assumption of sufficiently slow dynamics).

Although the shock is insulated from the far-field flow in the streamwise direction, the reaction zone provides a path by which disturbances can propagate in the direction transverse to the shock-normal direction. In particular, the disturbance generated at the edge of the explosive, where the high-pressure detonation products expand to low pressure, propagates through the reaction zone in the transverse direction, leading to a substantial decrease in the

[1] Corresponding author.

0021-9991/96 $18.00

pressure of the reaction zone, even far from the edge. More than any other influence, these lateral rarefactions from the edge of the explosive control the speed and hence the shape and location of the detonation shock.

For the purpose of this paper we assume a fundamental $D_n - \kappa$ relation exists, one that passes through $\kappa = 0$ at $D_n = D_{CJ}$ and is monotonic, for both positive and negative curvatures. Details of how to obtain a $D_n - \kappa$ relation can be found elsewhere [1, 8, 20]. The extension of the $D_n - \kappa$ for negative curvature has some experimental and numerical confirmation [4, 5]. The dynamics of this surface then is wholly described as propagating under the influence of its curvature. In [2], Osher and Sethian developed a numerical method to solve for the motion of such surfaces, originally dubbed PSC for "propagation of surfaces under curvature." Now the methods are more commonly known as level set (LS) methods. We turn to a brief description of their work next.

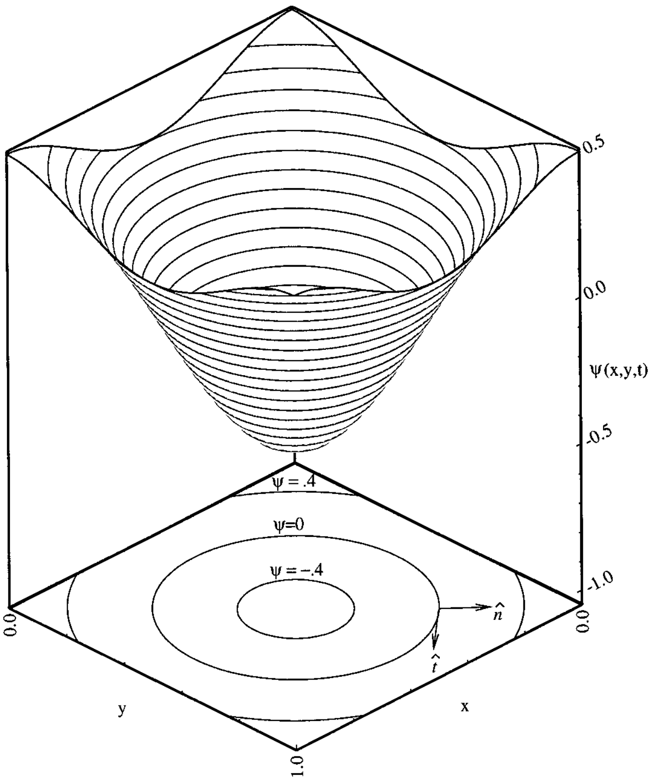

FIG. 1. Schematic of level surface and the projection of level-curves in the x, y-plane at an instant in time. Also shown are the normal and tangent to the level curve, $\psi = 0$.

1.2. The Level Set Algorithm of Osher and Sethian

Osher and Sethian [2] discussed a novel and powerful embedding concept that has an underlying simplicity for the calculations that we are concerned with in this paper and for front tracking in general. Specifically they considered the motion of a surface under the influence of a $D_n - \kappa$ relation. They pointed out some of the difficulties of attempting a numerical solution of surface dynamics that uses algorithms based on surface parameterizations. These difficulties include the corresponding loss of accuracy due to the bunching of nodes in regions where the front experiences a convergence, which results in a loss of stability of the method. Also, in regions of expansion, nodes diverge, and new nodes must be added to maintain stability. Rezoning is thus an essential feature of such methods. Furthermore, there is the logical complexity in the programming required to handle complex and, perhaps, unforeseen interactions, when sections of shock merge or break apart.

For a physical simulation that uses an underlying surface parameterization method, a separate and independent description of the topology of each disparate segment of the shock surface must be carried along with all the rules that give the details for extinguishing old segments, and creating new ones. A programmer who deals with the issues of trying to write reasonably robust code for engineering applications must confront a difficult task with these methods. These issues are especially important, when the tracking algorithm is to be used as a subroutine in part of a larger application code that solves problems with great system complexity.

The LS methods use instead a formulation where the surface of interest is embedded in a field of one higher dimension, in the physical space of the application. The surface of physical interest is found by taking a subset of the field, specifically a constant value of a field function which defines a level-contour in 2D or a level-surface in 3D. Thus for a 2D application, the level curves are embedded in 2D field, and for 3D, the level surfaces are embedded in a 3D field. In particular, one solves for the dynamics of the level curves, $\psi =$ const where all the level curves obey the $D_n - \kappa$ relation. The level curves of physical interest for the application are the ones that evolve from the initial configuration of the physical problem, where the level-constant is used to identify the physically relevant surface, during its evolution. The curve/surface of interest $\psi = 0$ is then the object of a contour search of the full field of $\psi(x, y, z, t)$.

Figure 1 shows a time snapshot of a representative 2D level surface, $\psi(x, y, t)$ and its projection onto the x, y-plane. The embedding relies on the contouring being uniquely defined, such that a single value of $\psi(x, y, t)$ is obtained for each point (x, y) at a given instant of time.

While it might seem that additional computation is required to represent a 2D surface by a solving for a 3D field, in fact, the gain in logical simplicity leads to computations that are very efficient and accurate. These advantages easily override any perceived increase in computational cost. We came to this conclusion in the course of developing numerical methods for our applications, having first used surface parameterization methods in 2D, [6] and having realized the limitations of our formulation and methods for our 3D applications.

In engineering applications for explosive materials, boundaries represent interfaces between the explosive and its confinement at the edge. The typical application has a charge of explosive material of finite dimension. At the boundaries of the charge, the explosive is adjacent to inerts or other reactive material. Detonation propagation from initiating centers into space wholly comprised of unreacted explosive is of interest, but it can be regarded as a special, ideal case. Therefore, the boundary conditions at the interfaces represent the confinement of the detonation shock since they (along with the initial conditions) determine the evolution of the detonation shock. For our purposes, the LS method must address physical boundary conditions and fit neatly into existing engineering code infrastructures.

1.3. Outline of the Paper

In Section 2, we present a self-contained discussion of the engineering application of DSD, which includes the formulation of the boundary conditions. In particular we include very specific $D_n - \kappa$ relations and boundary conditions that are relevant in explosive engineering problems. Nonetheless, our formulation here is quite general.

In Section 3, we give a brief explanation that derives the PDE for the LS function in the interior of the explosive domain, as applied to our applications with finite boundaries. We describe the numerical algorithm that is used (following Osher and Sethian for the most part) and in particular we discuss in detail the interior differencing, initial conditions, the differencing used for the boundary conditions and extensions to 3D. Also, the concept of a "burn table" is introduced, which is most useful for explosive and possibly other applications, when it is known that the front passes, at most, one time past any fixed, Eulerian point. In Section 4, we discuss the numerical stability and accuracy of the scheme.

In Section 5, we present a series of examples found in an explosives engineering problem. We examine the response of an initially planar CJ detonation to the three most common types of flows generated by the interaction of detonations with the edge of an explosive. The three problems are: (1) the sudden loss of confinement at a straight edge (referred to here as the ratestick problem); (2) the formation of a "Mach" reflection when a detonation enters a converging channel; and (3) the diffraction of a detonation produced when entering a diverging channel. Finally, in Section 6, we discuss some formal mathematical issues regarding the nature of the embedding of the level-curves and their relative motion under the action of boundary conditions.

2. DETONATION SHOCK DYNAMICS

As mentioned in the Introduction, DSD is the name given to a body of multi-dimensional theory that describes the dynamics of "near-Chapman–Jouguet" detonations. Its name follows from Whitham's theory of "geometrical shock dynamics," because of the similarity of the mathematical structure of the theories. The engineering application of DSD was originally set forth in two papers [20, 8]. The simplest result of DSD theory is that under suitable conditions, the detonation shock in the explosive propagates according to the simple formula

$$D_n = D_{CJ} - \alpha(\kappa), \tag{1}$$

where D_n is the normal velocity of the shock surface, D_{CJ} is the 1D, steady, Chapman–Jouguet velocity for the explosive, and $\alpha(\kappa)$ is a function of curvature κ that is a material property of the explosive. Figure 2 illustrates the sign of the curvature for a typical detonation shock. A sketch of a typical $D_n - \kappa$ relation is shown in Fig. 3.

2.1. Boundary Conditions

We have formulated a set of model DSD boundary conditions that involve the angle that the local shock normal, $\hat{n}_s$, makes with the outwards pointing normal vector of the boundary, $\hat{n}_b$ which we refer to as ω. Equivalently ω is the angle between the tangent to the edge and the tangent to the shock. See Fig. 4. A physical justification for the DSD angle boundary condition will be given next, followed by a summary of the model boundary conditions.

The condition to be applied depends on the flow type as witnessed by an observer riding with the point of intersection of the local shock and the edge. The boundary conditions are formulated by an analysis of the local singularities admitted by the Euler equations [21] and the results are summarized in this section. The flow type is characterized by the local sonic parameter, $\mathscr{S}$, evaluated at the shock in the detonation reaction-zone and as measured by an observer moving with the point of intersection of the detonation shock and the material interface

$$\mathscr{S} \equiv C^2 - (U_n)^2 - D_n^2 \cot^2(\omega), \tag{2}$$

where C is the sound speed in the explosive, U_n is the explosive particle velocity in the shock-normal direction, and D_n is the detonation normal speed. When $\mathscr{S} < 0$, the flow is locally supersonic at the edge and no boundary condition is applied. The application of *no* boundary condition is, in practice, the application of a continuation boundary condition, where information flows from the interior to the exterior of the domain. More will be said about the numerical implementation of the continuation boundary

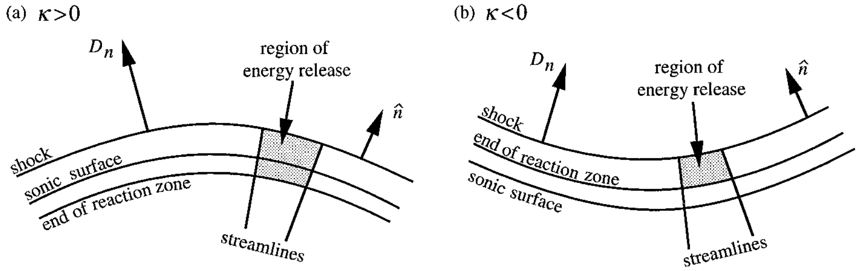

FIG. 2. A snapshot of the x, y-plane, showing a diverging and a converging detonation. For a diverging detonation, the transverse dimension of the region of chemical-energy release is smaller than the dimension of the region of shock surface that it supports (the detonation speed falls below D_{CJ}). For a converging detonation the reverse is true and the detonation speed exceeds D_{CJ}.

condition in Section 3.3. When $\mathscr{S} > 0$, the flow is locally subsonic and the presence of the edge influences the reaction zone. The form of the boundary condition for the $\mathscr{S} > 0$ case is determined by the properties of the inert material that is adjacent to the explosive.

The problem geometry and the various cases, supersonic, sonic, and subsonic, that are modeled correspond to a steady flow in the reference frame of the shock/edge intersection point. Figures 5–7 show instantaneous time snapshots of the interaction between the explosive and inert. The explosive induces a shock into the inert (labeled *inert shock*), which typically generates a reflected wave into the explosive (labeled either the *reflected shock* or the *limiting characteristic*, depending on whether the reflected wave is a shock or a rarefaction, respectively).

Figure 5 corresponds to a supersonic flow, $\mathscr{S} < 0$. As previously mentioned, no boundary condition is applied irrespective of the degree of confinement that the inert provides to the explosive. The shock reflected into the explosive does not influence the detonation shock. As the angle ω is increased to the value ω_s, where $\mathscr{S} = 0$, the flow in the explosive turns sonic and therefore can sense the degree of confinement that the adjacent inert provides. Note that ω_s is a constant in our model, given by the explosive equation of state.

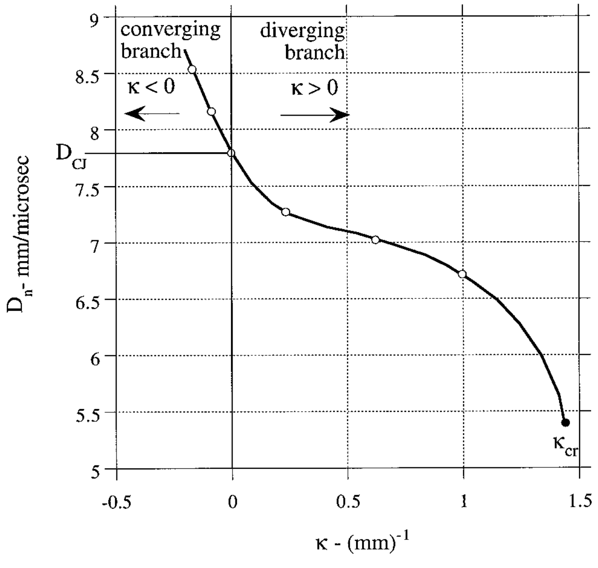

FIG. 3. The $D_n - \kappa$ relation for a typical condensed phase explosive after Bdzil *et al.*'s calibration of PBX9502 [4].

