

# Numerical Schemes for the Hamilton–Jacobi and Level Set Equations on Triangulated Domains

Timothy J. Barth\* and James A. Sethian†<sup>1</sup>

\**Information Sciences Directorate, NASA Ames Research Center, Moffett Field, California 94035; and*

†*Department of Mathematics, University of California at Berkeley, Berkeley, California 94720*

E-mail: \*barth@nas.nasa.gov and †sethian@math.berkeley.edu

Received October 2, 1997; revised March 18, 1998

---

Borrowing from techniques developed for conservation law equations, numerical schemes which discretize the Hamilton–Jacobi (H-J), level set, and Eikonal equations on triangulated domains are presented. The first scheme is a provably monotone discretization for the H-J equations. Unfortunately, the basic scheme lacks Lipschitz continuity of the numerical Hamiltonian. By employing a “virtual” edge flipping technique, local Lipschitz continuity of the numerical flux is restored on acute triangulations. Next, schemes are introduced and developed based on the weaker concept of positive coefficient approximations for homogeneous Hamiltonians. These schemes possess a discrete maximum principle on arbitrary triangulations and exhibit local Lipschitz continuity of the numerical Hamiltonian. Finally, a class of Petrov–Galerkin approximations is considered. These schemes are stabilized via a least-squares bilinear form. The Petrov–Galerkin schemes do not possess a discrete maximum principle but generalize to high order accuracy. Discretization of the level set equation also requires the numerical approximation of a mean curvature term. A simple mass-lumped Galerkin approximation is presented in Section 6 and analyzed using maximum principle analysis. The use of unstructured meshes permits several forms of mesh adaptation which have been incorporated into numerical examples. These numerical examples include discretizations of convex and nonconvex forms of the H-J equation, the Eikonal equation, and the level set equation. © 1998 Academic Press

---

## 1. INTRODUCTION

In this paper, three related equations are considered

- (1) The first-order Hamilton–Jacobi (H-J) equation

$$u_t + H(x, \nabla u) = 0, \quad (x, t) \in \Omega \times \mathbf{R}^+. \quad (1)$$

<sup>1</sup> Supported in part by the Office of Naval Research.

(2) The general level set equation

$$u_t + F(x, \nabla u, \kappa) |\nabla u| = 0, \quad (x, t) \in \Omega \times \mathbf{R}^+. \quad (2)$$

(3) The Eikonal equation

$$F(x) |\nabla u| = \pm 1, \quad F(x) > 0, x \in \Omega \quad (3)$$

for a domain  $\Omega \subset \mathbf{R}^d$ . In the level set equation,  $\kappa$  denotes the mean curvature of the  $(d - 1)$ -dimensional level set surface and is calculated from the divergence formula

$$\kappa = \nabla \cdot \frac{\nabla u}{|\nabla u|}. \quad (4)$$

The connection between H-J equations and conservation law equations in one space dimension is well known. Let  $u$  denote an entropy solution of the conservation law equation

$$\begin{aligned} u_t + f(u)_x &= 0, \quad (x, t) \in \mathbf{R} \times \mathbf{R}^+ \\ u(x, 0) &= u_0(x). \end{aligned} \quad (5)$$

Substituting  $u = v_x$  yields a related H-J equation

$$\begin{aligned} v_t + f(v_x) &= 0, \quad (x, t) \in \mathbf{R} \times \mathbf{R}^+ \\ v(x, 0) &= v_0(x). \end{aligned} \quad (6)$$

Both equations admit generalized solutions [17, 18, 10] and have related viscosity-limit solutions: if  $v$  is a viscosity solution of the H-J equation (6) then  $u$  is a entropy solution of the conservation law (5). This connection between conservation laws and H-J equations has been exploited in the design of numerical schemes for the H-J equations using finite differences and generalizations of Godunov's method on structured meshes [22, 11, 27, 28]. Of both theoretical and practical interest are monotone schemes for the H-J equations [22, 11, 19] since it can be shown that these schemes converge to the correct viscosity-limit solutions. Higher order accurate TVD and ENO extensions have also been demonstrated but it still remains an open problem to show convergence to viscosity-limit solutions in space dimension  $d \geq 2$ .

Our motivation for studying these equations stems, in part, from problems involving evolving interfaces. The general level set equation given above describes the motion of an interface advancing under a speed  $F$  in its normal direction. This perspective, which views the front as an implicitly defined embedded hypersurface, together with approximations based on finite difference formulations, was introduced by Osher and Sethian [22], based in part on the theory of evolving curves and surfaces developed earlier in [26], as well as [27].

The resulting "level set method" has been applied in a large number of areas, including problems that arise in geometry, fluid mechanics, computer vision, and manufacturing processes, see Sethian [30]. Numerous advances have been made to the original approach, including the adaptive narrow band methodology [2], the Fast Marching Method for solving the static Eikonal equation [29], and the variational level set method [33]. For details and summaries of level set techniques, see [28, 30].

Previous simulations using level set techniques have focussed on finite difference approximations on fixed, logically rectangular meshes. Such techniques have the advantage of high

degrees of accuracy, and programming ease. However, in some situations, a triangulated domain/finite element type approximation is desired. Three reasons include:

- *Adaptive mesh refinement.* Adaptive mesh refinement is straightforward in a triangulated setting, due to the ability to subdivide elements while avoiding nonconforming approximations, i.e., “hanging nodes.” Thus, in problems where one wants additional resolution, not just around the interface, but in response to other variables as well, this approach is valuable. While adaptive level set methods have been constructed on Cartesian meshes using finite difference approximations, such methods have additional complications especially when parabolic curvature terms are added to the formulation, see [21].

- *Interface-Fitted Coordinates.* In certain moving interface problems, jump conditions across the boundary are critical to both the solution of partial differential equations on either side of the interface and to evaluating the speed of the interface. Interpolation of these terms to neighboring grid elements can be delicate. In contrast, an interface-fitted coordinate system often offers a straightforward way to build these terms. In a triangulated setting, construction of a local interface-fitted set of vertices, similar to mesh adaptivity, can be a considerable advantage.

- *Propagating Interfaces on Manifolds.* In some applications, one wants to compute the motion of interfaces on non-planar manifolds, for example, on the surface of a body. In this case, a triangulated domain may be much more readily available on the surface than an orthogonal rectilinear coordinate system.

In the remaining sections, numerical schemes are developed for the Hamilton–Jacobi, Eikonal, and level set equations on triangulated domains in  $\mathbf{R}^d$ . Our strategy is to first develop a monotone numerical approximation for the H-J equations and then to relax the strict order preserving monotonicity condition in favor of a weaker positivity condition. This follows closely the development process taken for conservation law equations. In addition, an important class of Petrov–Galerkin approximations is considered which have no monotonicity or positivity principle but are capable of achieving high order accuracy. In later sections, the newly developed schemes are further extended to include the additional parabolic terms often present in level set equations. Finally, numerical calculations on uniform and adapted meshes are shown to demonstrate the accuracy and generality of the schemes. Specifically, the paper is organized in the following order of topics:

(1) *Numerical Fundamentals for H-J Equations.* The concepts of consistency and monotonicity of the prototype numerical Hamiltonian are defined and discussed.

(2) *Monotone Schemes.* A simple monotone updating formula is developed for simplices. Lipschitz continuity of the numerical Hamiltonian is improved using a virtual edge flipping strategy.

(3) *Positive Coefficient Schemes.* The weaker concept of positive coefficient schemes is introduced. A simple, compact scheme is developed for the H-J equation with homogeneous Hamiltonian.

(4) *Petrov–Galerkin Schemes.* A Galerkin approximation with least-squares stabilization is developed for the H-J equation with homogeneous Hamiltonian. A discontinuity capturing operator is added to resolve slope-discontinuous solutions.

(5) *Time Integration Strategies.* Time integration schemes for the positive coefficient and Petrov–Galerkin schemes are discussed. Monotonicity and positivity preserving one- and two-stage Runge–Kutta schemes are analyzed. Explicit and implicit space-time

formulations for the Petrov–Galerkin scheme are developed. The implicit Petrov–Galerkin scheme yields a formulation extendable to arbitrary order accuracy in both time and space.

(6) *Numerical Implementations.* Algorithmic implementations for the explicit positive coefficient and Petrov–Galerkin schemes are given.

(7) *Numerical Accuracy.* Accuracy of the positive coefficient and Petrov–Galerkin schemes is numerically evaluated for smooth and non-smooth solutions of the H-J equation.

(8) *Schemes for Curvature Flow.* A mass-lumped Galerkin approximation for mean curvature is considered and analyzed using maximum principle analysis.

(9) *Mesh Adaptivity.* Conformal and Steiner adaptivity strategies are defined and applied to H-J calculations.

(10) *Numerical Calculation of Curvature Flow on Adaptive Meshes.* Grayson’s 2D curvature flow problem is solved using conformal adaptation. Several 3D minimal surface calculations are presented.

(11) *Nonconvex Hamiltonians.* The performance of numerical schemes for nonconvex Hamiltonian problems arising in semiconductor etching and deposition is qualitatively evaluated.