Figure 6 shows two cases, labeled as 1 and 2, that correspond to different degrees of confinement provided by the inert. For these cases, the pressure decreases towards the right of the explosive sonic locus. Case 1 corresponds to weak confinement, for which the pressure induced in the inert is considerably below the detonation pressure at the edge. The influence of the confinement propagates in the explosive no farther to the left than the limiting characteristic labeled 1. The subsonic part of the reaction zone remains totally unaffected by the confinement and the flow remains *sonic* at the shock/edge intersection point. The detonation propagates as if it were totally unconfined.

As the degree of confinement is increased further, the drop in pressure in going from the explosive to the inert becomes less, until at some critical degree of confinement the influence of the inert extends up to the limiting characteristic labeled 2. At this critical degree of confinement, the

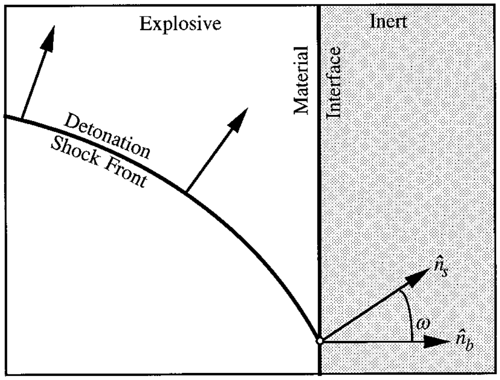

FIG. 4. Definition of the angle, ω, and the normals, $\hat{n}_s$ and $\hat{n}_b$.

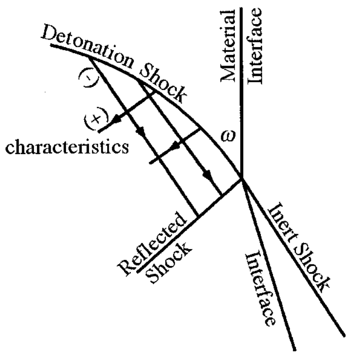

FIG. 5. DSD boundary conditions. A snapshot of the x, y-plane showing the *supersonic* type of explosive/inert boundary interaction. The magnitude of ω controls the type of interaction that occurs. Figure 5 corresponds to a supersonic flow in the explosive, measured relative to an observer riding with the shock/edge intersection point.

detonation continues to propagate as if it were unconfined. Any further *increase* in the confinement destroys the sonic isolation of the reaction zone from the influence of the confinement and leads to the case shown in Fig. 7.

If for the angle ω_s, corresponding to $\mathscr{S} = 0$, the pressure induced in the confining inert part is greater than the pressure in the explosive, then the flow that develops is that shown in Fig. 7. The reflected wave can now enter into the subsonic part of the reaction zone. This results in an increase in pressure in the reaction zone and the concomitant increase of the normal shock velocity, D_n. The angle ω increases until the pressure in the inert and reaction zones balance. Since the flow in the explosive is subsonic, a reflected shock is not generated in the explosive. The value of ω at the point of pressure equilibrium is ω_c. The value of ω_c is a constant that depends only on the specific explosive/inert pair. It is easily calculated from a shock polar analysis, assuming no reflected wave in the explosive.

2.2. Summary

In summary, the boundary interaction has the following properties: (i) When the flow in the explosive is supersonic (i.e., $\omega < \omega_s$), the continuation (outflow) boundary condition is applied. This corresponds to extrapolating the front to the exterior, without changing the angle at the boundary. (ii) When the flow turns sonic $\omega = \omega_s$, two cases can arise: (a) The pressure induced in the inert is below that immediately behind the detonation shock and the confinement has no influence on the detonation. The sonic boundary condition is applied, $\omega = \omega_s$; (b) The pressure induced in the inert is above that immediately behind the detonation shock. The angle ω increases (i.e., $\omega > \omega_s$) until the pressure in the inert and explosive are equilibrated. This angle $\omega = \omega_c$ is the equilibrium value for the angle and is regarded as a material constant that is a function of the explosive/inert pair. Thus the boundary condition recipe can be summarized as follows: (1) A continuation boundary condition is applied for supersonic flows and (2) when the flow becomes either sonic or subsonic, ω is bounded from above by a critical angle ω_c (unique for each explosive/inert pair) that is determined using the above discussion.

Figure 8 shows a time history of the evolution of the angle $\omega(t)$ along the edge of confinement that corresponds to a typical application. Figure 8a shows a detonation interacting with an edge at three different times, t_1, t_2, t_3. At time t_1, the shock/edge intersection is highly oblique and the supersonic (continuation) boundary condition applies. At time t_2, it is assumed that the intersection angle first becomes sonic, $\omega = \omega_s$. If the confinement is heavy enough,

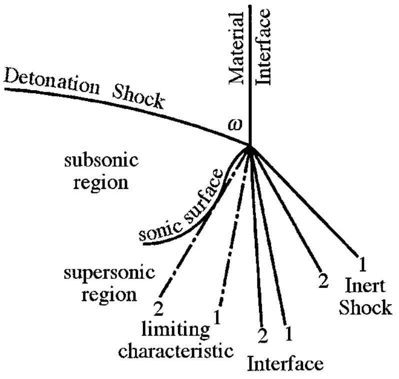

FIG. 6. DSD boundary conditions. A snapshot of the x, y-plane showing the *sonic* type of explosive/inert boundary interaction. The magnitude of ω controls the type of interaction that occurs. Figure 6 corresponds to a sonic flow in the explosive, measured relative to an observer riding with the shock/edge intersection point.

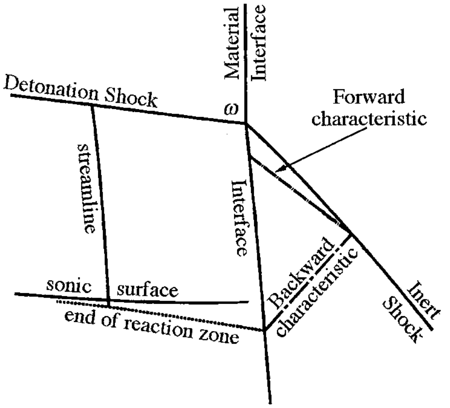

FIG. 7. DSD boundary conditions. A snapshot of the x, y-plane showing the *subsonic* type of explosive/inert boundary interaction. The magnitude of ω controls the type of interaction that occurs. Figure 7 corresponds to a subsonic flow in the explosive, measured relative to an observer riding with the shock/edge intersection point.

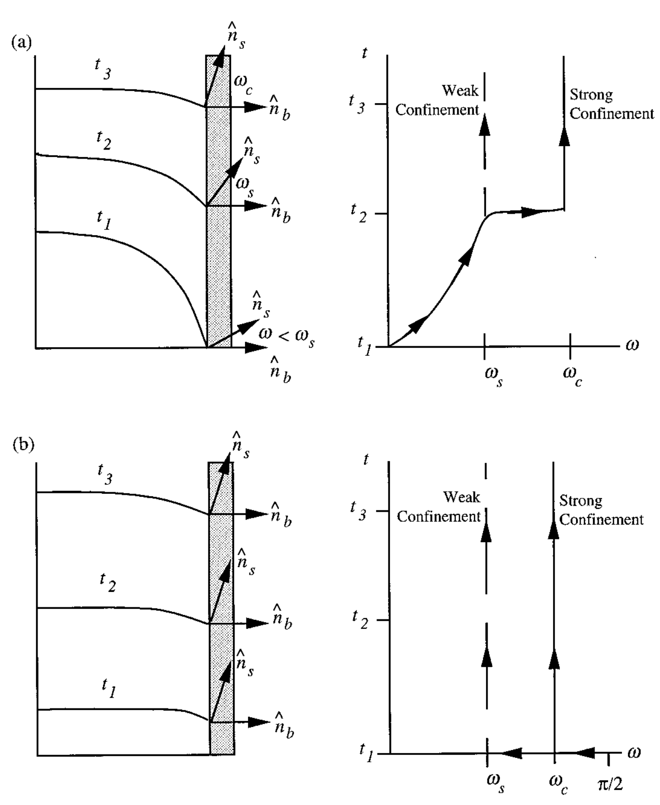

FIG. 8. Time histories of shock/edge interactions for typical (a) oblique interactions and (b) normal interactions.

a rapid acoustic transient can take place and a rapid adjustment to the equilibrium value, ω_c, can occur. After that adjustment, shown at t_3 (say), the angle remains at $\omega = \omega_c$ which corresponds to that for the explosive/confinement pair. The right-hand portion of Fig. 8a shows the time history of the shock interaction at the edge. The value of $\omega(t)$ is determined by the solution for $\omega < \omega_s$. Once ω_s is attained, a rapid jump to ω_c occurs and from then on $\omega = \omega_c$ applies. This is shown in the right-hand portion of the figure. If the confinement were sufficiently weak, no jump to ω_c would be needed, and the angle would simply remain at ω_s. This is shown by the broken line.

Figure 8b shows a different scenario. It is assumed that the detonation is initially flat and $\omega = \pi/2$. For heavy confinement, a rapid acoustic transition to $\omega = \omega_c$ is assumed to occur and maintained from then on. If the confinement is sufficiently light, then the transition is from $\omega = \pi/2$ to $\omega = \omega_s$. Again this is shown in the broken line.

3. THE LEVEL SET METHOD AND NUMERICAL IMPLEMENTATION

Here we outline the LS method, explain its application and utility as a tool for computing the dynamics of propagating interfaces, and explain the numerical method used to solve the resulting partial differential equation (PDE).

First, notice that a surface (or the shock in DSD) is a subset with a dimension one lower than the space it travels in. The LS method with applied boundary conditions solves for a field function $\psi(x, y, z, t)$ that depends on physical space and time, and the field identifies surfaces of constant values of ψ. The surface $\psi(x, y, z, t) = 0$, is typically identified with the surface of physical interest. Therefore, the computational task involves computing a field in space-time, and then exhibiting the surface of interest by searching for the special surface $\psi = 0$. Since a level curve is given by $\psi(x, y, z, t) = \text{const}$, it follows that its total derivative is zero, i.e.,

$$\frac{\partial\psi}{\partial t} + \frac{\partial\psi}{\partial x}\frac{dx}{dt} + \frac{\partial\psi}{\partial y}\frac{dy}{dt} + \frac{\partial\psi}{\partial z}\frac{dz}{dt} = 0,$$

where the time derivatives, dx/dt and so on, are the components of the surface velocity $\mathbf{D}$, defined by that particular level curve. In coordinate independent form the above equation is

$$\frac{\partial\psi}{\partial t} + \nabla\psi \cdot \mathbf{D}(\kappa) = 0. \tag{3}$$

We choose the outward surface normal $\hat{n}$ to be positive in the direction of *outward* propagation. (In our physical application the detonation shock propagates from the burnt explosive towards the unburnt explosive and the positive normal points into the unburnt material.) In terms of the LS function, the normal is given by $\hat{n} = \nabla\psi/|\nabla\psi|$. The total curvature satisfies the relation

$$\kappa \equiv \kappa_1 + \kappa_2 = \nabla \cdot \hat{n}. \tag{4}$$

Using $\mathbf{D} \cdot \hat{n} = D_n$ and $\nabla\psi \cdot \hat{n} = |\nabla\psi|$ in (3) obtains a Hamilton–Jacobi-like equation for the LS function that we mainly use in the following discussions,

$$\frac{\partial\psi}{\partial t} + D_n(\kappa)|\nabla\psi| = 0. \tag{5}$$

The curvature κ is simply related to the level set field by using the definition of the curvature from (4) and by then carrying out the indicated differentiations. For example, for two dimensions and for Cartesian coordinates, the curvature is given by

$$\kappa = \frac{\psi_{xx}\psi_y^2 - 2\psi_{xy}\psi_x\psi_y + \psi_{yy}\psi_x^2}{(\psi_x^2 + \psi_y^2)^{3/2}}. \tag{6}$$

In summary, the shock (i.e., the surface of physical interest) is assigned the level $\psi = 0$, while the unburnt material has $\psi > 0$ and the burnt material has $\psi < 0$. A unique way to specify ψ initially is to choose $\psi =$ signed minimum distance from the initial shock surface. Equation (5) is then a partial differential equation for the LS function ψ that is to be solved subject to its initial data.

The solution of the PDE with initial and boundary conditions, generates the field $\psi(x, y, z, t)$, and the location of the shock is then simply found by a search for the level surface $\psi = 0$. This is easily done by creating a table of arrival times of the shock across the computational grid. We call this the burn table. Numerically generating a burn table will be discussed in Section 3.5.

3.1. Interior Differencing

Here we give a brief description of the numerical method we use for solving the LS equation (5) on a fixed Eulerian finite difference grid. For the interior algorithm, we follow Osher and Sethian [3]. The time advance of the LS equation

$$\frac{\partial \psi}{\partial t} + D_{CJ}|\nabla\psi| - \alpha(\kappa)|\nabla\psi| = 0 \tag{7}$$

is operator split into two steps. First, ψ is advanced using the suboperator, L_{P}, defined by the first and third terms in Eq. (7). This is then followed by the advance for the suboperator, L_{H}, defined by the first and second terms in Eq. (7). The motivation for this operator splitting is related to the fact that L_{H} is a hyperbolic operator and L_{P} is a "nearly parabolic" operator. We consider these issues more fully in Section 6. Different numerical methods are thus appropriate for these different type operators. The differencing for each of the three terms in (7) is now considered separately.

For the time derivative, we use simple, first-order, forward Euler differencing

$$\frac{\partial \psi}{\partial t} = \frac{\psi_{i,j}^{n+1} - \psi_{i,j}^{n}}{\Delta t}, \tag{8}$$

where i and j represent the x and y nodes and n represents the time level in the usual way. Higher order Runge–Kutta type schemes can be used and have been derived in [2, 3].

The first-order spatial derivatives in the second term in (7) are calculated using a combination of upwinding and essentially nonoscillatory (ENO) interpolation. In the following text, first-order interpolation is equivalent to first-order differencing and second-order interpolation is equivalent to second-order differencing. Let us consider first a 2D problem using upwinding and first-order interpolation. We need to approximate $|\nabla\psi|$, and, thus, ψ_x and ψ_y. First, we construct four linear interpolants between node $\psi_{i,j}^n$ and its four surrounding nodes, $\psi_{i+1,j}^n$, $\psi_{i-1,j}^n$, $\psi_{i,j+1}^n$, and $\psi_{i,j-1}^n$. Define the usual forward and backward difference operators

$$D_x^+\psi_{i,j}^n = \frac{\psi_{i+1,j}^n - \psi_{i,j}^n}{\Delta x}, \quad D_x^-\psi_{i,j}^n = \frac{\psi_{i,j}^n - \psi_{i-1,j}^n}{\Delta x},$$

$$D_y^+\psi_{i,j}^n = \frac{\psi_{i,j+1}^n - \psi_{i,j}^n}{\Delta y}, \quad D_y^-\psi_{i,j}^n = \frac{\psi_{i,j}^n - \psi_{i,j-1}^n}{\Delta y}.$$

Next we combine these differences to define the first-order upwind difference,

$$\begin{aligned}|\nabla\psi| = [&f_x^+(D_x^+\psi_{i,j}^n) + f_x^-(D_x^-\psi_{i,j}^n)\\ &+ f_y^+(D_y^+\psi_{i,j}^n) + f_y^-(D_y^-\psi_{i,j}^n)]^{1/2},\end{aligned} \tag{9}$$

where

$$f_x^+(a) = \begin{cases} a^2, & \text{if } D_x^+\psi_{i,j}^n < 0 \\ 0, & \text{otherwise;} \end{cases} \qquad f_x^-(a) = \begin{cases} a^2, & \text{if } D_x^-\psi_{i,j}^n > 0 \\ 0, & \text{otherwise;} \end{cases}$$

$$f_y^+(a) = \begin{cases} a^2, & \text{if } D_y^+\psi_{i,j}^n < 0 \\ 0, & \text{otherwise;} \end{cases} \qquad f_y^-(a) = \begin{cases} a^2, & \text{if } D_y^-\psi_{i,j}^n > 0 \\ 0, & \text{otherwise.} \end{cases}$$

To achieve second-order spatial accuracy, a quadratic interpolant with three nodes is used. For each of the four directions, there are two choices for the interpolant. For example, consider the linear interpolant between $\psi_{i,j}^n$ and $\psi_{i+1,j}^n$. To construct a quadratic interpolant, another node, either $\psi_{i-1,j}^n$ or $\psi_{i+2,j}^n$, is used. The choice is made by picking the node which gives the smallest second derivative in magnitude. If the second derivatives are of opposite sign, then the second-order correction is taken to be zero. This same procedure is used in the other three directions resulting in the second-order scheme

$$\begin{aligned}|\nabla\psi| = \Bigg[&f_x^+\left(D_x^+\psi_{i,j}^n - \frac{\Delta x}{2}\min\mathrm{mod}(D_x^-D_x^+\psi_{i,j}^n, D_x^+D_x^+\psi_{i,j}^n)\right)\\ &+ f_x^-\left(D_x^-\psi_{i,j}^n + \frac{\Delta x}{2}\min\mathrm{mod}(D_x^-D_x^-\psi_{i,j}^n, D_x^+D_x^-\psi_{i,j}^n)\right)\\ &+ f_y^+\left(D_y^+\psi_{i,j}^n - \frac{\Delta y}{2}\min\mathrm{mod}(D_y^-D_y^+\psi_{i,j}^n, D_y^+D_x^+\psi_{i,j}^n)\right)\\ &+ f_y^-\left(D_y^-\psi_{i,j}^n + \frac{\Delta y}{2}\min\mathrm{mod}(D_y^-D_y^-\psi_{i,j}^n, D_y^+D_y^-\psi_{i,j}^n)\right)\Bigg]^{1/2},\end{aligned} \tag{10}$$

where the min mod function is defined by

$$\min \mathrm{mod}(a, b) = \begin{cases} a, & \text{if } |a| \le |b|, ab > 0 \\ b, & \text{if } |b| < |a|, ab > 0 \\ 0, & \text{otherwise.} \end{cases}$$

The third term in (7) is essentially a diffusion term, and we use second-order central differences to calculate κ and, thus, $\alpha(\kappa)$. Central differences are also used to calculate $|\nabla\psi|$ in this term.

3.2. Initial Conditions

The LS function, ψ, must be defined initially at $t = 0$, where $\psi(x, y, t = 0) = 0$ represents the initial shock locus. We choose $\psi(x, y, t = 0)$ to be the signed distance from the initial shock locus, with $\psi(x, y, t = 0)$ positive in the unburnt material and $\psi(x, y, t = 0)$ negative in the burnt material. Thus the normal, $\hat{n}$, points into the unburnt material. For example, two initially expanding cylindrical shocks with radii $= r$ located at (x_1, y_1) and (x_2, y_2) would be given by

$$\psi(x, y, t = 0) = \min[\sqrt{(x - x_1)^2 + (y - y_1)^2} - r, \\ \sqrt{(x - x_2)^2 + (y - y_2)^2} - r],$$

while two collapsing cylindrical shocks at the same location and radii would be given by

$$\psi(x, y, t = 0) = \max[r - \sqrt{(x - x_1)^2 + (y - y_1)^2}, \\ r - \sqrt{(x - x_2)^2 + (y + y_2)^2}].$$

3.3. Boundary Conditions

Three types of boundary conditions have been implemented into our LS formulation. These are symmetric (perfectly reflecting), non-reflecting (inflow/outflow), and angle (physical) boundary conditions. The formulation uses two levels of ghost nodes to enforce the particular boundary conditions. The symmetric boundary condition is trivially satisfied by reflecting the values of ψ from the interior to the exterior. For example if $x = 0$ is a symmetry plane and $\psi^n_{0,j}$ is at $x = 0$, then $\psi^n_{-1,j} = \psi^n_{1,j}$ and $\psi^n_{-2,j} = \psi^n_{2,j}$.

The non-reflecting boundary conditions are applied by using quadratic extrapolation. This is equivalent to keeping the second derivative along the normal to the boundary as a constant. The upwinded first-order spatial derivatives do not need to have ghost nodes, since they look in the proper direction. However, ghost nodes are used in the calculation of the second-order derivatives and the curvature at the boundary. For example, if non-reflecting boundary conditions are applied at $x = 0$, then $\psi^n_{-1,j} = 3\psi^n_{0,j} - 3\psi^n_{1,j} + \psi^n_{2,j}$, etc.

3.3.1. *Implementation of Angle Boundary Conditions*

Of the three boundary conditions, angle boundary conditions need the most attention. A class of physical boundary conditions within DSD theory concerns detonation waves interacting with inert boundaries were described in Section 2.1. For each inert–explosive pair, two angles are needed to define the boundary conditions at an interface. These are the sonic angle, ω_s, and the steady state angle, ω_c.

In general, the location of the inert–explosive interface, where angle boundary conditions need to be applied, can be quite complex. Unfortunately, it is not always simple to find a computational grid (body-fitted grid) whose boundaries coincide with the physical inert–explosive interface. Next we develop an internal boundary (IB) method to numerically treat these boundary conditions for arbitrarily complex interfaces on a uniform ($\Delta = \Delta x = \Delta y$) 2D Cartesian grid. In spirit, this method is similar to the Cartesian boundary method of Leveque [16] and others [17, 18], although the mathematical boundary conditions being applied are quite different. It will be shown that angle boundary conditions involve spatial derivatives of the LS function, ψ (which are similar to Neumann boundary conditions.) The mathematical boundary conditions, and the corresponding numerical implementation are given next.

First, define a new (non-evolving) level set function, $\phi(x, y)$, such that $\phi(x, y) = 0$ at the inert–explosive interface. The function $\phi(x, y)$ is defined at computational grid points as $\phi_{i,j}$ (where again i and j correspond to the x-location and y-location, respectively). We define ϕ to be the signed distance function from the inert–explosive interface, with ϕ negative in the explosive and ϕ positive in the inert. To enforce the angle boundary conditions on the interior of the computational domain, an array of (i, j) nodes near $\phi = 0$ will be used. We call this array of nodes the internal boundary (IB) nodes. These IB nodes are found in the following manner. Sweep through the grid, and if at a (i, j) node $\phi_{i,j} > 0$ and if at any of the eight surrounding nodes one of the following conditions is true, $\phi_{i+1,j} \le 0$, $\phi_{i-1,j} \le 0$, $\phi_{i,j+1} \le 0$, $\phi_{i,j-1} \le 0$, $\phi_{i+1,j+1} \le 0$, $\phi_{i-1,j-1} \le 0$, $\phi_{i-1,j+1} \le 0$, or $\phi_{i+1,j-1} \le 0$, then the (i, j) node is an IB node. This is analogous to computationally finding the $\phi = 0$ contour. This search for internal boundary points is only done once at the beginning of the computation. The angle boundary conditions will be enforced by specifying $\psi_{i,j}$ at these IB nodes. Furthermore, the interior differencing of Section 3.1 only needs to be applied at nodes where $\phi_{i,j} \le 0$, since the others correspond to inert regions.

The inert–explosive interface normal, $\hat{n}_b$, at an IB node is given by

$$\hat{n}_b = n_{bx}\hat{\imath} + n_{by}\hat{\jmath} = \frac{\nabla\phi}{|\nabla\phi|}, \tag{11}$$

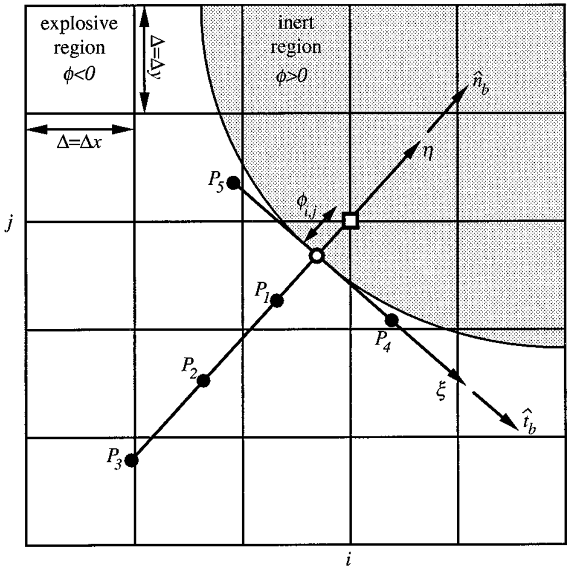

FIG. 9. Schematic of internal boundary condition stencil: ● interpolated stencil points; □ internal boundary node (i, j); ○ point where boundary condition is to be applied.

which is approximated by second-order central differences at IB nodes. For each IB node, a locally rotated orthogonal stencil is defined which is lined up with the inert–explosive interface normal, $\hat{n}_b$, and inert–explosive interface tangential unit vector, $\hat{t}_b = n_{by}\hat{\imath} - n_{bx}\hat{\jmath}$. The coordinates associated with the $\hat{n}_b$ and $\hat{t}_b$ directions are η and ξ, respectively. See Fig. 9. Since, the angle boundary condition will involve spatial derivatives of the LS function, ψ, we need to know values of ψ at the discrete points, labeled P_k, associated with each IB node. These points are given by

$$P_1 = (\phi_{i,j} - \Delta)\hat{n}_b, \quad P_2 = (\phi_{i,j} - 2\Delta)\hat{n}_b,$$
$$P_3 = (\phi_{i,j} - 3\Delta)\hat{n}_b, \quad P_4 = \Delta\hat{t}_b \quad P_5 = -\Delta\hat{t}_b.$$

Values of ψ at these rotated stencil points, P_1, P_2, P_3, P_4, P_5, are given by second-order accurate bilinear interpolation. At every timestep, the following algorithm is applied:

Step 1: Quadratically extrapolate ψ from the interior to the IB nodes along the $\hat{n}_b$ direction.