## 2. NUMERICAL FUNDAMENTALS FOR THE H-J AND LEVEL SET EQUATIONS ON TRIANGULATED DOMAINS

Consider the following specialized form of the H-J equation (1)

$$\begin{aligned} u_t + H(\nabla u) &= f(x), & (x, t) \in \Omega \times \mathbf{R}^+ \\ u(x, 0) &= u_0(x) \end{aligned} \quad (7)$$

so that the Eikonal equation (3) can also be modeled by dropping the time derivative term. Let  $\mathcal{T}$  denote a triangulation set in  $\mathbf{R}^d$ ,  $\mathcal{T} = \{T_1, T_2, \dots, T_{|\mathcal{T}|}\}$ , composed of simplices covering  $\Omega$  such that  $\mathcal{T} = \cup T_j$ ,  $T_i \cap T_j = \emptyset$  for  $i \neq j$ . We will also refer to the vertex set  $V = \{v_1, v_2, \dots, v_{|V|}\}$ . Unless otherwise stated, the solution on  $\mathcal{T}$  is approximated using a standard piecewise linear finite element subspace denoted by  $\mathcal{V}_1^h$ . For  $(x, t) \in \mathcal{V}_1^h \times \mathbf{R}^+$ , the numerical solution at vertex  $v_j$  at a time  $n\Delta t$  is denoted by  $u_j^n$ . Next, consider numerical approximations of Eq. (7) of the form

$$u_j^{n+1} = u_j^n - \Delta t \mathcal{H}_j(\nabla u_1^n, \nabla u_2^n, \dots, \nabla u_{|\mathcal{T}|}^n, x) \quad (8)$$

with

$$\mathcal{H}_j(\nabla u_1, \nabla u_2, \dots, \nabla u_{|\mathcal{T}|}, x) = \frac{\sum_{l=1}^{|\mathcal{T}|} \alpha_j^l (H(\nabla u) - f(x))_{T_l}}{\sum_{l=1}^{|\mathcal{T}|} \alpha_j^l \text{meas}(T_l)}, \quad \alpha_j^l \geq 0, \quad (9)$$

where  $(\cdot)_{T_l} = \int_{T_l} (\cdot) d\Omega$ . As will be shown,  $\alpha_j^l = \alpha_j(\nabla u_l)$  are bounded, positive weighting coefficients computed at a simplex  $T_l$  which contribute to the numerical Hamiltonian  $\mathcal{H}$  at vertex  $v_j$ . In developing numerical discretizations, we follow Crandall and Lions [11], Osher and Sethian [22], and Abgrall [1] in proposing design criterion that reflect properties of the underlying H-J differential equation (without source term):

(1) (Consistency). If  $u(x, t)$  varies linearly in space and time, consistency dictates that the numerical scheme (8) must yield exact evolution:

$$u(x_j, n \Delta t) = u_0(x_j) - n \Delta t H(\nabla u_0). \quad (10)$$

In our formulations this design criteria is always satisfied, since the assumed form of the numerical Hamiltonian (9) satisfies

$$\mathcal{H}(\nabla u, \nabla u, \dots, \nabla u) = H(\nabla u) \quad (11)$$

and it can be shown that at least one nonzero  $\alpha_j^l$  exists for each interior vertex  $v_j$  location.

(2) (Monotonicity). Crandall and Lions [11] have previously shown that consistent, monotone schemes for the H-J equations are stable and converge to the correct viscosity limit solution. Monotonicity as used here can be defined in terms of order preservation, i.e.,

$$u^n \geq v^n \quad \text{implies} \quad u^{n+1} \geq v^{n+1}. \quad (12)$$

For self-maps of the form

$$u_j^{n+1} = u_j^n - \Delta t G_j(u^n), \quad (13)$$

it is sufficient to show that

$$\frac{\partial G_j}{\partial u_i} \leq 0 \quad \text{and} \quad 0 \leq \Delta t \leq \left( \frac{\partial G_j}{\partial u_j} \right)^{-1}, \quad \forall i, j \in \{1, 2, \dots, |V|\}, i \neq j \quad (14)$$

for the scheme to be monotone. In later sections, this design criterion is relaxed in favor of the weaker *positivity* condition,

$$u_j^{n+1} = u_j^n - \Delta t \sum_{i=1, i \neq j}^{|V|} \beta_j^i(u^n) (u_j^n - u_i^n) \quad (15)$$

with

$$\beta_j^i \geq 0 \quad \text{and} \quad 0 \leq \Delta t \leq \left( \sum_{i=1, i \neq j}^{|V|} \beta_j^i \right)^{-1}, \quad \forall j \in \{1, 2, \dots, |V|\}. \quad (16)$$

Both the monotone and positive schemes satisfy a global discrete maximum principle

$$\min_{i=1,2,\dots,|V|} u_i^n \leq u_j^{n+1} \leq \max_{i=1,2,\dots,|V|} u_i^n, \quad \forall j \in \{1, 2, \dots, |V|\} \quad (17)$$

as well as the local maximum principle

$$\min_{i \in \text{supp}_j} u_i^n \leq u_j^{n+1} \leq \max_{i \in \text{supp}_j} u_i^n, \quad \forall j \in \{1, 2, \dots, |V|\}, \quad (18)$$

where  $\text{supp}_j$  denotes the index support set for the discretized scheme at vertex  $v_j$ .

In the following sections, a procedure is described for calculating the  $\alpha_j^l$  coefficients so that monotonicity and/or positivity properties are obtained for specific Hamiltonians.

### 3. DEVELOPMENT OF NUMERICAL SCHEMES FOR APPROXIMATING H-J EQUATIONS

#### 3.1. A Monotone Scheme for $H(\nabla u)$ Hamiltonians

Consider a  $d$ -dimensional simplex  $T$  with linearly varying  $u(x)$  uniquely defined by vertex values  $u_i$  and linear basis functions  $N_i(x)$  satisfying  $N_i(x_j) = \delta_{ij}$

$$u(x) = \sum_{i=1}^{d+1} N_i(x) u_i. \quad (19)$$

The gradient  $\nabla u$  in simplex  $T$  is readily computed in terms of gradients of the basis functions

$$\nabla u = \sum_{i=1}^{d+1} \nabla N_i u_i. \quad (20)$$

To gain a better geometric intuition, the gradient formula is rewritten in terms of inward pointing normals  $\vec{n}_i$  scaled by the measure of the  $(d-1)$ -facet opposite vertex  $v_i$  in the simplex, i.e.,

$$\nabla u = \frac{1}{d \text{ meas}(T)} \sum_{i=1}^{d+1} \vec{n}_i u_i. \quad (21)$$

Note that

$$\sum_{i=1}^{d+1} \vec{n}_i = 0 \quad (22)$$

due to the scaling of normals and the fact that the surface of a simplex is closed. It is also useful to define the gradient of the Hamiltonian with respect to the gradient arguments

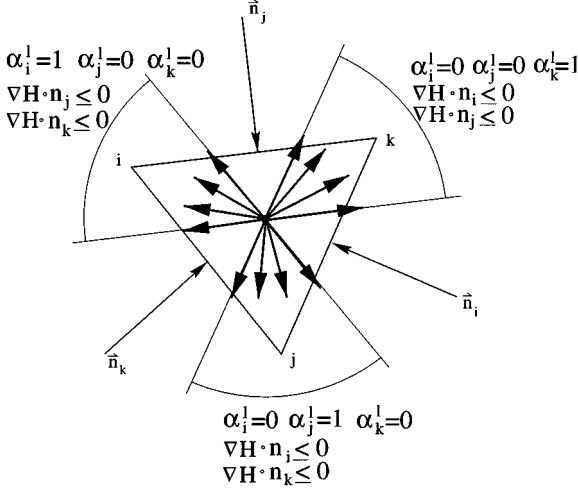
$$\nabla H \equiv \begin{pmatrix} \frac{\partial H}{\partial u_{x_1}} \\ \frac{\partial H}{\partial u_{x_2}} \\ \vdots \\ \frac{\partial H}{\partial u_{x_d}} \end{pmatrix}. \quad (23)$$

Next form  $(H(\nabla u))_T = \int_T H(\nabla u) d\Omega$ , the Hamiltonian integrated in a simplex, and differentiate with respect to the simplex unknowns

$$\frac{\partial (H(\nabla u))_T}{\partial u_i} = \frac{1}{d} (\nabla H \cdot \vec{n}_i), \quad i = 1, 2, \dots, d+1. \quad (24)$$

Due to the scaling of normals and the constancy of  $H(\nabla u)$  within a linear simplex

$$\sum_{i=1}^{d+1} \frac{\partial (H(\nabla u))_T}{\partial u_i} = 0. \quad (25)$$



**FIG. 1.** Typical 2D simplex  $T_i$  showing inward pointing normals, index convention, and sector demarcation.

When combined with the monotonicity conditions (14), certain orientations of  $\nabla H$  relative to the geometry of simplex  $T$  give rise to monotone updates as shown in Fig. 1 for a single isolated two-dimensional simplex.

$$\nabla H \cdot n_j \leq 0, \nabla H \cdot n_k \leq 0.$$

$$u_i^{n+1} = u_i^n - \Delta t \frac{(H(\nabla u^n))_T}{\text{meas}(T)} \quad (26)$$

$$\nabla H \cdot n_i \leq 0, \nabla H \cdot n_k \leq 0.$$

$$u_j^{n+1} = u_j^n - \Delta t \frac{(H(\nabla u^n))_T}{\text{meas}(T)} \quad (27)$$

$$\nabla H \cdot n_i \leq 0, \nabla H \cdot n_j \leq 0.$$

$$u_k^{n+1} = u_k^n - \Delta t \frac{(H(\nabla u^n))_T}{\text{meas}(T)}. \quad (28)$$