Step 2: Check if interaction at each IB node is subsonic or supersonic.

Step 3: Apply angle boundary condition to all IB nodes which have a subsonic interaction.

Quadratic extrapolation is accomplished by solving

$$\psi_{i,j} = 3\psi_{P_1} - 3\psi_{P_2} + \psi_{P_3} \tag{12}$$

at all IB nodes. In general, $\psi_{P_1}, \psi_{P_2}, \psi_{P_3}$ can be dependent on IB nodes. For example, in Fig. 9, ψ_{P_1} will be a linear combination of the three interior values $\psi_{i-1,j-1}$, $\psi_{i-1,j}$, $\psi_{i,j-1}$ and the IB node value $\psi_{i,j}$. Therefore, (12) will result in a system of linear equations, where the number of equations and unknowns is equal to the number of IB nodes. This system is solved by the following iterative method: View (12) as $\psi_{i,j} = F_1(\psi_{i,j})$. Start with an initial guess for each $\psi_{i,j}^{\text{guess}}$, say the value of $\psi_{i,j}$ at the old time step. Evaluate $F_1(\psi_{i,j}^{\text{guess}})$, and set

$$\psi_{i,j}^{\text{new}} = (1 - w)\psi_{i,j}^{\text{guess}} + wF_1(\psi_{i,j}^{\text{guess}}), \tag{13}$$

where $w < 1$ for the iterative method to converge. Repeat (13) until $\max(|\psi_{i,j}^{\text{new}} - \psi_{i,j}^{\text{guess}}|) < \varepsilon\Delta$. The values $w = 0.9$ and $\varepsilon = 10^{-3}$ work well and typically converge in 10 iterations or less. Note that the number of equations being solved iteratively is of the order $(N_x N_y)^{1/2}$, where N_x and N_y are the number of x and y grid points, so the algorithm is relatively inexpensive, compared to the interior scheme.

To check if an interaction at an IB node is subsonic or supersonic, an approximation for the angle, ω, between the shock normal, $\hat{n}_s$, and the inert–explosive interface normal, $\hat{n}_b$, is needed. The vector, $\hat{n}_b$, is given from (11) and the normal $\hat{n}_s$ is given by

$$\hat{n}_s = \frac{\psi_\eta}{\sqrt{\psi_\eta^2 + \psi_\xi^2}}\hat{n}_b + \frac{\psi_\xi}{\sqrt{\psi_\eta^2 + \psi_\xi^2}}\hat{t}_b$$

and therefore ω is given by

$$\cos\omega = \hat{n}_s \cdot \hat{n}_b = \frac{\psi_\eta}{\sqrt{\psi_\eta^2 + \psi_\xi^2}}. \tag{14}$$

Approximations to the derivative terms in (14) are needed at the point where the boundary condition is to be applied, see Fig. 9. Taylor series expansions reveal the approximation

$$\psi_\eta = \frac{\psi_{P_2} - 4\psi_{P_1} + 3\psi_{i,j}}{2\Delta} - \frac{\psi_{P_2} - 2\psi_{P_1} + \psi_{i,j}}{\Delta^2}\phi_{i,j}, \tag{15}$$

where $\phi_{i,j}$ appears in (15) since it is the signed distance from the node (i, j) to the location where the boundary condition is to be applied. See Fig. 9. A central difference approximation to ψ_ξ is

$$\psi_\xi = \frac{\psi_{P_4} - \psi_{P_5}}{2\Delta}. \tag{16}$$

Equation (14), together with (15) and (16), gives an approximation to $\cos\omega$ at the boundary point corresponding to the IB node. If $\cos\omega > \cos\omega_s$ then the interaction is supersonic, else the interaction is subsonic.

All IB nodes, which have subsonic interaction, must have the angle at the boundary point set to $\omega = \omega_c$. Therefore we need to solve

$$\cos\omega_c = \hat{n}_b \cdot \hat{n}_s = \frac{\psi_\eta}{\sqrt{\psi_\eta^2 + \psi_\xi^2}}.$$

Solving for the derivative, ψ_η, yields

$$\psi_\eta = \cos\omega_c(\psi_\xi^2 \csc^2\omega_c)^{1/2}. \qquad (17)$$

Substitution of (15) and (16) into (17), and solving for $\psi_{i,j}$ yields

$$\psi_{i,j} = \frac{\begin{array}{l}\cos\omega_c((\psi_{P_4} - \psi_{P_5})^2 \csc^2\omega_c)^{1/2} \\ \quad - \Delta^2(\psi_{P_2} - 4\psi_{P_1}) + \Delta\phi_{i,j}(2\psi_{P_2} - 4\psi_{P_1})\end{array}}{3\Delta^2 - 2\Delta\phi_{i,j}}. \qquad (18)$$

Now, the values $\psi_{P_1}, \psi_{P_2}, \psi_{P_4}, \psi_{P_5}$ appear on the right-hand side of (18) in a nonlinear way. But this system of nonlinear equations can be solved by viewing (18) as $\psi_{i,j} = F_2(\psi_{i,j})$ and applying the same iterative technique as before (but with F_1 replaced by F_2.)

3.4. Extensions to Three Dimensions

Extensions of the LS method described in the previous sections to three dimensions is relatively straightforward. Since each term in the hyperbolic part is treated separately (i.e., approximations to ψ_x, ψ_y, and ψ_z), only an additional term in the approximation to $|\nabla\psi|$ will be needed. The parabolic terms in the LS formulation in three dimensions can again be calculated using second-order central differences, just as in two dimensions. Using the signed distance function as initial conditions works in three dimensions as well. Reflecting boundary conditions are simply applied in three dimensions. Non-reflecting boundary conditions are also easily applied by using quadratic extrapolation in the interface normal, $\hat{n}_b$, direction. The same methodology of 3.3.1 can be applied in 3D to enforce arbitrarily complex boundaries.

3.5. Creating a Burn Table

For a $D_n - \kappa$ relation such that D_n is always greater than zero, any initial wave will only cross a node once. This follows from fact that $\psi_{,t} = -D_n(\kappa)|\nabla\psi| \leq 0$ and is, hence, monotonically decreasing in time. So, instead of saving several $\psi(x, y, t)$ arrays in time and taking contours at $\psi(x, y, t) = 0$, it is more efficient to create a burn table. A burn table is just a record of wave arrival times as a function of space, $t_b(x, y)$. This is accomplished numerically by checking each node to see if its value of ψ changes sign after each timestep. If it has, then the $\psi(x, y, t) = 0$ contour has passed the node, and linear interpolation in time is used to record the burn time.

4. NUMERICAL STABILITY AND ACCURACY

In this section, we investigate the stability restrictions placed on the numerical algorithms described in Section 3. Accuracy of the algorithms is also examined by making comparisons with an exact solution. Since the numerical algorithms are explicit in time, certain restrictions on the timestep are required to ensure numerical stability. As one might expect, the hyperbolic operator will have a CFL type restriction, $c_1\ \Delta t/\Delta x \leq 1$, while the "mostly parabolic" operator will have a restriction like $c_2\ \Delta t/\Delta x^2 \leq 1$.

Since (7) is nonlinear, classical methods for determining the stability of difference equations cannot be used. We will first obtain the timestep restriction for the first-order hyperbolic part of the operator, consisting of (8) and (9), by requiring that the scheme be monotone. Then a timestep restriction for the second-order parabolic part of the operator, consisting of (8) with central differenced curvature terms, will be found by a frozen coefficient analysis. Then, we will give the timestep required for a general $D_n(\kappa)$.

We briefly describe the timestep restriction for the $D_n = D_{CJ}$ case, with first-order accurate differences (9) and no curvature dependence. The resulting PDE is hyperbolic and has the property of being monotone. Monotonicity implies the following (see [15] for details): If two sets of initial data are given (say in 2D), $\psi_1(x, y, t = 0)$ and $\psi_2(x, y, t = 0)$, such that $\psi_2(x, y, t = 0) \geq \psi_1(x, y, t = 0)$ for all x and y, then for all time and space, $\psi_2(x, y, t) \geq \psi_1(x, y, t)$. A scheme which has this property is called a monotone method. Denote the solution of our difference equation as $\psi_{i,j}^{n+1} = \mathcal{H}(\psi_{i,j}^n)$, where the function $\mathcal{H}$ is given from (8) and (9). Obviously, $\mathcal{H}$ will depend on Δx and Δt. To ensure that a numerical scheme is monotone, we require that

$$\frac{\partial}{\partial\psi_{i,j}^n}\mathcal{H}(\psi_{k,l}^n) \geq 0$$

for all i, j, k, l.

Carrying out all the possible forms of $\mathcal{H}$ (which depend on the upwinding) gives the CFL restriction on a uniform grid,

$$\frac{2D_{CJ}\,\Delta t}{\Delta x} \leq 1 \qquad (19)$$

in 2D and

$$\frac{\sqrt{6}D_{CJ}\,\Delta t}{\Delta x} \leq 1 \qquad (20)$$

in 3D. In general, monotone schemes are limited to first-order accuracy. But the first-order stability results can be used for the second-order scheme, since the second-order scheme reduces to first-order in non-smooth regions (although the second-order scheme will not strictly be monotone).

Now, we consider the problem of determining the timestep restriction due only to the curvature dependent terms, i.e., (8), and central differenced curvature terms. Also assume a linear dependence on the curvature. Then the level set equation becomes

$$\psi_t = \alpha \frac{\psi_{xx}\psi_y^2 - 2\psi_{xy}\psi_x\psi_y + \psi_{yy}\psi_x^2}{\psi_x^2 + \psi_y^2},$$

where α is a positive constant. Notice the above can be rewritten as

$$\psi_t = \alpha(a^2\psi_{xx} - 2ab\psi_{xy} + b^2\psi_{yy}), \tag{21}$$

where $a^2 + b^2 = 1$. For the purposes of this discussion, we assume that a and b are constants and carry out the standard von Neumann stability analysis on the resulting linear operator. The timestep restriction for the "linearized" curvature dependent term is

$$\frac{2\,\Delta t|\alpha|}{(\Delta x)^2} \le 1$$

in 2D and

$$\frac{4\,\Delta t|\alpha|}{(\Delta x)^2} \le 1$$

in 3D. Thus, for the linear $D_n(\kappa) = D_{CJ} - \alpha\kappa$, the timestep restriction is

$$\frac{2D_{CJ}\,\Delta t}{\Delta x} + \frac{2\,\Delta t|\alpha|}{(\Delta x)^2} \le 1 \tag{22}$$

in 2D and

$$\frac{\sqrt{6}D_{CJ}\,\Delta t}{\Delta x} + \frac{4\,\Delta t|\alpha|}{(\Delta x)^2} \le 1 \tag{23}$$

in 3D. The above timestep restriction can be used for a nonlinear $D_n(\kappa)$, by replacing the constant, $|\alpha|$, in (22) or (23), with $\max(|\partial\alpha/\partial\kappa|)$.

Next, we demonstrate that (8) and (10) with central differenced curvature dependent terms gives second-order convergence. Although the truncation error of this scheme is $O(\Delta t) + O((\Delta x)^2)$, we expect second-order convergence since for stability $\Delta t \propto (\Delta x)^2$ as $\Delta x \to 0$. We demonstrate the convergence properties of the above algorithm by comparison with an exact solution.

TABLE I

Numerical Accuracy for an Expanding Circle with $D_n = 1 - 0.1\ \kappa$

$\Delta x = \Delta y$	E_1	R_c
1/40	1.14×10^{-3}	
1/80	3.26×10^{-4}	1.81
1/160	8.70×10^{-5}	1.91
1/320	2.16×10^{-5}	2.01

The example problem will be an expanding quarter circle, whose center is at the origin and has with initial radius, $r = 0.2$. The numerical domain will be $0 \le x \le 1$ and $0 \le y \le 1$, with symmetry conditions at $x = 0$ and $y = 0$, and non-reflective conditions at $x = 1$ and $y = 1$. We take $D_n = 1 - 0.1\ \kappa$, to test the second-order ENO/upwinding scheme. For error analysis purposes we use the error measured on the discrete L_1 norm,

$$E_1 = \sum_{i,j} \left|t_b^{\text{exact}} - t_b^{\text{numerical}}\right| \Delta x\, \Delta y.$$

The exact solution is obtained by noticing that the problem is really one-dimensional, with the velocity of the front being only a function of the radius ($\kappa = 1/r$), and integrating the resulting ODE for the radius as a function of time. This gives $t_b^{\text{exact}}(x, y) = \sqrt{x^2 + y^2} - 0.2 + 0.1 \log[(0.1 - \sqrt{x^2 + y^2})/(0.1 - 0.2)]$ (with $\sqrt{x^2 + y^2} \ge 0.2$). Table I shows the error, E_1, for several $\Delta x = \Delta y$'s. The timestep was taken to be 0.8 of the maximum allowed by (22). Also shown is the calculated numerical order of accuracy, R_c. Notice that second-order convergence is achieved.