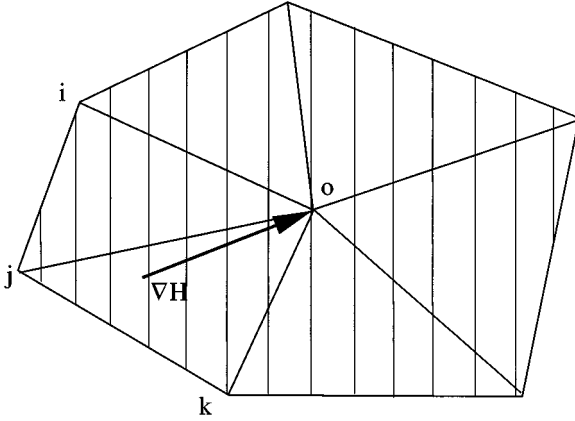
More generally, several simplices may contribute to the update at a single vertex. In this case the generalized numerical Hamiltonian formula (9) can be used

$$u_j^{n+1} = u_j^n - \Delta t \frac{\sum_{l=1}^{|T|} \alpha_j^l (H(\nabla u^n))_{T_l}}{\sum_{l=1}^{|T|} \alpha_j^l \text{meas}(T_l)}, \quad (29)$$

where

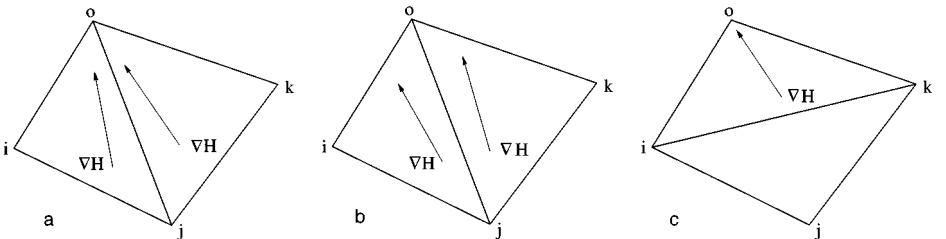
$$\alpha_j^l = \begin{cases} 1, & \nabla H \cdot n_k \leq 0, \quad k = 1, 2, \dots, d+1, \quad j \neq k, \\ 0, & \text{otherwise,} \end{cases} \quad x \in T_l. \quad (30)$$

Observe that consistency is achieved, given the form of the numerical Hamiltonian, if it can be shown that there exists at least one simplex producing a nonzero  $\alpha_j^l$  coefficient when  $u(x)$  varies linearly over the entire support of the scheme. From Fig. 2 it is clear that this must always happen. Consider a constant gradient vector  $\nabla H$  located at a vertex



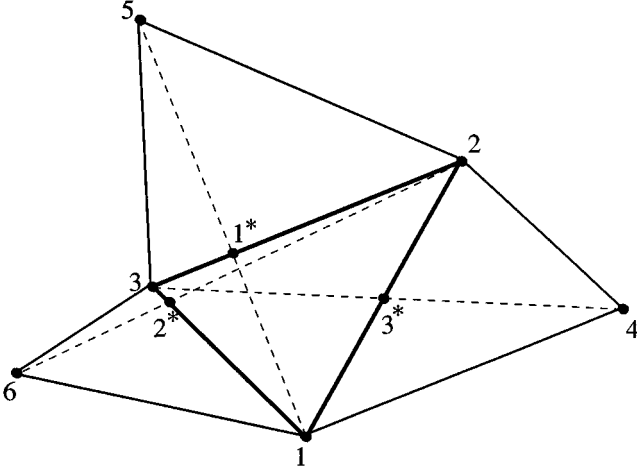
**FIG. 2.** Triangles surrounding  $v_0$  with linearly varying  $u(x)$ . Isolevels of constant  $u(x)$  are shown as solid lines and the arrow denotes the constant gradient vector  $\nabla H$  at vertex  $v_0$ .

$v_0$  to be updated. By simply tracing backwards along the gradient vector into a simplex  $T_m$  surrounding  $v_0$ , a simplex has been found which contributes a nonzero  $\alpha_0^m$ . Unfortunately, the scheme suffers from a serious flaw. The numerical Hamiltonian fails to be a Lipschitz continuous function of its arguments. This non-Lipschitzian behavior is shown in Figs. 3a–3b. The numerical Hamiltonian varies discontinuously with certain small perturbations in its arguments. This can greatly reduce the accuracy of solutions. This will be demonstrated in a numerical example given later. To overcome this problem, one might consider an edge flipping strategy as illustrated in Fig. 3c. By flipping the position of the diagonal for the triangle pair, vertex  $v_0$  obtains a single update with a numerical Hamiltonian which now varies continuously with small perturbations in data. In principle, this edge flipping strategy could be implemented in a computer code by testing continuity of the numerical Hamiltonian and edge flipping when appropriate. Rather than implementing this elaborate procedure, a *virtual* edge flipping strategy has been developed as depicted in Fig. 4. Referring to this figure, consider the triangle  $T(v_1, v_2, v_3)$  and the edge  $e(v_1, v_2)$ . If the triangle pair  $(T(v_1, v_2, v_3), T(v_2, v_1, v_4))$  adjacent to  $e(v_1, v_2)$  forms a convex quadrilateral, one can consider the possibility of swapping the diagonal position to form  $e(v_3, v_4)$ . Given that the solution unknowns are associated with vertices of the mesh, a unique piecewise continuous linear representation of the solution can be associated with the two new triangles  $T'(v_1, v_4, v_3)$  and  $T'(v_3, v_4, v_2)$ . With respect to the original triangle  $T(v_1, v_2, v_3)$ ,



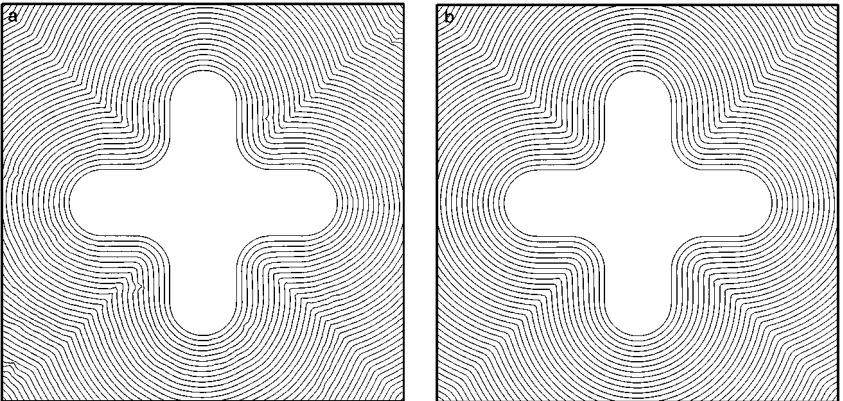
**FIG. 3.** Non-Lipschitzian behavior of the monotone scheme with changing gradient direction (left and middle figures). Diagonal flipping maneuver to improve Lipschitz continuity of the numerical Hamiltonian. (a) Two  $v_0$  updates; (b) no  $v_0$  update; (c) one  $v_0$  update.



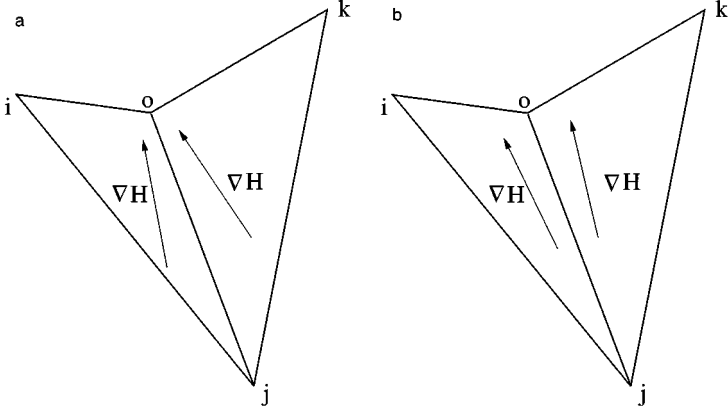


**FIG. 4.** Virtual edge flipping strategy. Given the triangle  $T(v_1, v_2, v_3)$  with vertex solution values  $u_1, u_2$ , and  $u_3$ , additional solution values  $u_1^*, u_2^*, u_3^*$  and positions are computed based on (virtual) diagonal flipped edges  $e(v_3, v_4), e(v_1, v_5), e(v_2, v_6)$ .

these two new linear functions can be encoded by simply computing the solution  $u_3^*$  and its location  $e(v_3, v_4) \cap e(v_1, v_2)$ . Next, the general monotone update formula (29) is used for each triangle in its original configuration as well as in diagonally flipped positions. In Fig. 4, three possible updates at  $v_1$  can be computed that originate from  $T(v_1, v_2, v_3)$  as well as  $T(v_1, v_2, v_2^*)$  and  $T(v_1, v_3^*, v_3)$  with similar permutations for other vertices of  $T$ . In performing calculations using virtual edge flipping, all possible contributions have been summed. Figure 5 shows a sample level set calculation described in full detail in a later section. The correct solution consists of Euclidean distance contours from the innermost star-shaped figure. The left figure shows the baseline monotone scheme without virtual edge flipping. Several noticeable defects are present in the numerical solution originating from non-Lipschitz behavior of the numerical Hamiltonian. The right figure shows the same test problem computed with the monotone update scheme with virtual edge flipping. The level set contours now approximate the true solution quite well. Note that edge flipping



**FIG. 5.** Level set solutions obtained (a) without and (b) with virtual edge flipping.



**FIG. 6.** Non-Lipschitzian behavior of the monotone scheme which cannot be repaired by edge flipping. (a) two  $v_0$  updates; (b) no  $v_0$  update.

can be problematic since it requires that adjacent triangle pairs form convex quadrilaterals. For example, the triangle pair shown in Fig. 6 forms a nonconvex quadrilateral, hence the Lipschitz discontinuous Hamiltonian cannot be repaired by edge flipping. Clearly, this suggests the sufficient but not necessary condition for edge flipping that the triangulation be acute. The technique also extends to higher space dimensions, albeit with more update possibilities for each simplex.

The edge flipping technique required for the monotone scheme introduces an undesirable aspect to the scheme since adjacent elements must be interrogated. When combined with the geometric convexity constraints associated with triangle pairs, we are motivated to seek alternate discretization design criteria applicable to general triangulations with computations which remain local to a simplex. In the next section, a strategy is developed which achieves these goals.

### 3.2. A Positive Scheme for Homogeneous $H(\nabla u)$ on Triangulated Domains

To gain Lipschitz continuity of the numerical Hamiltonian without resorting to edge flipping, the condition of monotonicity is relaxed in favor of a positivity condition. We tacitly assume that the Hamiltonian  $H(\nabla u)$  and its gradient are smooth, continuous functions of  $\nabla u$ . In addition, the data itself is assumed to have a small Lipschitz constant between elements. In this framework, it will be sufficient to show that the  $d + 1$  coefficients  $\alpha'_j$  are Lipschitz continuous functions of the gradient vector  $\nabla u$  for the simplex  $T_l$ . For accuracy reasons  $\alpha'_j$  should also be bounded and positive. Note that since monotone schemes depend fundamentally on the quantity

$$\frac{\partial(H(\nabla u))_T}{\partial u_i} = \frac{1}{d} \nabla H \cdot \vec{n}_i,$$

in principle any new positive scheme should also depend on this quantity. This would ensure that monotone updates are recovered whenever possible.

Recall Euler's theorem for homogeneous functions: if  $F(u, v) : \mathbf{R} \times \mathbf{R} \mapsto \mathbf{R}$  is homogeneous of degree  $p$  then  $F(tu, tv) = t^p F(u, v)$  and

$$pF(u, v) = \frac{\partial F}{\partial u} u + \frac{\partial F}{\partial v} v. \quad (31)$$

Our attention focuses on Hamiltonians  $H(\nabla u)$  which are homogeneous functions of degree  $p$  in  $\nabla u$ . In this case, Euler's theorem yields the following relation for a simplex  $T$  in  $\mathbf{R}^d$

$$(H(\nabla u))_T = \frac{1}{p} \nabla H \cdot \nabla u = \frac{1}{dp} \sum_{i=1}^{d+1} (\nabla H \cdot \vec{n}_i) u_i. \quad (32)$$

Next define

$$K_i = \frac{\nabla H \cdot \vec{n}_i}{dp} \quad (33)$$

so that (32) can be expressed in the following canonical form

$$(H(\nabla u))_T = \sum_{i=1}^{d+1} K_i u_i \quad \text{with} \quad \sum_{i=1}^{d+1} K_i = 0. \quad (34)$$

Once written in canonical form, we can draw upon a number of well-known techniques for constructing positive coefficient schemes for conservation laws due to Roe [23, 24] and Deconinck *et al.* [13]. The basis for these schemes is the following straightforward manipulation of the canonical form

$$\begin{aligned} \sum_{i=1}^{d+1} K_i u_i &= \sum_{j=1}^{d+1} K_j^- u_j + \sum_{i=1}^{d+1} K_i^+ u_i \\ &= \left( \sum_{l=1}^{d+1} K_l^+ \right)^{-1} \left( \sum_{i=1}^{d+1} K_i^+ \right) \sum_{j=1}^{d+1} K_j^- u_j - \left( \sum_{l=1}^{d+1} K_l^+ \right)^{-1} \left( \sum_{j=1}^{d+1} K_j^- \right) \sum_{i=1}^{d+1} K_i^+ u_i \\ &= \sum_{i=1}^{d+1} K_i^+ \left( \sum_{l=1}^{d+1} K_l^+ \right)^{-1} \sum_{j=1}^{d+1} K_j^- (u_j - u_i). \end{aligned} \quad (35)$$

In deriving this formula, the useful identity  $\sum_{j=1}^{d+1} K_j = \sum_{j=1}^{d+1} K_j^+ + \sum_{j=1}^{d+1} K_j^- = 0$  has been used. Equation (35) motivates a decomposition of the Hamiltonian in a simplex  $T$  of the form

$$(H(\nabla u))_T = \sum_{i=1}^{d+1} \delta_i \quad (36)$$

with

$$\delta_i = K_i^+ \left( \sum_{l=1}^{d+1} K_l^+ \right)^{-1} \sum_{j=1}^{d+1} K_j^- (u_j - u_i). \quad (37)$$

When written in matrix form

$$\begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{d+1} \end{pmatrix} = \begin{bmatrix} \ddots & & - \\ & + & \\ - & & \ddots \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{d+1} \end{pmatrix} \quad (38)$$

it becomes clear that the decomposition produces a matrix sign pattern with zero row sum useful in constructing a positive coefficient schemes satisfying Eq. (15). To do so simply let

$$\alpha_j^l = \frac{\delta_j}{(H(\nabla u))_{T_l}} \quad (39)$$

and insert into our prototype numerical Hamiltonian

$$\mathcal{H}_j(\nabla u_1, \nabla u_2, \dots, \nabla u_{|T|}) = \frac{\sum_{l=1}^{|T|} \alpha_j^l (H(\nabla u))_{T_l}}{\sum_{l=1}^{|T|} \alpha_j^l \text{meas}(T_l)} = \frac{\sum_{l=1}^{|T|} \delta_j^l}{\sum_{l=1}^{|T|} \alpha_j^l \text{meas}(T_l)}. \quad (40)$$

Note that by construction  $\sum_{j=1}^{d+1} \alpha_j^l = 1$ . It is not difficult to show that for certain Hamiltonians (e.g.,  $H(\nabla u) = |\nabla u|^p$ ) and acute triangulations,  $\alpha_j^l$  are nonnegative and hence bounded. In this case the conditions for a positive coefficient scheme are obtained:

$$u_j^{n+1} = u_j^n - \Delta t \sum_{i=1, i \neq j}^{|V|} \beta_j^i(u^n) (u_j^n - u_i^n) \quad (41)$$

with

$$\beta_j^i \geq 0 \quad \text{and} \quad 0 \leq \Delta t \leq \left( \sum_{i=1, i \neq j}^{|V|} \beta_j^i \right)^{-1}, \quad j = 1, 2, \dots, |V|. \quad (42)$$

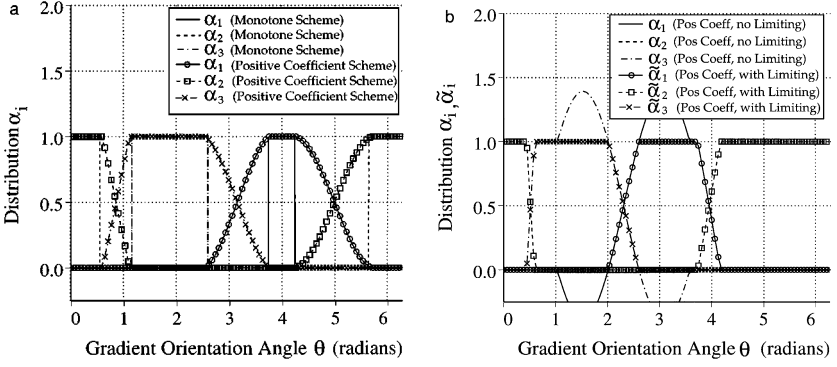
For general (obtuse) triangulations and homogeneous Hamiltonians, additional nonlinearity must be introduced into the formulation so that the  $\alpha_j^l$  coefficients remain positive. Deconinck [13], in the context of conservation law equation discretization, suggests a simple nonlinear modification of  $\alpha_j^l$  to obtain a positive coefficient scheme for general triangulations:

$$\tilde{\alpha}_j^l = \frac{\max(0, \alpha_j^l)}{\sum_{k=1}^{d+1} \max(0, \alpha_k^l)}. \quad (43)$$

The inclusion of this modification produces a robust scheme for the H-J equations (with source term), i.e.,

$$\mathcal{H}_j(\nabla u_1, \nabla u_2, \dots, \nabla u_{|T|}, x) = \frac{\sum_{l=1}^{|T|} \tilde{\alpha}_j^l (H(\nabla u) - f(x))_{T_l}}{\sum_{l=1}^{|T|} \tilde{\alpha}_j^l \text{meas}(T_l)}. \quad (44)$$

By construction,  $\tilde{\alpha}_j^l \geq 0$  and  $\sum_{j=1}^{d+1} \tilde{\alpha}_j^l = 1$  for any orientation of  $\nabla H$ . Note that this nonlinear modification still permits the scheme to be written in the positive coefficient form (41). To address continuity of the distribution coefficients with respect to  $\nabla u$  in a simplex, observe that if  $K_i \geq 0$ ,  $K_j, K_k \leq 0$ ,  $j, k \neq i$  (or any permutation of  $\{i, j, k\}$ ) then the monotone scheme update depicted in Fig. 1 is always obtained. As shown in Fig. 7, the positive coefficient scheme smoothly fills in the “transition gaps” present in the monotone update scheme of Subsection 3.1 where all  $\alpha_j^l$  are identically zero for the simplex  $T_l$ . This follows directly from the continuity of the  $\delta_j^l$ , hence  $\alpha_j^l$ , coefficients at transition gap boundaries and the implied smoothness between gap boundaries under mild assumptions on the  $H(\nabla u)$  and  $\nabla H(\nabla u)$  with respect to their gradient argument. The effect of the limiting is to bound the distribution coefficients in the interval  $[0, 1]$ . Apart from degeneracies that can occur



**FIG. 7.** Distribution coefficients for  $H(\nabla u) = |\nabla u|$  with  $\nabla u = (\cos \theta, \sin \theta)^T$ . Monotone and positive coefficient schemes on (a) nearly equilateral simplex and positive coefficient scheme with and (b) without limiting on obtuse simplex.

when the transition gap shrinks to zero size, the transition gaps are filled smoothly by both the limited and unlimited positive coefficient scheme.