5. COMPARISON OF DNS AND LEVEL SET SOLUTION OF DSD

Here, we make comparisons of DSD theory with the direct numerical simulation (DNS) of detonations. The direct numerical simulations were carried out with the Los Alamos code CAVEAT [19]. CAVEAT is based on a second-order Godunov-type shock capturing scheme, which can be used in either Eulerian or Lagrangian mode. Of particular interest is the location and subsequent dynamics of the detonation front. Next, we give the mathematical formulation of the detonation model used in the DNS.

5.1. Compressible Reactive Euler Equations

DSD and CAVEAT both have the reactive compressible Euler equations as the basic model for studying detonations. These are

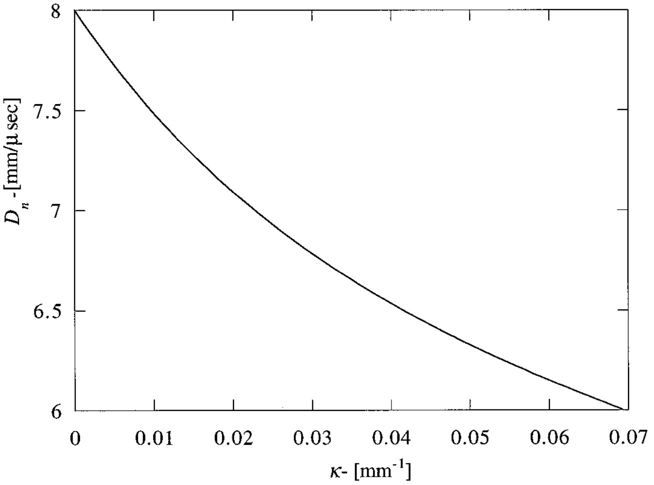

FIG. 10. $D_n(\kappa)$ law for ideal equation of state model.

$$\begin{aligned} \frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{u} &= 0,\\ \rho\frac{D\mathbf{u}}{Dt} + \nabla p &= 0,\\ \frac{De}{Dt} + p\frac{D(1/\rho)}{Dt} &= 0,\\ \frac{D\lambda}{Dt} &= r(p,\rho,\lambda), \end{aligned} \tag{24}$$

with the ideal equation of state

$$e = \frac{p}{\rho(\gamma-1)} - Q\lambda,$$

where Q is the heat of detonation, λ is the reaction progress variable ($\lambda = 0$ for unreacted material, and $\lambda = 1$ for completely reacted material), and r is the reaction rate. For our comparison, we take

$$r = 2.5147\ \mu\text{s}^{-1}(1-\lambda)^{1/2},$$

as the rate law and use $Q = 4\ \text{mm}^2/\mu\text{s}^2$, $\gamma = 3$ and upstream conditions $p_o = 10^{-4}$ GPa, $\rho_o = 2$ g/cc and $\mathbf{u} = 0$. These parameters were chosen to mock up a condensed phase explosive with the ideal equation of state. These parameters give $D_{CJ} = 8$ mm/μs, and a steady-state 1D reaction-zone length of 4 mm. For this model, DSD theory gives a $D_n(\kappa)$ relation shown in Fig. 10; see [1].

5.2. Numerical Examples and Comparisons

As stated previously, we can obtain the dynamics of the detonation front by solving the compressible, reactive Euler equations with a DNS. Unfortunately, information like the detonation shock speed, curvature of the shock front, etc. are not directly available from a DNS. But, since the fluid begins to react just after it passes through the inert shock (detonation front), and the reaction progress is monotone, we may approximate the front location as the level-curve (contour) $\lambda = 0.01$, say. And for problems with quiescent upstream conditions, we know that the detonation shock front will pass a fixed Eulerian point at most only once. So then it is possible to create a DNS burn table, by sweeping over the computational grid and searching for grid points where the quantity ($\lambda - 0.01$) changes sign from one time level to the next. Again, linear interpolation in time can be used to get an accurate estimate of the burn time, $t_b^{\text{DNS}}(x, y)$. Once we have this DNS burn table, important quantities such as shock speed, curvature, etc. may be found. For example, the shock speed is given by $D_n = 1/|\nabla t_b|$. The front locations are given simply as contours of $t_b^{\text{DNS}}(x, y)$, and curvature of the shock front is given by (6) with t_b replacing ψ. All the problems we consider represent difficult tests for DSD, since the deviation of D_n from D_{CJ} is large. Next, we give results from the DNS/DSD comparison.

The first example is a ratestick problem. A ZND detonation wave, initially at $x = 5$ mm (and traveling to the right), is used to initiate the unburnt explosive material located at $x \geq 5$ mm, $0 \leq y \leq 40$ mm. An inert confining material (with $Q = 0$, $\rho_o = 1.5$ g/cc, $\gamma = 1.4$) is located in the region 40 mm $\leq y \leq$ 45 mm. Perfect confinement (reflection) is applied at $y = 0$ mm. CAVEAT is run in Lagrangian mode to handle this multi-material problem. Also, $\Delta x = \Delta y = 0.2$ mm, which puts roughly 20 cells in the reaction zone. Wave fronts at various times, along with a spatial history of the detonation velocity are shown in Fig. 11a. Notice the slowing of the detonation front at the inert interface, $y = 40$ mm. This sends a disturbance along the lead shock (and through the subsonic portion of the reaction zone) which propagates into the interior of the ratestick. This, in turn, affects the shape of the detonation wave and the axial propagation speed which is approximately 7.73 mm/μs at 20 μs.

Figure 11b shows the level set solution to the DSD problem with $D_n(\kappa)$ given in Fig. 10, and $\omega_s = \omega_c = 54.7°$, and $\Delta x = \Delta y = 1$ mm. The shock front is slowed since the angle boundary condition is applied at $y = 40$ mm. Notice that the DSD solution calculates the front shapes well, compared to the DNS. It also predicts the angle at the interface, and the axial velocity, which is 7.63 mm/μs at 20 μs. The discrepancies in the solution are due to the fact that the $D_n - \kappa$ relation is parabolic and reaches steady state faster than the underlying hyperbolic Euler equations. This causes the DSD solution to lag the DNS by roughly one reaction-zone length. If faster transients (i.e., shock acceleration) are kept in the DSD theory, one recov-

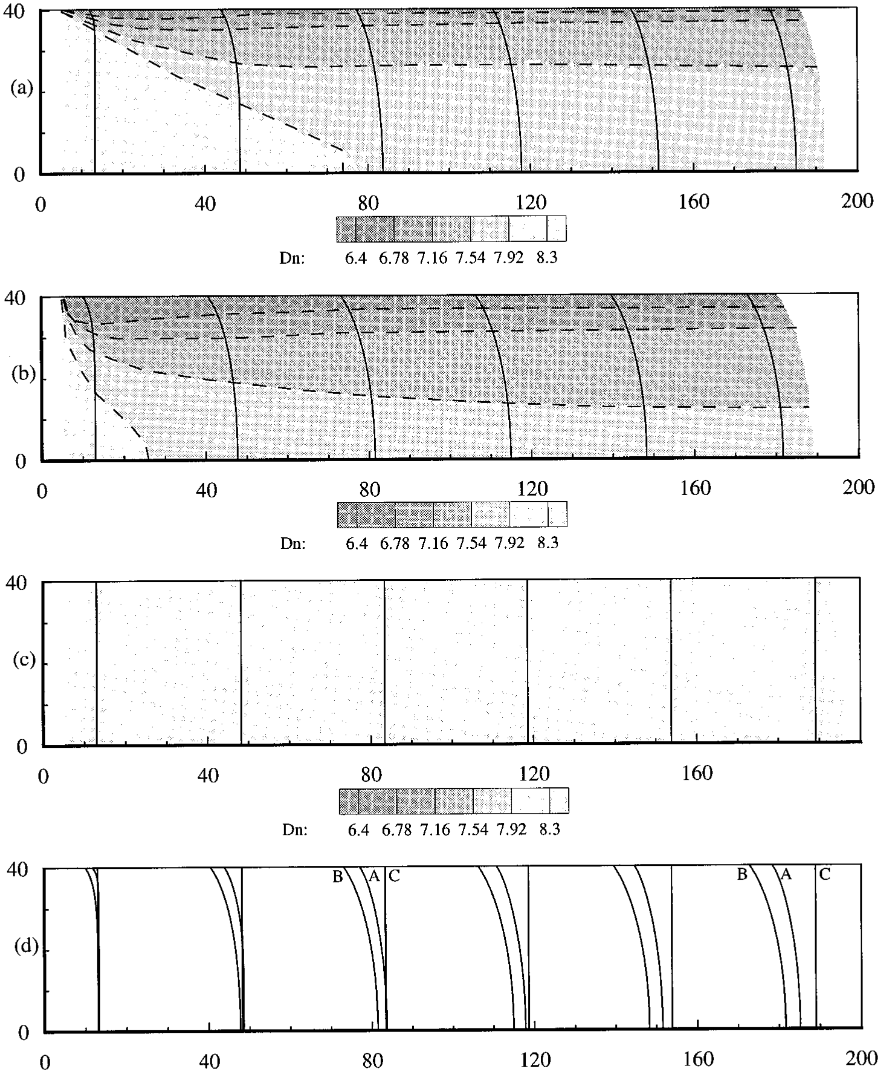

FIG. 11. Ratestick example: (a) direct numerical simulation; (b) detonation shock dynamics; (c) Huygens construction; (d) comparison of shock front locations at 1, 5.4, 9.8, 14, 18.6, 23 μs.

ers hyperbolicity, and more realistic finite wave speed disturbances can be seen [22].

For comparison sake, the Huygens ($D_n = 8$ mm/μs) solution is given in Fig. 11c. A comparison of wave fronts is given in Fig. 11d. Obviously, the Huygens solution does not predict any velocity deficit, nor does it calculate the correct wave shapes.

The second example is a converging channel problem with perfect confinement along the walls. A ZND detonation wave, again located initially at $x = 5$ mm, encounters a 20° ramp. Once the detonation reaches the ramp, a compressive wave is reflected downwards, and the detonation velocity increases as a result. See Fig. 12a for wave front and detonation velocity plots. Note that the detonation front is broadly curved, and no "Mach"-like reflection appears.

Figure 12b shows the DSD solution with a linear extrapolation for the converging branch ($D_n = 8$ mm/μs $-$ (66.8 mm^2/μs)κ). Here, $\omega_c = 90°$ and $\omega_s = 54.7°$. Notice how well the DSD solution reproduces the shock fronts. Also shown is the Huygens solution in Fig. 12c and the comparison of wave fronts in Fig. 12d.

The final comparison is a diverging channel problem with perfect wall confinement. This is the same as the previous problem, but the channel diverges at 45°. When the detonation shock diffracts around the corner, a rarefaction wave is propagated out from the wall, and the detonation velocity decreases as a result. See Fig. 13a. Notice, at about 19 μs, the curvature of the detonation front is decreasing and the front begins to accelerate.

Figure 13b shows the DSD solution. Notice, again, how well the DSD results compare with the DNS. Figure 13c

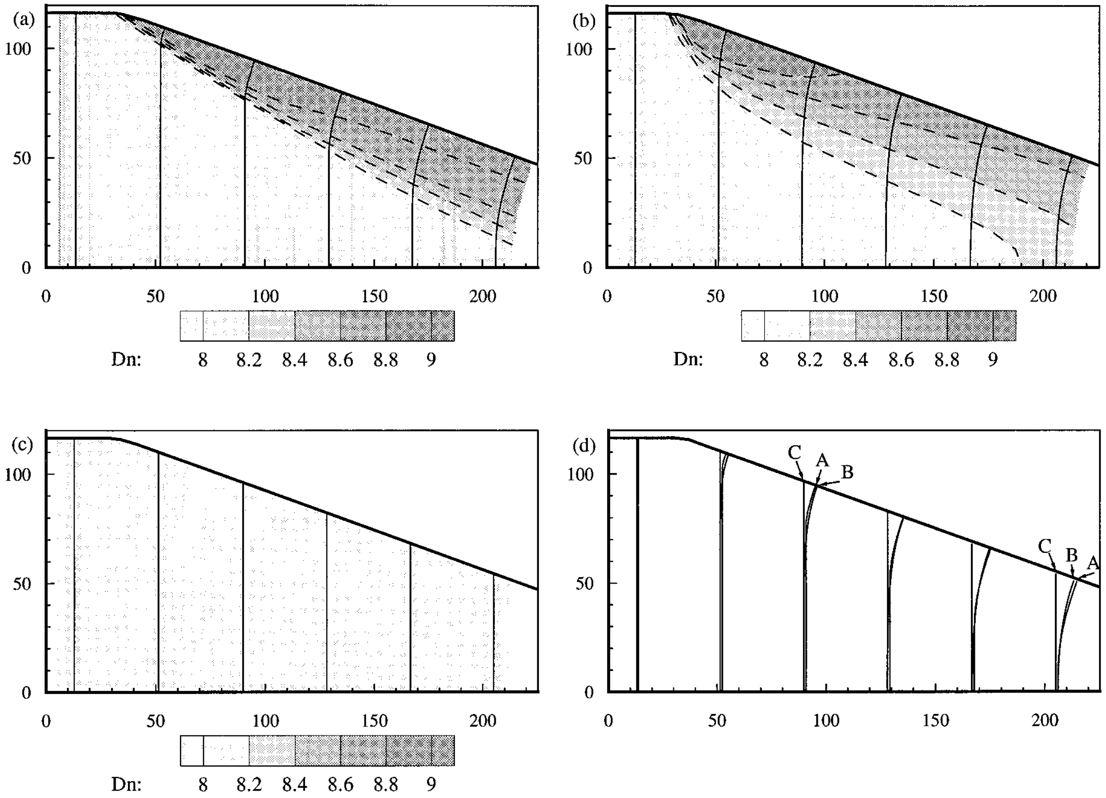

FIG. 12. Converging channel example: (a) direct numerical simulation; (b) detonation shock dynamics; (c) Huygens construction; (d) comparison of shock front locations at 1, 5.8, 10.6, 15.4, 20.2, 25 μs.

shows the Huygens result. A comparison of shock fronts is given in figure 13d.

5.3. 3D Seven-Point Detonation in PBX9502

We demonstrate the ability of a level set formulation to easily handle 3D multiple front interaction with the following example. We use the $D_n(\kappa)$ relation from Fig. 3, in a cube with length 64 mm. Initially, there are seven spherically expanding detonations, six in a hexagonal pattern, and one in the center. See Fig. 14. The spherical detonations merge, then intersect the edges of the cube, and eventually burn out of the domain.

6. EMBEDDING AND RELATIVE MOTION OF THE LEVEL-SURFACE CURVES

Here we discuss the embedding of a level curve of interest (the shock) and the dynamics of the relative motion of the level curves in the same family. We do this to develop insight into (1) why the embedding idea of the LS method algorithm works so well for a monotonic $D_n(\kappa)$ relation and (2) to point out restrictions that arise when trying to extend the LS method to include boundary conditions at edges. Boundaries are nearly always present in explosive applications. In large measure, the boundary conditions force the evolution of the level curve in our DSD problems.

To uniquely identify the physical shock, requires that neighboring level curves not cross each other in a finite time. A crossing of the level curves leads to non-uniqueness in D_n (x, y, t). Then the problem of the propagation of the level curves in x, y-space, described by (5), is not uniquely posed. In this section we discuss three topics related to embedding.

In Section 6.2 we describe some of the properties of the level-set equation and show how DSD front theory derived in [8] is contained in the LS method formulation. This discussion is focused on exposing the mathematical properties of the multivariable, second-order spatial operator that appears in the LS PDE. The operator type (i.e., whether it is elliptic or parabolic) is sensitive to the spatial direction. In turn, this defines the operator type of the fully time-dependent LS PDE (i.e., whether it is parabolic or hyperbolic). This discussion leads naturally to our demonstrating how DSD front theory, a parabolic theory, is contained in the LS formulation of the problem.

In Section 6.3 we derive an auxiliary PDE for the level-curve spacing. This PDE is used to study the motion of level curves relative to one another. We use $|\nabla\psi|^{-1}$ to

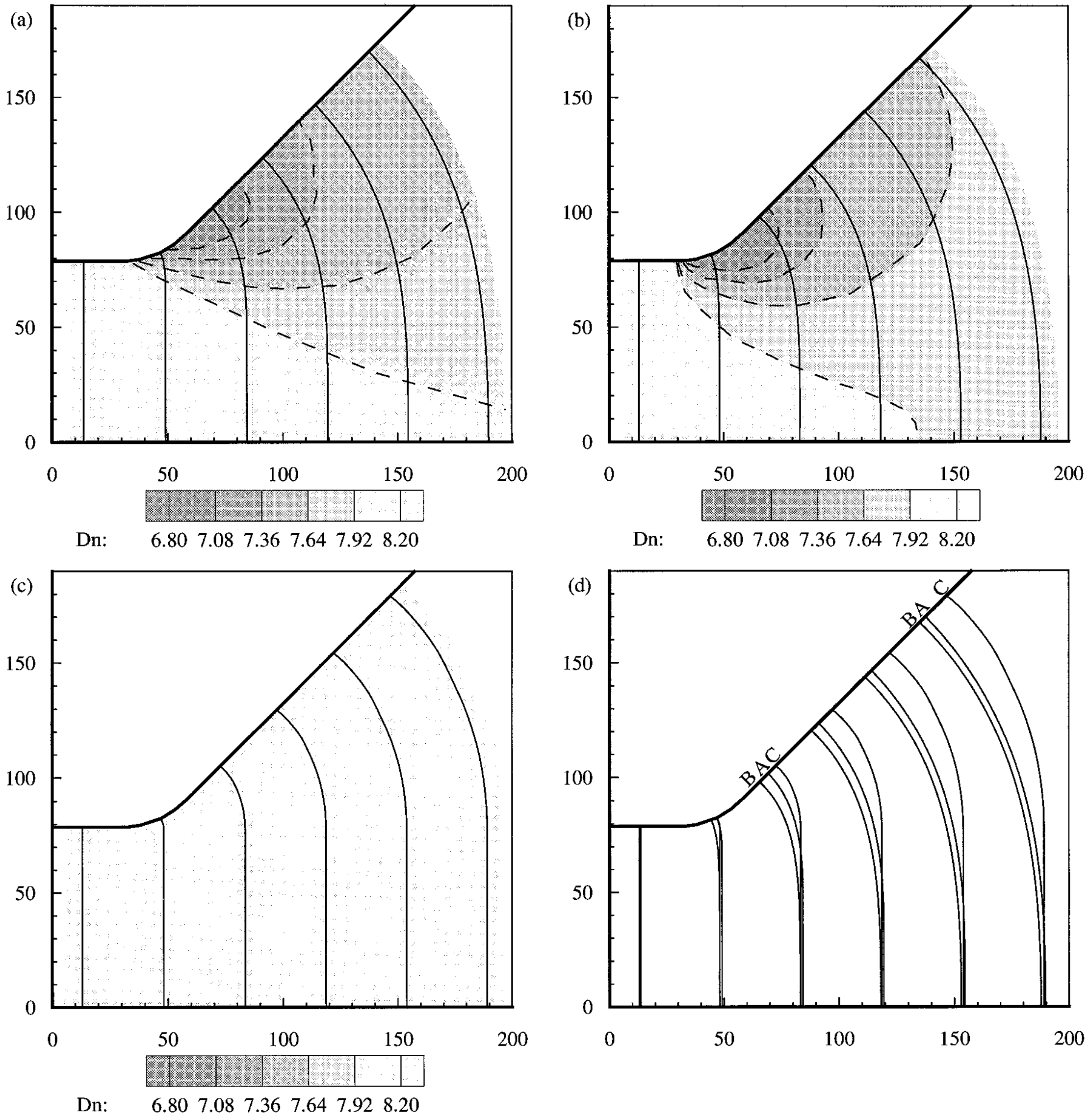

FIG. 13. Diverging channel example: (a) direct numerical simulation; (b) detonation shock dynamics; (c) Huygens construction; (d) comparison of shock front locations at 1, 5.4, 9.8, 14, 18.6, 23 μs.

measure the local relative spacing of the level curves. A qualitatively interpretation for this choice is that regions of large gradients in the smooth level-set function correspond to a high-density of curves (i.e., closely spaced level curves). This at least motivates the name "distance function" for the quantity $d(x, y, t) = |\nabla\psi|^{-1}$ and why we derive an auxiliary PDE for d.

Our use of a distance function $d(x, y, t)$, shares both similarities and differences with previous work. Sussman, Smereka, and Osher [9] introduced $|\nabla(\psi)|$ as a "distance function" with which to measure the spacing of level curves. Sussman *et al.* use $(1 - |\nabla\psi|)$ to drive a continual renormalization of $\psi(x, y, t)$ such that $|\nabla\psi| = 1$. This renormalization is used to inhibit the formation of large gradients in $\psi(x, y, t)$, which otherwise occur and destroy the solution quality. Unlike their work, we use $d(x, y, t)$ as a passive observer of the evolution of $\psi(x, y, t)$. In a series of papers by Evans and Spruck [10, 11] the evolution of the level-curve spacing, for problems, where $D_n(\kappa) = -\kappa$, is studied with a "distance function" of a different type. Their distance function $\hat{d}$ is also a passive variable. It measures the signed, minimum normal distance from a given level curve to some nearby point fixed in space. Points ahead of the curve are signed positively and those behind with a negative sign. In our earlier discussions, we used this same distance function to set up initial data for $\psi(x,$

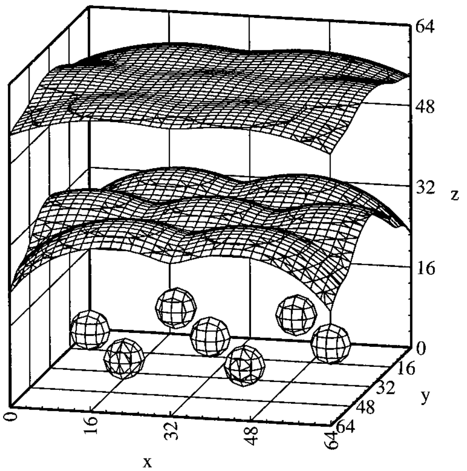

FIG. 14. Seven-point detonation, shock front locations at 0, 3, 6 μs.

y, $t = 0$). By differencing two such oppositely signed distances, the separation of two level curves is followed with a variable $w = \hat{d}_1 - \hat{d}_2$. The evolution equation they get for w is similar in form to the one we derive. In spirit at least, our discussion follows Evans and Spruck [10]. The advantage of $|\nabla\psi|^{-1}$ as a distance function in our application, rests with the ease with which DSD boundary conditions can be expressed with this d.

Our application is the first to use level-set methods for problems with real boundary conditions (i.e., not simply using continuation conditions). In Section 6.4, we use this auxiliary PDE for $d(x, y, t)$ to demonstrate how DSD-type boundary conditions influence the spacing of the level curves near boundaries. To illustrate the issues, we consider the most difficult boundary situation vìs-a-vìs the convergence of level curves; the expansion (i.e., diffraction) of detonation around a corner (see Fig. (16)). Like the LS equation, the d-equation is of mixed parabolic/hyperbolic type. To simplify the analysis of this system, we introduce a small amount of additional "diffusion" to the d-equation, to obtain a strictly parabolic equation for a "viscosity" subsolution (i.e., an equation for a lower bound on d). Using the existing literature on parabolic PDEs, we show via a maximum principal that $d(x, y, t)$ remains bounded away from zero.

We begin our discussion by dealing with some simple mathematical preliminaries.