Repeating the example calculation of the previous section shown in Fig. 4, Fig. 8 compares the monotone scheme with virtual edge flipping and the present positive coefficient scheme. The resulting solutions compare favorably. Note that the positive coefficient scheme requires approximately 20% less computing time owing to the local computation and simplified updating. Accuracy of this scheme is evaluated in a later section.

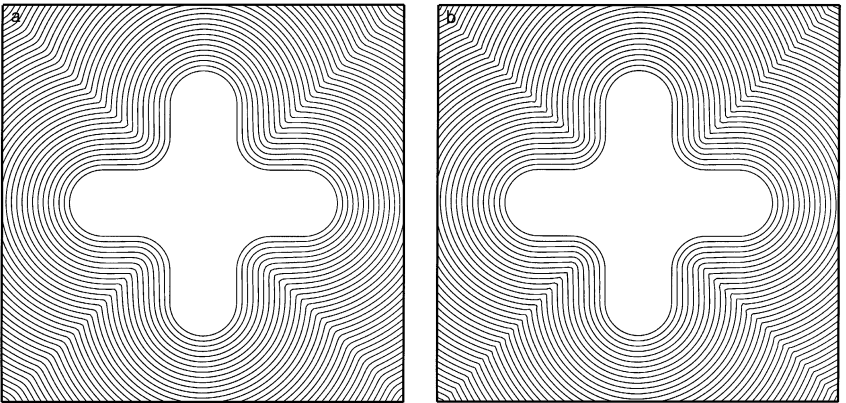
### 3.3. A Petrov–Galerkin Formulation

Next, consider a stabilized finite element approximation for the H-J equation

$$u_t + H(\nabla u) = 0. \quad (45)$$

The Hamiltonian  $H(\nabla u)$  is assumed to homogeneous of degree  $p$

$$H(\nabla u) = p^{-1} \nabla H \cdot \nabla u \quad (46)$$



**FIG. 8.** Level set solution obtained using monotone scheme with (a) virtual edge flipping and (b) positive coefficient scheme.

for positive  $p$ . The formulation considered here is inspired by the stabilized Petrov–Galerkin methods developed for conservation law equations, see Johnson [16] for a detailed discussion. Let  $\mathcal{P}_k$  denote  $k$ th order interpolation polynomials in each simplex  $T$  and  $\mathcal{D}^n = \Omega \times I^n$  the space-time slab with  $I^n = [t_+^n, t_-^{n+1}]$ . For a given positive integer  $k$  define the usual trial space

$$\mathcal{V}_n^h = \{v^h | v^h \in C^0(\mathcal{D}^n), v^h|_{T \times I^n} \in \mathcal{P}_k(T \times I^n) \forall T \in \mathcal{T}\}, \quad (47)$$

i.e.,  $\mathcal{V}_n^h$  is the space of continuous piecewise polynomial functions of degree  $k$  for the space-time slab  $\mathcal{D}^n$ . Note that between space-time slabs  $\mathcal{V}_n^h$  is allowed to jump discontinuously, i.e.,  $u^h(x, t_-^n) \neq u^h(x, t_+^n)$ . Next define the inflow/outflow portions of the boundary  $\Gamma$ :

$$\Gamma_+ = \Gamma \setminus \Gamma_- = \{x \in \Gamma : n(x) \cdot \nabla H \geq 0\}. \quad (48)$$

From this, the following stabilized finite element approximation with weakly imposed boundary conditions can be stated.

Find  $u^h \in \mathcal{V}_n^h$  such that for all  $w^h \in \mathcal{V}_n^h$

$$B(u^h, w^h)_{gal} + B(u^h, w^h)_{ls} + B(u^h, w^h)_{bc} = 0 \quad (49)$$

with

$$\begin{aligned} B(u, w)_{gal} &= \int_{I^n} \int_{\Omega} w(u_t + H(\nabla u)) d\Omega dt \\ B(u, w)_{ls} &= \int_{I^n} \int_{\Omega} (w_t + p^{-1}(\nabla H \cdot \nabla w)) \tau (u_t + p^{-1}(\nabla H \cdot \nabla u)) d\Omega dt \\ B(u, w)_{bc} &= \int_{I^n} \int_{\Gamma_-} w(g - u) p^{-1}(\nabla H \cdot n) d\Gamma dt + \int_{\Omega} w(t_+^n)(u(t_+^n) - u(t_-^n)) d\Omega \end{aligned} \quad (50)$$

for  $\tau > 0$  and  $\nabla H = \nabla H(\nabla u)$  everywhere in (50). Note that even though  $w^h$  and  $u^h$  reside in the same space of functions, the scheme still is of Petrov–Galerkin type since the Galerkin and least-squares variational operators can be combined into a single Galerkin variational operator with perturbed test space  $w'$ , i.e.,

$$B_{gal}(u, w') = B_{gal}(u, w) + B_{ls}(u, w)$$

with

$$w' = w + \tau p^{-1}(\nabla H \cdot \nabla w).$$

The motivation for the least-squares stabilization  $B(u, w)_{ls}$  comes from looking at the variation of the quadratic potential

$$\mathcal{P}(u) = \int_{\mathcal{D}^n} \frac{1}{2} \tau (u_t + p^{-1}(\nabla H \cdot \nabla u))^2 d\mathcal{D} \quad (51)$$

for constant  $\nabla H$

$$\delta \mathcal{P} = \lim_{\sigma \rightarrow 0} \frac{d}{d\sigma} \mathcal{P}(u + \sigma w) = B(u, w)_{ls}, \quad \forall w \in \mathcal{V}. \quad (52)$$

Note that  $B(u, w)_{ls}$  removes energy from the system since

$$B(u, u)_{ls} \geq 0 \quad (53)$$

and

$$\frac{\partial}{\partial t} \int_{\Omega} (u/2)^2 d\Omega + \int_{\Omega} u H(\nabla u) d\Omega + B(u, u)_{ls} + B(u, u)_{bc} = 0. \quad (54)$$

Removing the assumption of constant  $\nabla H$  in the calculation of the first variation yields

$$\delta \mathcal{P} = B(u, w)_{ls} + p^{-1} \int_{\Omega} (u_t + p^{-1}(\nabla H \cdot \nabla u)) \tau (\nabla^T w \text{Hess}(H) \nabla u) d\Omega, \quad (55)$$

where  $\text{Hess}(H)$  denotes the Hessian matrix containing second derivatives of  $H$ . Note the appearance of a new term in the first variation statement. If  $H(\nabla)$  is homogeneous of degree  $p$ , it can be shown that

$$\text{Hess}(H) \nabla u = p(p-1) \nabla H \quad (56)$$

so that

$$\delta P = B(u, w)_{ls} + (p-1) \int_{\Omega} (u_t + p^{-1}(\nabla H \cdot \nabla u)) \tau (\nabla H \cdot \nabla w) d\Omega. \quad (57)$$

Although similar to the standard least-squares term, the time derivative in this new term makes the sign of the associated energy undetermined. Fortunately, for many physically relevant Hamiltonians of interest,  $H(\nabla u)$  is homogeneous of degree one ( $p = 1$ ) in which case the second integral term vanishes identically and Eq. (51) is recovered. For this reason, we defer further consideration of this new term.

Our choice of the parameter  $\tau$  is motivated from standard practice for conservation laws, see Johnson [16] and references therein,

$$\tau = \left( \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2|\nabla H|}{ph} \right)^2 \right)^{-1/2}, \quad h \equiv \text{diam}(T) \quad (58)$$

with the  $\Delta t$  term omitted for discretizations of the Eikonal equation. The next result shows that for any  $\tau > 0$ , the least-squares term removes energy from the system.

**ENERGY BALANCE.** *The Petrov–Galerkin scheme (50) exhibits the following energy balance:*

$$\begin{aligned} & \frac{1}{2} \|u(t_-^N)\|_{\Omega}^2 + \frac{1}{2} \sum_{n=0}^{N-1} \|u(t_+^n) - u(t_-^n)\|_{\Omega}^2 + \sum_{n=0}^{N-1} \|\sqrt{\tau}(u_t + H(\nabla u))\|_{\Omega \times I^n}^2 \\ & + \frac{1}{2} \sum_{n=0}^{N-1} \int_{\Gamma} \int_{I^n} u^2 p^{-1} |\nabla H \cdot n| dx dt + \frac{1}{2} \sum_{T \in \mathcal{T}} \int_{I^n} \int_T u^2 (\nabla \cdot \nabla H) dx dt \\ & + \frac{1}{2} \sum_{e \in E, x(e) \notin \Gamma} \int_{I^n} \int_e u^2 [\nabla H] \cdot n dx dt = \frac{1}{2} \|u(t_-^0)\|_{\Omega}^2 \\ & + \sum_{n=0}^{N-1} \int_{\Gamma^-} \int_{I^n} u g p^{-1} |\nabla H \cdot n| dx dt. \end{aligned} \quad (59)$$

*Proof.* To construct the energy at  $t_-^N = t_-^0 + \sum_{n=0}^{N-1} I^n$  let  $w = u$  and evaluate the various integrals. The least-squares stabilization term produces a pure quadratic form and will not be discussed further. Combining the first term

$$\int_{\Omega} \int_{I^n} uu_t dt dx = \frac{1}{2} \int_{\Omega} [u^2(t_-^{n+1}) - u^2(t_+^n)] dx \quad (60)$$

and the jump integral

$$\int_{\Omega} u(t_+^n)[u(t_+^n) - u(t_-^n)] dx = \frac{1}{2} \int_{\Omega} [u(t_+^n) - u(t_-^n)]^2 dx + \frac{1}{2} \int_{\Omega} [u^2(t_+^n) - u^2(t_-^n)] dx \quad (61)$$

yields

$$\begin{aligned} \int_{\Omega} \int_{I^n} uu_t dt dx + \int_{\Omega} u(t_+^n)[u(t_+^n) - u(t_-^n)] dx \\ = \frac{1}{2} \int_{\Omega} [u(t_+^n) - u(t_-^n)]^2 dx + \frac{1}{2} \int_{\Omega} [u^2(t_-^{n+1}) - u^2(t_-^n)] dx \end{aligned} \quad (62)$$

for a single time slab. Next rewrite the spatial integral term

$$\begin{aligned} \int_{I^n} \int_{\Omega} u(\nabla H \cdot \nabla u) dx dt &= \frac{1}{2} \int_{I^n} \int_{\Omega} (\nabla H \cdot \nabla u^2) dx dt \\ &= \frac{1}{2} \sum_{T \in \mathcal{T}} \int_{I^n} \int_T (\nabla H \cdot \nabla u^2) dx dt. \end{aligned} \quad (63)$$

From Green's theorem

$$\int_T \nabla H \cdot \nabla u^2 dx = \int_T u^2(\nabla \cdot \nabla H) dx + \int_{\partial T} u^2(\nabla H \cdot n) dx \quad (64)$$

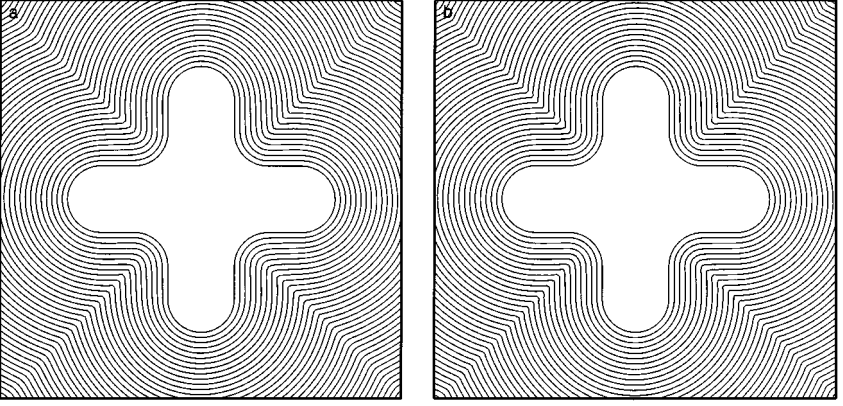
so that

$$\begin{aligned} \int_{I^n} \int_{\Omega} u(\nabla H \cdot \nabla u) dx dt - \int_{I^n} \int_{\Gamma^-} u^2(\nabla H \cdot n) dx dt \\ = \frac{1}{2} \sum_{T \in \mathcal{T}} \int_{I^n} \int_T u^2(\nabla \cdot \nabla H) dx dt + \frac{1}{2} \sum_{e \in E, x(e) \notin \Gamma} \int_{I^n} \int_e u^2[\nabla H] \cdot n dx dt \\ + \frac{1}{2} \int_{I^n} \int_{\Gamma} u^2|\nabla H \cdot n| dx dt, \end{aligned}$$

where  $[\nabla H] \cdot n = (\nabla H_R - \nabla H_L) \cdot n$  if the normal vector for edge  $e$  points from  $T_L$  to  $T_R$ . Inserting the finite-dimensional subspace  $u^h$  and summing over time slabs yields the energy balance equation (59).  $\blacksquare$

This result, while perhaps of limited interest, shows how the least-squares term, boundary conditions, and time-discontinuous solution representation each improve energy stability of the scheme. In the trivial case  $\nabla H = \text{constant}$ , two terms drop out and the energy balance formally bounds energy at time  $t^N$  in terms of inflow boundary data and initial data at time  $t_-^0$ .





**FIG. 9.** Level set solution obtained using monotone scheme with (a) virtual edge flipping and (b) Petrov–Galerkin scheme.

### 3.4. The Petrov–Galerkin Discontinuity Capturing Operator

Although least-square stabilization is sufficient for many problems of interest, the operator does not control solution oscillations as do monotone and positive coefficient discretizations. Consequently, small solution oscillations are sometimes present near slope discontinuities. To remove these oscillations, a discontinuity capturing operator is employed that is similar to that commonly used for conservation law discretizations, see, for example, Hansbo [15]:

$$B(u, w)_{dc} = \int_T v(u)(\nabla w \cdot \nabla u) \, dx, \quad v = \frac{\sigma_{dc} h |u_t + H(\nabla u)|}{(|u_t|^q + |H(\nabla u)|^q + \epsilon^q)^{1/q}}. \quad (65)$$

In this formula,  $\epsilon$  is a small computer dependent parameter and  $q$  a parameter with typical values  $q \in \{1, 2, \infty\}$ . Observe that this form of discontinuity capturing operator can be viewed as a form of nonlinear artificial viscosity which retains the weighted residual property of the basic finite element method, i.e., the viscosity coefficient vanishes when the exact solution is inserted into the discontinuity capturing operator. More generally, the viscosity coefficient is small when the Hamilton–Jacobi equation is well-approximated in an element. The mesh width  $h$  is included in the definition so that the overall accuracy of the Galerkin least-squares method for smooth solutions is retained. In a later section, the numerical calculation of nonconvex Hamiltonians arising in etching and deposition modeling is considered. In this context, discontinuity capturing operators such as (65) have proven invaluable for accurately computing these solutions. For completeness, the example calculation shown in Figs. 4 and 8 is repeated for the Petrov–Galerkin discretization with discontinuity capturing term ( $\sigma_{dc} = .1$ ). Figure 9 compares the monotone scheme with edge flipping and the present Petrov–Galerkin scheme. The solutions are visually indistinguishable.

## 4. TIME INTEGRATION SCHEMES

The H-J and level set equations both require accurate time integration schemes. Ideally, these time integration schemes should not destroy the monotonicity or positivity properties of the spatial discretization under some CFL-like restriction on the time step parameter.

Specifically, single- and two-stage Runge–Kutta time advancement schemes designed for this purpose are considered. Note that in solving the Eikonal equation, a fictitious time derivative is added to the equation so that steady state solutions are then sought. While not of optimal complexity when compared to the fast marching methods developed for the static Eikonal equation on Cartesian meshes [29], it does permit a valid study of the spatial accuracy of these discretizations.

Next, time discretization schemes for the Petrov–Galerkin scheme are considered. The formulation permits piecewise constant in time elements. When combined with mass lumping, a simple explicit scheme is produced. Higher order accuracy in time is naturally achieved using piecewise discontinuous polynomial approximations in time. This results in couple systems of equations for each space-time slab.

## 4.1. Explicit Time Stepping for Monotone and Positive Coefficient Schemes

### 4.1.1. Single-Stage Explicit Time Integration

The analysis is rather straightforward for the single-stage integration scheme. Recall the prototype self-map

$$u_j^{n+1} = u_j^n - \Delta t G_j(u^n). \quad (66)$$

This is a  $O(\Delta t)$  accurate integration in time for the equation  $u_t + G(u) = 0$ . As discussed earlier, strict order preservation yields the following CFL-like restriction on  $\Delta t$

$$0 \leq \Delta t \leq \left( \frac{\partial G_j}{\partial u_j} \right)^{-1}, \quad \forall j \in \{1, 2, \dots, |V|\}. \quad (67)$$

The positive coefficient scheme follows a similar argument. When the scheme is written in prototype coefficient form

$$u_j^{n+1} = u_j^n - \Delta t \sum_{i=1, i \neq j}^{|V|} \beta_j^i(u^n) (u_j^n - u_i^n), \quad (68)$$

positivity of coefficients in time<sup>1</sup> yields the following CFL-like time step restriction similar to the monotone scheme:

$$0 \leq \Delta t \leq \left( \sum_{i=1, i \neq j}^{|V|} \beta_j^i \right)^{-1}, \quad \forall j \in \{1, 2, \dots, |V|\}. \quad (69)$$

Finally, observe that these CFL-like conditions for monotonicity and/or positivity both imply stability in a maximum norm since both imply a global discrete maximum principle of the form

$$\min_{i=1,2,\dots,|V|} u_i^n \leq u_j^{n+1} \leq \max_{i=1,2,\dots,|V|} u_i^n, \quad \forall j \in \{1, 2, \dots, |V|\}. \quad (70)$$

<sup>1</sup> That is,  $u_j^{n+1}$  is a positive weighted combination of  $u^n$ .

#### 4.1.2. Two-Stage Time Integration

It is not difficult to improve the time integration accuracy to  $O(\Delta t^2)$  using the following well-known two-stage integration scheme which uses the single-stage scheme as a basic building block:

$$\begin{aligned} u_i^{n+1/2} &= u_i^n - \Delta t G_i(u^n) \\ u_i^{n+1} &= \frac{1}{2}(u_i^n + u_i^{n+1/2}) - \frac{1}{2}\Delta t G_i(u^{n+1/2}), \quad i = 1, 2, \dots, |V|. \end{aligned} \quad (71)$$

Furthermore, the next lemma proves that this time integration scheme preserves positive self-maps under a CFL-like condition equal to the single-stage time integration. This again implies stability in a maximum norm. This scheme has been analyzed in the context of total variation diminishing (TVD) preservation by Shu and Osher [32].