6.1. Preliminary Calculations

In this section we introduce the normal and tangent vectors to a level curve at a fixed Eulerian point and the corresponding directional derivatives. We then derive two auxiliary PDEs by taking the gradient of the LS equation to obtain equations for the magnitude and phase of the gradient vector, $\nabla\psi$. Refer to Fig. 15 for a description of the coordinate geometry. Given a level curve $\psi(x, y, t) =$ const the instantaneous normal and tangent vectors at a fixed Eulerian point are obtained by taking the total derivative

$$\psi_x\, dx + \psi_y\, dy = 0, \tag{25}$$

from which we get the slope of a level curve

$$\frac{dy}{dx} = -\frac{\psi_x}{\psi_y} = -\tan(\theta). \tag{26}$$

The angle $\theta(x, y, t)$ is defined as the angle between the shock normal and the y-axis, where $\theta(x, y, t)$ is increasing in the clockwise direction. In terms of this angle, the normal and tangent to a level curve are

$$\begin{aligned} \hat{n} &\equiv \frac{\nabla\psi}{|\nabla\psi|} = \sin(\theta)\hat{\imath} + \cos(\theta)\hat{\jmath}, \\ \hat{t} &= \frac{(\psi_y\hat{\imath} - \psi_x\hat{\jmath})}{|\nabla\psi|} = \cos(\theta)\hat{\imath} - \sin(\theta)\hat{\jmath}, \end{aligned} \tag{27}$$

from which the directional derivatives normal and tangent to a level curve are

$$\begin{aligned} \hat{n}\cdot\nabla(\ ) &= \sin(\theta)(\ )_x + \cos(\theta)(\ )_y, \\ \hat{t}\cdot\nabla(\ ) &= \cos(\theta)(\ )_x - \sin(\theta)(\ )_y. \end{aligned} \tag{28}$$

The equations for the auxiliary variables that we seek are obtained by taking the gradient of Eq. (26)

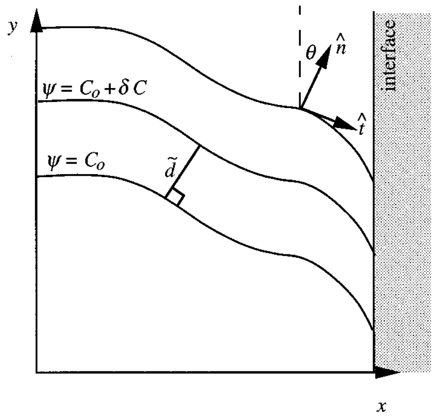

FIG. 15. Geometry of the space of level curves, showing the definition of the local normal and tangent vectors and shock normal angle $\theta(x, y, t)$. The normal distance between two "neighboring" level curves C_o and $C_o + \delta C$ is $\tilde{d}(x, y, t)$.

$$\nabla\psi_x = \nabla\psi_y \tan(\theta) + \psi_y \sec^2(\theta)\nabla\theta \tag{29}$$

and subsequently constructing

$$\hat{t}\cdot\nabla(|\nabla\psi|) = |\nabla\psi|\hat{n}\cdot\nabla(\theta) \tag{30}$$

and the gradient of the level-set equation

$$\hat{n}(|\nabla\psi|)_t + \hat{t}|\nabla\psi|(\theta)_t + |\nabla\psi|\nabla(D_n) + D_n\nabla(|\nabla\psi|) = 0, \tag{31}$$

where we have used

$$\nabla\psi = \hat{n}|\nabla\psi|, \quad (\hat{n})_t = \hat{t}(\theta)_t. \tag{32}$$

Equation (31) is a vector equation. The tangential component yields an equation for the evolution of the shock-normal angle, θ,

$$(\theta)_t + D_n\hat{n}\cdot\nabla(\theta) = -\hat{t}\cdot\nabla(D_n), \tag{33}$$

while the normal component yields an equation for the evolution of $|\nabla\psi|$,

$$(|\nabla\psi|)_t + D_n\hat{n}\cdot\nabla(|\nabla\psi|) = -|\nabla\psi|\hat{n}\cdot\nabla(D_n). \tag{34}$$

The shock normal angle, $\theta(x, y, t)$, and $|\nabla\psi|$ represent the phase and magnitude of $\nabla\psi$. They describe how the local orientation and "slope" of the level surface is changing. In the next subsections, we use these equations to show how the level-set formulation relates to our previous description of DSD theory and how the spacing between level curves evolves. We restrict our developments to a linear $D_n(\kappa)$-law,

$$D_n(\kappa) = D_{CJ} - \alpha\kappa, \tag{35}$$

where α is a positive constant.

6.2. Properties of the Level Set Equation

Inserting Eq. (35) for $D_n(\kappa)$ in Eq. (5), obtains

$$\psi_t + D_{CJ}\hat{n}\cdot\nabla(\psi) - \alpha\mathscr{L}(\psi) = 0, \tag{36}$$

where $\mathscr{L}(\psi)$ is the curvature and the operator $\mathscr{L}$ is given by

$$\begin{aligned}\mathscr{L}(\ ) &\equiv a_{11}\frac{\partial^2}{\partial x^2} + 2a_{12}\frac{\partial^2}{\partial x\partial y} + a_{22}\frac{\partial^2}{\partial y^2},\\ &= \nabla^2(\ ) - \hat{n}\cdot\nabla(\hat{n}\cdot\nabla(\ )) + \hat{n}\cdot\nabla(\theta)(\hat{t}\cdot\nabla(\ )),\\ &= \hat{t}\cdot\nabla(\hat{t}\cdot\nabla(\ )) + \hat{t}\cdot\nabla(\theta)(\hat{n}\cdot\nabla(\ )),\end{aligned} \tag{37}$$

and the coefficient a_{ij} are the elements of the real, symmetric array

$$\begin{aligned}\mathscr{A} &= I - \frac{1}{|\nabla\psi|^2}\begin{pmatrix}\psi_x^2 & \psi_x\psi_y\\ \psi_x\psi_y & \psi_y^2\end{pmatrix}\\ &= \begin{pmatrix}1-\sin^2(\theta) & -\sin(\theta)\cos(\theta)\\ -\sin(\theta)\cos(\theta) & 1-\cos^2(\theta)\end{pmatrix}.\end{aligned} \tag{38}$$

Now for all real vectors $\boldsymbol{\xi} = (\xi_1, \xi_2)$ it follows that since $\mathscr{A}$ is real and symmetric that $\boldsymbol{\xi}^{\mathrm{T}}\cdot\mathscr{A}\boldsymbol{\xi} \geq 0$, that is,

$$\sum_{i,j=1}^{2} a_{ij}\xi_i\xi_j = \frac{(\xi_1\psi_y - \xi_2\psi_x)^2}{|\nabla\psi|^2} \geq 0 \tag{39}$$

is positive semidefinite for $\boldsymbol{\xi} \neq 0$. From the theory of partial differential equations [7], it follows that $\mathscr{L}(\ )$ is not strictly elliptic and therefore Eq. (36) is *not* uniformly parabolic. From Eq. (39), we see that $\mathscr{L}(\ )$ is parabolic (i.e., Eq. (39) is zero) in the direction normal to the level curve, $y - \tilde{y} = (x - \tilde{x})\cot(\theta)$, where $(\tilde{x}, \tilde{y})$ denotes an Eulerian point on the level curve. Parabolicity of Eq. (36) requires that Eq. (39) be positive definite. Thus Eq. (36) is of a hybrid type and is only parabolic along level curves. Further, when α is identically zero, Eq. (36) becomes *strictly* hyperbolic, the eikonal equation of geometrical optics. Thus we see that the curvature-related terms, those proportional to α, describe effects that propagate only *along* level curves.

Turning to Eq. (33) for the evolution of $\theta(x, y, t)$, we can now understand how our earlier DSD work [8], which obtained a front theory for the detonation shock that is parabolic, is embedded in the LS formulation. To see this, we examine the connections that exist between the DSD front equation, boundary conditions, and Eq. (36).

The DSD boundary condition is a condition on the angle θ. From the orthogonality of the normal $\hat{n}$ and tangent $\hat{t}$ vectors, Eqs. (27)–(28), it follows that at every point in space $t\cdot\nabla(\psi) = 0$ and, consequently,

$$\psi_x\cos(\theta) - \psi_y\sin(\theta) = 0. \tag{40}$$

If θ is prescribed at the edge as in DSD theory, Eq. (40) is then the LS boundary condition for the DSD problem.

The equation for the evolution of the shock-normal angle θ is

$$(\theta)_t + D_n\hat{n}\cdot\nabla(\theta) = -\hat{t}\cdot\nabla(D_n). \tag{41}$$

Since $D_n(\kappa)$ is a linear function and κ is given by

$$\kappa = \hat{t}\cdot\nabla(\theta), \tag{42}$$

from Eqs. (28), (42) it is clear that Eq. (41) contains only θ as a dependent variable which with the boundary condition on θ constitutes a totally self-contained, nonlinear problem for θ,

$$(\theta)_t + D_n\hat{n}\cdot\boldsymbol{\nabla}(\theta) = \alpha\hat{t}\cdot\boldsymbol{\nabla}(\hat{t}\cdot\boldsymbol{\nabla}\mathbf{d}(\theta)). \tag{43}$$

Using Eq. (37), we can also write Eq. (43) as

$$(\theta)_t + D_{CJ}\hat{n}\cdot\boldsymbol{\nabla}(\theta) = \alpha\mathscr{L}(\theta). \tag{44}$$

Equation (43) is equivalent to the parabolic PDE, derived in [8] that describes the evolving shape of the detonation front. The operator on the left corresponds to the rate of change of θ as one travels with the level curve as it expands normal to itself. These changes are driven by the variations in θ that "diffuse" *along* the level curve as controlled by $\alpha\hat{t}\cdot\boldsymbol{\nabla}(\hat{t}\cdot\boldsymbol{\nabla}(\theta))$. Thus our DSD model, for which the variations in the wavefront θ depend, only information from the front is contained in Eq. (5). Next we show that $|\boldsymbol{\nabla}\psi|^{-1}$ is related to the local spacing between the level curves.

6.3. The Distance Function

To track how the distance between nearby level curves evolves, we introduce a measure of this distance called the "distance" function, $d(x, y, t)$. Referring to Fig. 15, two nearby level curves are pictured whose LS values differ by the small amount δC. Pick a point on the level curve $\psi(x, y, t) = C_o$ at time t, labeled with $x = \tilde{x}$ and $y = \tilde{y}$. Then express ψ on the level curve $\psi = C_o + \delta C$ as an expansion about $\tilde{x}$, $\tilde{y}$,

$$\psi(\tilde{x}, \tilde{y}, t) + \boldsymbol{\nabla}(\psi)\cdot d\mathbf{r} = C_o + \delta C. \tag{45}$$

Therefore, it follows that the normal distance, $\tilde{d}(\tilde{x}, \tilde{y}, t)$ from $\tilde{x}$, $\tilde{y}$ on curve C_o to curve $C_o + \delta C$, is given by

$$\hat{n}\cdot d\mathbf{r} = \frac{\delta C}{|\boldsymbol{\nabla}\psi(\tilde{x}, \tilde{y}, t)|}. \tag{46}$$

If we assume that initially the curves are labeled by their separation, then $\delta C = (\hat{n}\cdot d\mathbf{r})_o$, and $|\boldsymbol{\nabla}\psi|^{-1}$ is the scaled distance between nearby curves,

$$d \equiv |\boldsymbol{\nabla}\psi|^{-1}. \tag{47}$$

We refer to the $d(x, y, t)$ defined by Eq. (47) as the "distance" function. Returning to Eq. (34), we find that it describes the evolution of this distance function. Recognizing that $D_n = D_{CJ} - \alpha\kappa$ and $\kappa = \hat{t}\cdot\boldsymbol{\nabla}(\theta)$ allows us to rewrite the right-hand side of Eq. (34) as

$$\hat{n}\cdot\boldsymbol{\nabla}(D_n) = -\alpha\hat{n}\cdot\boldsymbol{\nabla}(\hat{t}\cdot\boldsymbol{\nabla}(\theta)), \tag{48}$$

which on using the commutation rule

$$\hat{n}\cdot\boldsymbol{\nabla}(\hat{t}\cdot\boldsymbol{\nabla}(\theta)) = \hat{t}\cdot\boldsymbol{\nabla}(\hat{n}\cdot\boldsymbol{\nabla}(\theta)) - (\hat{n}\cdot\boldsymbol{\nabla}(\theta))^2 - (\hat{t}\cdot\boldsymbol{\nabla}(\theta))^2, \tag{49}$$

Eqs. (30), (47), and the definition of level-curve curvature, can be written as

$$\begin{aligned}\hat{n}\cdot\boldsymbol{\nabla}(D_n) = \alpha\Bigg(&\hat{t}\cdot\boldsymbol{\nabla}\left(\frac{1}{d}\hat{t}\cdot\boldsymbol{\nabla}(d)\right)\\ &+ \left(\frac{1}{d}\hat{t}\cdot\boldsymbol{\nabla}(d)\right)^2 + \kappa^2\Bigg)\end{aligned} \tag{50}$$

and then simplified to

$$\hat{n}\cdot\boldsymbol{\nabla}(D_n) = \alpha\left(\frac{1}{d}\hat{t}\cdot\boldsymbol{\nabla}(\hat{t}\cdot\boldsymbol{\nabla}(d)) + \kappa^2\right). \tag{51}$$

Substituting Eqs. (51), (47) into Eq. (34) then yields

$$(d)_t + D_n\hat{n}\cdot\boldsymbol{\nabla}(d) = \alpha(\kappa^2\, d + \hat{t}\cdot\boldsymbol{\nabla}(\hat{t}\cdot\boldsymbol{\nabla}(d))). \tag{52}$$

Equation (52) is a *linear* PDE in $d(x, y, t)$ that can be used to follow the evolution of the distance between level curves. Although the coefficients of this equation depend on the solution of the LS equation, properties like the sign of the term $\kappa^2\, d$ and the sign and magnitude of the coefficients appearing in $\hat{t}\cdot\boldsymbol{\nabla}(\hat{t}\cdot\boldsymbol{\nabla}(\ ))$ (all of which are bounded by -1 and 1) are known. Because these coefficients have "nice" properties, we are able to get general results about the solution without actually solving the parent LS equation.

The boundary condition for the d-equation follows directly from Eq. (30) by replacing $|\boldsymbol{\nabla}\psi|$ by $1/d$,

$$\hat{t}\cdot\boldsymbol{\nabla}(d) + d\hat{n}\cdot\boldsymbol{\nabla}(\theta) = 0. \tag{53}$$

Recall that for our problem, $\hat{n}\cdot\boldsymbol{\nabla}(\theta)$ is prescribed at the edge by the DSD boundary condition.