**LEMMA.** *If  $\mathcal{G}(u) : \mathbf{R}^{|V|} \rightarrow \mathbf{R}^{|V|}$  is a positive self-map which commutes with the addition of scalars, i.e.,  $\mathcal{G}(u + \omega) = \mathcal{G}(u) + \omega$  then the 2-stage time integration scheme Eq. (71) with  $\mathcal{G}(u) = u - \Delta t G(u)$  is a positive self-map which commutes with the addition of scalars.*

*Proof.* The time integration scheme is of the form

$$u^{n+1} = \frac{1}{2}(u^n + \mathcal{G}(u^n)), \quad (72)$$

where  $\mathcal{G}(u) = u - \Delta t G(u)$  so our attention focuses on the composition term  $\mathcal{G}(\mathcal{G}(u))$ . The analysis is straightforward when  $\mathcal{G}(u) : \mathbf{R}^{|V|} \rightarrow \mathbf{R}^{|V|}$  is an order preserving self-map. From order preservation, it follows that

$$u \geq v \text{ implies } \mathcal{G}(u) \geq \mathcal{G}(v) \text{ implies } \mathcal{G}(\mathcal{G}(u)) \geq \mathcal{G}(\mathcal{G}(v)) \quad (73)$$

so that  $\mathcal{G}(\mathcal{G}(u))$  is an order preserving self-map and consequently  $\frac{1}{2}(u + \mathcal{G}(\mathcal{G}(u)))$  is also an order preserving self-map. If  $\mathcal{G}(u)$  commutes with the addition of constant scalars, i.e.,  $\mathcal{G}(u + \omega) = \mathcal{G}(u) + \omega$  then

$$\mathcal{G}(\mathcal{G}(u + \omega)) = \mathcal{G}(\mathcal{G}(u) + \omega) = \mathcal{G}(\mathcal{G}(u)) + \omega.$$

From results found in Crandall and Tartar [12], it follows that  $\mathcal{G}(u)$ ,  $\mathcal{G}(\mathcal{G}(u))$ , and  $\frac{1}{2}(u + \mathcal{G}(\mathcal{G}(u)))$  are all nonexpansive. More generally, consider the weaker concept of positive self-maps

$$\mathcal{G}(u) = T(u)u, \quad T(u) : \mathbf{R}^{|V|} \rightarrow \mathbf{R}^{|V| \times |V|}, T(u) \geq 0, \quad (74)$$

where  $T(u) \geq 0$  denotes that elements of  $T(u)$  are nonnegative. Considering the composition term

$$\mathcal{G}(\mathcal{G}(u)) = T(T(u)u)T(u)u, \quad (75)$$

it is clear that  $T(T(u)u)T(u) \geq 0$  and using previous arguments  $\mathcal{G}(\mathcal{G}(u + \omega)) = \mathcal{G}(\mathcal{G}(u)) + \omega$ . From this the stated lemma is concluded since  $\frac{1}{2}(u^n + \mathcal{G}(\mathcal{G}(u^n)))$  is also a positive self-map which commutes with the addition of scalars. ■

## 4.2. Space-Time Petrov–Galerkin Approximation

The Petrov–Galerkin formulation readily permits tensor product space-time discretization. Let  $N_i(x)$  denote the simplex basis functions as described in Subsection 3.1. Consider the following space-time approximations:

*Piecewise constant in time*  $x \in T$ ,  $t \in [t_+^n, t_-^{n+1}]$ . In this case, the time derivatives  $u_t = w_t = 0$  vanish in slab interiors and evolution takes place via the space-time slab jump integral

$$\int_{\Omega} w(t_+^n)(u(t_+^n) - u(t_-^n)) d\Omega \quad (76)$$

with  $u(t_+^n) = u^{n+1}$  and  $u(t_-^n) = u^n$ . This produces the scheme

$$\begin{aligned} \int_{\Omega} w^{n+1} \frac{u^{n+1} - u^n}{\Delta t} d\Omega + \int_{\Omega} (w^{n+1} + p^{-1}(\nabla H \cdot \nabla w^{n+1})) \tau H(\nabla u^{n+1}) d\Omega \\ + \int_{\Gamma_-} w^{n+1} (g - u^{n+1}) p^{-1}(\nabla H \cdot n) d\Gamma = 0. \end{aligned}$$

To regain consistency, as defined in Section 2, the jump integral weighting term is perturbed by  $\tau p^{-1}(\nabla H \cdot \nabla w)$  thereby producing the following consistent approximation

$$\begin{aligned} \int_{\Omega} (w^{n+1} + p^{-1}(\nabla H \cdot \nabla w^{n+1})) \tau \left( \frac{u^{n+1} - u^n}{\Delta t} + H(\nabla u^{n+1}) \right) d\Omega \\ + \int_{\Gamma_-} w^{n+1} (g - u^{n+1}) p^{-1}(\nabla H \cdot n) d\Gamma = 0. \end{aligned}$$

Note that this scheme still requires the implicit inversion of a matrix despite our expectation that the order of accuracy in time is  $O(\Delta t)$ . This form motivates a simple explicit scheme achieved by shifting  $w^{n+1}$  and  $u^{n+1}$  back to  $w^n$  and  $u^n$  everywhere except for the  $u^{n+1}$  term appearing in the time derivative approximation.

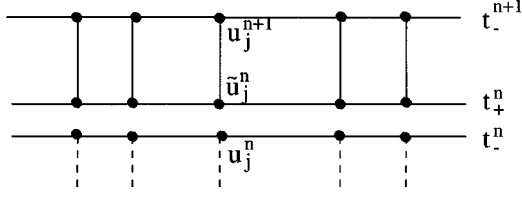
$$\begin{aligned} \int_{\Omega} (w^n + p^{-1}(\nabla H \cdot \nabla w^n)) \tau \left( \frac{u^{n+1} - u^n}{\Delta t} + H(\nabla u^n) \right) d\Omega \\ + \int_{\Gamma_-} w^n (g - u^n) p^{-1}(\nabla H \cdot n) d\Gamma = 0. \end{aligned}$$

When combined with standard mass lumping, an explicit time stepping scheme similar to the positive coefficient scheme is obtained. Full algorithmic details are given later.

*Piecewise linear in time*  $x \in T$ ,  $t \in [t_+^n, t_-^{n+1}]$ .

$$\begin{aligned} u(x, t) &= \sum_{j=1}^3 N_j(x) \left( \frac{t - t_+^n}{\Delta t} u_j^{n+1} + \left( 1 - \frac{t - t_+^n}{\Delta t} \right) \tilde{u}_j^n \right) \\ w(x, t) &= \sum_{j=1}^3 N_j(x) \left( \frac{t - t_+^n}{\Delta t} w_j^{n+1} + \left( 1 - \frac{t - t_+^n}{\Delta t} \right) \tilde{w}_j^n \right) \end{aligned} \quad (77)$$

as depicted in Fig. 10. The use of time-discontinuous approximations effectively decouples successive space-time slabs. Even so, for a given space-time slab, both  $\tilde{u}_j^n$  and  $u_j^{n+1}$  are basic coupled unknowns. In addition, the required integrals are nonlinear functions of these unknowns. This motivates the use of a locally linearized form of the Petrov–Galerkin



**FIG. 10.** One-dimensional space-time slab depicting multiple data values from  $t_{-}^n$  to  $t_{+}^n$ .

scheme. Using midpoint quadrature formula to evaluate Petrov–Galerkin integrals, let  $\bar{u}$  denote a reference state at the quadrature point location. Define  $\nabla \bar{H} = \nabla H(\bar{u})$  and  $\bar{\tau} = \tau(\bar{u})$  and the following variational forms:

$$\begin{aligned}
 B(u, w; \bar{u})_{gal} &= \int_{I^n} \int_{\Omega} w (u_t + p^{-1} \nabla \bar{H} \cdot \nabla u) d\Omega dt \\
 B(u, w; \bar{u})_{ls} &= \int_{I^n} \int_{\Omega} (w_t + p^{-1} (\nabla \bar{H} \cdot \nabla w)) \bar{\tau} (u_t + p^{-1} (\nabla \bar{H} \cdot \nabla u)) d\Omega dt \quad (78)
 \end{aligned}$$

so that the desired solution is obtained as  $\bar{u} \rightarrow u$ . The following iterative solution procedure, consisting of  $m$  inner iterations, is used for updating each space-time slab

1. Initialize  $\bar{u} = u^n$ .

2. For  $s = 1, m$

Given the reference state  $\bar{u}$ , solve the linearized Petrov–Galerkin discretization thereby obtaining  $u^{n+s/m}$  and  $\tilde{u}^{n+s/m-1}$

Update the linearize state  $\bar{u} = \bar{u}(u^{n+s/m}, \tilde{u}^{n+s/m-1})$

EndFor

In practice, 2 or 3 inner iterations are usually required for convergence. While we do not recommend this as a final strategy for solving the Petrov–Galerkin system, it is sufficient for evaluating the merits of the discretization.

## 5. IMPLEMENTATION OF EXPLICIT POSITIVE AND PETROV–GALERKIN SCHEMES FOR THE LEVEL SET EQUATION

### 5.1. The Explicit Positive Coefficient Scheme for the Level Set Equation

Consider the level set equation

$$\phi_t + F(\nabla \phi, x) |\nabla \phi| = f(x), \quad x \in \mathbf{R}^d \quad (79)$$

discretized using the positive coefficient scheme described in Subsection 3.2. This algorithm can be implemented using the following algorithmic steps:

(1) Initialize  $\phi_i^* = w_i = 0$ ,  $i = 1, 2, \dots, |V|$ .