Equation (53) is of the form of a linear, Dirichlet–Neumann boundary condition; a standard form of the boundary condition for an equation like Eq. (52). This allows us to get important, general results concerning how $d(x, y, t)$ responds to DSD boundary conditions.

We conclude this section by describing how the various terms appearing in Eq. (52) contribute to changing $d(x, y, t)$. For the Huygens problem, for which $\alpha = 0$ and $D_n = D_{CJ}$, $d(x, y, t)$ is simply advected through space at the constant speed D_{CJ}. Thus the spacing of the level curves stays fixed. Since D_n is known a priori, there are really no issues concerning embedding here.

When $\alpha > 0$, two terms drive changes in $d(x, y, t)$. The term $(\kappa^2 d)$ acts as a "source" that leads to increases in $d(x, y, t)$. The term $(\hat{t} \cdot \nabla(\hat{t} \cdot \nabla(d)))$ serves to "diffuse" any concentrations of $d(x, y, t)$ that develop, out in the direction of the level curve. This term acts so as to flatten any variations in $d(x, y, t)$. If for the moment we consider problems for which $\psi(x, y, 0)$ depends only on $r = \sqrt{x^2 + y^2}$ and not on θ, then the "diffusive" term disappears and $d(x, y, t)$ increases for problems in which either the level curves converge on $r = 0$ or expand outwards. When symmetry is lost, the "diffusive" term acts to diminish the action of $(\kappa^2 d)$ by spreading concentrations of $d(x, y, t)$; it does not act as a sink.

In the next section, we discuss the mathematical properties of Eq. (52) and how boundary conditions influence $d(x, y, t)$. For this purpose, we adopt a Cartesian coordinate representation and render the coefficients in Eq. (52) explicit,

$$(d)_t + D_{CJ}\hat{n} \cdot \nabla(d) = \alpha(\kappa^2 d + \mathscr{L}(d)), \tag{54}$$

where $\mathscr{L}(\;)$ is given by Eqs. (37)–(38).

6.4. DSD Boundary Conditions and Embedding

The principal DSD boundary condition is the angle boundary condition; the angle between the outward normals to the shock and the HE boundary is constant. It follows from the distance function boundary condition of Eq. (53),

$$\hat{t} \cdot \nabla(d) + d\hat{n} \cdot \nabla(\theta) = 0, \tag{55}$$

that in regions where $\hat{n} \cdot \nabla(\theta) \neq 0$, the distance function can have a nontrivial gradient at the boundary. In locations where θ experiences rapid changes along the boundary, like where the confinement changes or the explosive has a corner, either $\hat{t} \cdot \nabla(d)$ can become large or $d(x, y, t)$ could go to zero, or both. To illustrate the most difficult boundary issue that confronts us, we consider the diverging channel problem, with a finite radius of curvature transition section, shown in Fig. 16.

For this problem, the DSD boundary condition sets the angle between the edge and the level curves to 90°. If we construct $d(x, y, 0)$ so that it satisfies this edge boundary condition initially, then $(\theta(x, y, t))_{\text{edge}} = (\theta(x, y, 0))_{\text{edge}}$ and the time derivative in Eq. (43) is zero along the edge, so that

$$\left(\frac{\alpha\hat{t} \cdot \nabla(\kappa)}{D_n}\right)_{\text{edge}} = (\hat{n} \cdot \nabla(\theta))_{\text{edge}}. \tag{56}$$

For the problem shown in Fig. (16),

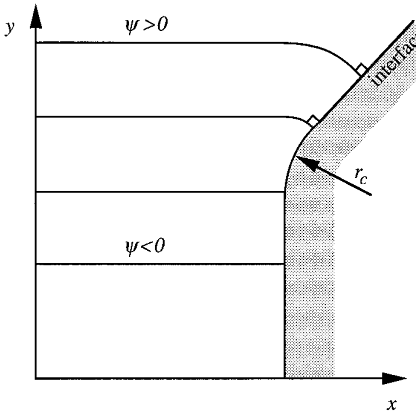

FIG. 16. The diverging channel problem with an infinite confinement wall. The angle boundary condition requires that the level curves be normal to the wall.

$$(\hat{n} \cdot \nabla(\theta))_{\text{edge}} = r_c^{-1} > 0, \tag{57}$$

in the region where θ_{edge} is changing and zero elsewhere. From Eq. (57) it follows that $(\alpha\hat{t} \cdot \nabla(\kappa)/D_n)_{\text{edge}} \geq 0$ for this problem. Therefore, if $(\alpha\hat{t} \cdot \nabla(\kappa))_{\text{edge}} \geq 0$ then $(D_{\text{n}})_{\text{edge}} \geq 0$, while if $(\alpha\hat{t} \cdot \nabla(\kappa))_{\text{edge}} < 0$ then $(D_n)_{\text{edge}} < 0$. In any event, $(D_n)_{\text{edge}}$ remains bounded so that κ remains bounded. Consequently, the "source" term $(\kappa^2 d)$ that appears in Eqs. (52) and (54) will remain bounded.

The analysis of the PDE for the distance function, Eq. (54), and the boundary condition, Eq. (53), for the problem of Fig. 16, is most easily carried forward by first modifying Eq. (54). The change we make is to take the operator $\mathscr{L}(\;)$, which is not strictly elliptic, and make it so. Following Evans and Spruck [10], we make $\mathscr{L}(\;)$ elliptic by adding a small term ε^2 to $|\nabla\psi|^2$ to get a slightly different operator

$$\mathscr{L}^\varepsilon(\;) = a_{11}^\varepsilon \frac{\partial^2}{\partial x^2} + 2a_{12}^\varepsilon \frac{\partial^2}{\partial x \partial y} + a_{22}^\varepsilon \frac{\partial^2}{\partial y^2}, \tag{58}$$

where

$$\mathscr{A}^\varepsilon = \mathscr{A} + \left(\frac{\varepsilon^2}{|\nabla\psi|^2 + \varepsilon^2}\right)(I - \mathscr{A}) \tag{59}$$

and where

$$\mathscr{L}^\varepsilon(\;) = \mathscr{L}(\;) + \left(\frac{\varepsilon^2}{|\nabla\psi|^2 + \varepsilon^2}\right)(\nabla^2(\;) - \mathscr{L}(\;)). \tag{60}$$

From Eq. (37) we see that $\nabla^2(\;) - \mathscr{L}(\;)$ is the second-order directional derivative in the normal direction. Recall that $\mathscr{L}(\;)$ corresponds to diffusion only along level curves. Think of the ε^2 as adding a small amount of "diffusion" to Eq. (54) in the direction normal to the level curve. This allows $d(x, y, t)$ to "diffuse" in all directions, which contributes to further decreases in d. Thus we see that the $O(\varepsilon^2)$ term in Eq. (60) serves to regularize the parabolic character of $\mathscr{L}(\ )$ in the direction normal to the level curves, thereby producing a strictly elliptic operator,

$$\sum_{i,j=1}^{2} a_{ij}^{\varepsilon}\xi_i\xi_j = \frac{\mathrm{u}(\xi_1\psi_y - \xi_2\psi_x)^2}{|\nabla\psi|^2} + \frac{\varepsilon^2}{|\nabla\psi|^2+\varepsilon^2}\left(\frac{(\xi_1\psi_x + \xi_2\psi_y)^2}{|\nabla\psi|^2}\right) \geq \mu(\xi_1^2 + \xi_2^2), \tag{61}$$

where μ is a positive constant.

Replacing $\mathscr{L}(\ )$ in Eq. (54) by $\mathscr{L}^{\varepsilon}(\ )$ and using $\mathscr{L}^{\varepsilon}(\ ) \geq \mathscr{L}(\ )$, it follows from the work of Crandall, Ishii, and Lions [12], that the resulting equation,

$$(d^{\varepsilon})_t + D_{CJ}\hat{n}\cdot\nabla(d^{\varepsilon}) - \alpha\mathscr{L}^{\varepsilon}(d^{\varepsilon}) - \alpha\kappa^2\, d^{\varepsilon} = 0, \tag{62}$$

defines the "viscosity" subsolution of Eq. (54), where

$$d^{\varepsilon}(x, y, t) \leq d(x, y, t). \tag{63}$$

Equation (62) with κ^2 bounded, together with the boundary condition

$$(\hat{t}/\nabla(d^{\varepsilon}))_{\text{edge}} + (d^{\varepsilon})_{\text{edge}}(\hat{n}\cdot\nabla(\theta))_{\text{edge}} = 0, \tag{64}$$

where $(\hat{n}\cdot\nabla(\theta))_{\text{edge}} = r_c^{-1}$ or 0, and the initial condition,

$$d^{\varepsilon}(x, y, 0) \geq 0, \tag{65}$$

constitutes a strictly parabolic, linear problem [7]. Since the problem is parabolic, it satisfies a maximum principal (i.e., the maxima and minima occur only on the boundaries). Using this fact, Pao [13] has proved a positivity lemma which states that in the entire problem domain,

$$d^{\varepsilon}(x, y, t) > 0, \tag{66}$$

unless $d^{\varepsilon}(x, y, t)$ is identically zero. The proof relies only on κ^2 in Eq. (62) being bounded and

$$(\hat{n}\cdot\nabla(\theta))_{\text{edge}} \geq 0 \tag{67}$$

in Eq. (64). In the limit $\varepsilon \to 0$, we have

$$\lim_{\varepsilon\to 0} d^{\varepsilon}(x, y, t) \to d(x, y, t) > 0. \tag{68}$$

Thus, even though we expect that $d(x, y, t)$ becomes small at the sharp corner that results when $r_c \to 0$, it follows that $d(x, y, t) \neq 0$. Although problems can be expected with numerical solution algorithms when $d(x, y, t)$ becomes small due to grid resolution effects, the mathematical description of d near the boundaries remains well posed.

ACKNOWLEDGMENTS

T. D. Aslam and D. S. Stewart have been supported by the United States Air Force (USAF), Wright Laboratory, Armament Directorate, Eglin Air Force Base, F08630–92-K0057. Additional support was made available through the USAF Office of Scientific Research, AASERT Grant F49620–93-1–0532. John Bdzil has been supported by the U.S. Department of Energy.

REFERENCES

1. D. S. Stewart and J. B. Bdzil, *Combust. Flame* **72,** 311 (1988).
2. S. Osher and J. A. Sethian, *J. Comput. Phys.* **79,** 12 (1988).
3. C.-W. Shu and S. Osher, *J. Comput. Phys.* **77,** 439 (1988).
4. J. B. Bdzil, W. C. Davis, and R. R. Critchfield, *Phys. Fluids*, submitted.
5. D. S. Stewart and J. B. Bdzil, "Asymptotics and Multi-Scale Simulation in a Numerical Combustion Laboratory," in *Asymptotic Induced Numerical Methods for PDE's, Critical Parameters and Domain Decomposition*, edited by H. G. Kaper and E. Garbey, NATO/ASI Series, C, Vol. 384, (Kluwer, Dordrecht, 1993), p. 163.
6. J. B. Bdzil and W. Fickett, Los Alamos Report, LA-12235-MS, 1992 (unpublished).
7. M. H. Protter and H. F. Weinberger *Maximum Principles in Differential Equations* (Springer-Verlag, New York, 1984), p. 187.
8. J. B. Bdzil and D. S. Stewart, *Phys. Fluids A* **1,** 1261 (1989).
9. M. Sussman, P. Smereka, and S. Osher, *J. Comput. Phys.* **114,** 146 (1994).
10. L. C. Evans and J. Spruck, *J. Differential Geom.* **33,** 635 (1991).
11. L. C. Evans and J. Spruck, *Trans. Amer. Math. Soc.* **330**(1), 321 (1992).
12. M. G. Crandall, H. Ishii, and P.-L. Lions, Bull. Am. Math Soc. **27,** 1 (1992).
13. C. V. Pao, *Nonlinear Parabolic and Elliptic Equations* (Plenum, New York, 1992), p. 54.
14. W. Fickett and W. C. Davis, *Detonation* (Univ. of California Press, Berkeley, 1979).
15. R. J. Leveque, *Numerical Methods for Conservation Laws* (Birkhauser, Basel, 1992).
16. R. J. Leveque, *J. Comput. Phys.* **78,** 36 (1988).
17. M. J. Berger and R. J. Leveque, AIAA Paper 89-1930-CP, 1989 (unpublished).
18. J. J. Quirk, *Comput. in Fluids* **23,** 125 (1994).
19. F. L. Addessio, J. Baumgardner, J. K. Dukowicz, N. L. Johnson, B. A. Kashiwa, and R. M. Rauenzahn, Los Alamos Report, LA-10613, 1990 (unpublished).
20. D. S. Stewart and J. B. Bdzil, "A lecture on Detonation Shock Dynamics," in *Mathematical Modeling in Combustion Science*, Lecture Notes in Physics, Vol. 249, (Springer-Verlag, New York/Berlin, 1988), p.17.
21. J. B. Bdzil, *J. Fluid Mech.* **108,** 195 (1981).
22. J. Yao and D. S. Stewart, *J. Fluid Mech.* **309,** 225 (1996).