(2) Foreach  $T \in \mathcal{T}$ ,  $i = 1, 2, \dots, d + 1$

$$T \xrightarrow{\text{local}} \text{simplex}(x_1, x_2, \dots, x_{d+1})$$

$$N_i(x) = \{N_i(x) \in \mathcal{P}_1 \mid N_i(x_j) = \delta_{ij}, j = 1, 2, \dots, d + 1, x \in T\}$$

$$\vec{n}_i = d \text{ meas}(T) \nabla N_i$$

$$\begin{aligned}
\nabla\phi &= \sum_{j=1}^{d+1} \nabla N_j \phi_j \\
\bar{F} &= \frac{1}{\text{meas}(T)} \int_T F(\nabla\phi) dx \\
\bar{f} &= \frac{1}{\text{meas}(T)} \int_T f(x) dx \\
K_i &= \frac{\bar{F} \nabla\phi \cdot \vec{n}_i}{d|\nabla\phi|} \\
\delta\phi &= \sum_{l=1}^{d+1} K_l \phi_l \\
\delta\phi_i &= K_i^+ \left( \sum_{l=1}^{d+1} K_l^- \right)^{-1} \sum_{l=1}^{d+1} K_l^- (\phi_i - \phi_l) \\
\tilde{\alpha}_i &= \frac{\max(0, \delta\phi_i / \delta\phi)}{\sum_{l=1}^{d+1} \max(0, \delta\phi_l / \delta\phi)} \\
\phi_i^* &= \phi^* + \tilde{\alpha}_i (\delta\phi - \bar{f} \text{meas}(T)) \\
w_i &= w_i + \tilde{\alpha}_i \text{meas}(T)
\end{aligned}$$

Endfor

### Single-Stage Integration

(3) Foreach  $v_i \in V$

$$\phi_i^{n+1} = \phi_i^n - \Delta t \frac{(\phi_i^*)^n}{w_i^n}$$

Endfor

### Two-Stage Integration

(3) Foreach  $v_i \in V$

$$\begin{aligned}
\phi_i^{n+1/2} &= \phi_i^n - \Delta t \frac{(\phi_i^*)^n}{w_i^n} \\
\phi_i^{n+1} &= \frac{1}{2} (\phi_i^n + \phi_i^{n+1/2}) - \frac{\Delta t}{2} \frac{(\phi_i^*)^{n+1/2}}{w_i^{n+1/2}}
\end{aligned}$$

Endfor

## 5.2. The Explicit Petrov–Galerkin Scheme for the Level Set Equation

Once again consider the level set equation

$$\phi_t + F(\nabla\phi, x)|\nabla\phi| = f(x), \quad x \in \mathbf{R}^d. \quad (80)$$

The Petrov–Galerkin implementation follows closely the implementation for the positive coefficient scheme:

(1) Initialize  $\phi_i^* = w_i = 0$ ,  $i = 1, 2, \dots, |V|$ .

(2) Foreach  $T \in \mathcal{T}$ ,  $i = 1, 2, \dots, d+1$

$$T \xrightarrow{\text{local}} \text{simplex}(x_1, x_2, \dots, x_{d+1})$$

$$N_i(x) = \{N_i(x) \in \mathcal{P}_1 \mid N_i(x_j) = \delta_{ij}, \quad j = 1, 2, \dots, d+1, \quad x \in T\}$$

$$\vec{n}_i = d \operatorname{meas}(T) \nabla N_i$$

$$\nabla \phi = \sum_{j=1}^{d+1} \nabla N_j \phi_j$$

$$\bar{F} = \frac{1}{\operatorname{meas}(T)} \int_T F(\nabla \phi) dx$$

$$\bar{f} = \frac{1}{\operatorname{meas}(T)} \int_T f(x) dx$$

$$K_i = \frac{\bar{F} \nabla \phi \cdot \vec{n}_i}{d |\nabla \phi|}$$

$$\tau = \tau(\phi)$$

$$\delta \phi = \sum_{l=1}^{d+1} K_l \phi_l$$

$$\alpha_i = \frac{1}{d} + \tau K_i$$

$$\phi_i^* = \phi^* + \alpha_i (\delta \phi - \bar{f} \operatorname{meas}(T))$$

$$w_i = w_i + \alpha_i \operatorname{meas}(T)$$

Endfor

### Single-Stage Integration

(3) Foreach  $v_i \in V$

$$\phi_i^{n+1} = \phi_i^n - \Delta t \frac{(\phi_i^*)^n}{w_i^n}$$

Endfor

### Two-Stage Integration

(3) Foreach  $v_i \in V$

$$\phi_i^{n+1/2} = \phi_i^n - \Delta t \frac{(\phi_i^*)^n}{w_i^n}$$

$$\phi_i^{n+1} = \frac{1}{2} (\phi_i^n + \phi_i^{n+1/2}) - \frac{\Delta t}{2} \frac{(\phi_i^*)^{n+1/2}}{w_i^{n+1/2}}$$

Endfor

## 5.3. Numerical Accuracy

To assess the accuracy of the various schemes described in previous sections, the level set and Eikonal equations are discretized and solved for the problem of computing the distance from a given curve.

- In the case of the level set equation, we start with a given curve as the zero level set of the initial value  $u$  in a region  $\Omega$  and evolve the time dependent equation  $u_t + |\nabla u| = 0$ .
- In the case of the Eikonal equation, we start with a given curve as a boundary condition to the equation  $|\nabla u| = 1$ .

Two test problems are considered. The first computes the distance to a convex curve, which then must be a smooth solution. The second computes the distance to a nonconvex

curve, which gives rise to non-smooth solutions. Numerical calculations were carried out on a sequence of 4 meshes with characteristic element size  $h$  successively decreased by factors of 2. In both problems, an initial curve  $(x_0(\theta), y_0(\theta))$  is prescribed in terms of polar angle  $\theta$  such that the initial solution satisfies  $u(x_0, y_0, t) = 0$ . Next, the solution is extended everywhere via a signed distance-squared function; this is done so that the final solution is not trivially given by the initial data. The discretized equations are then advanced in time using the 2-stage scheme with  $\Delta t \approx h/2$  even though second order accuracy in time is not generally expected. During the course of the calculation, a *crossing time* solution  $u_c(x, y)$  is constructed which records the time at which the  $u(x, y, t) = 0$  level set passes over a given mesh vertex. In practice a parabolic curve fitting technique (in time) is used to obtain precise crossing times given the solution at three successive time steps. For the test problems mentioned below the crossing time solution is unique and valid only for the portion of  $\Omega$  exterior to the curve  $(x_0(\theta), y_0(\theta))$ . To assess accuracy, the absolute error in the crossing time solution is then measured in the Sobolev and  $L_p$  norms  $H^1$ ,  $L_2$ , and  $L_\infty$ .

### 5.3.1. Problem 1A. Smooth Solutions to the H-J and Level Set Equations

As a first example, the level set equation is solved starting from smooth initial solution data corresponding to an oval. Exterior to the oval, the solution remains smooth for the entire calculation. Figure 11 shows the mesh and crossing time solution computed using the positive coefficient scheme. Note that no crossing time solution exists interior to the specified oval curve. Graphs of the absolute crossing time error using the explicit positive coefficient, explicit Petrov–Galerkin, and implicit Petrov–Galerkin schemes are shown in Fig. 12. The two explicit schemes yield slightly less than first order accuracy. In contrast, the implicit Petrov–Galerkin scheme yields full second order accuracy in  $L_2$  and  $L_\infty$  norms and first order accuracy in the  $H^1$  semi-norm. These are optimal convergence rates in these norms for linear elements.

### 5.3.2. Problem 1B. Smooth Solutions to the Eikonal Equation

In this test problem, the crossing time solution from Problem 1A is computed as a field solution of the Eikonal equation. The mesh and field solution using the positive coefficient scheme are shown in Fig. 13. The specified oval curve is now a *boundary* for the domain

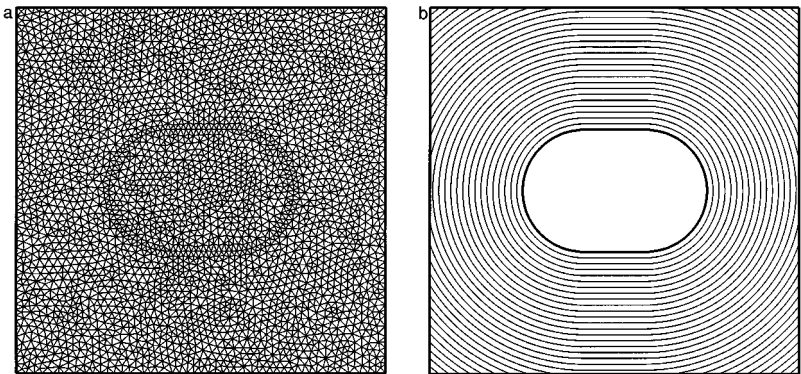
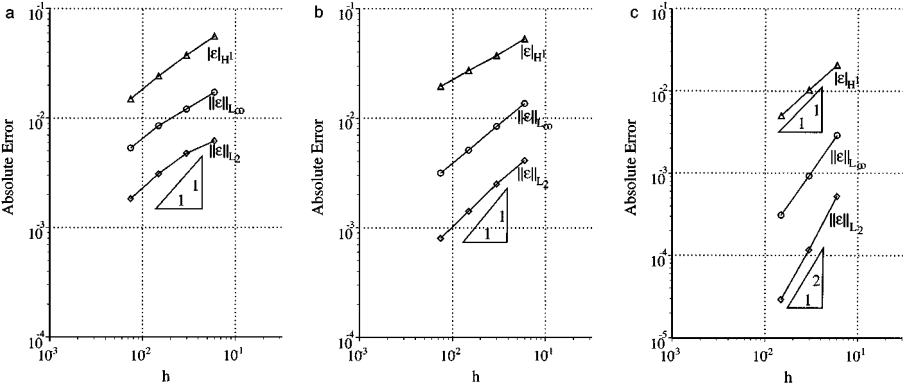


FIG. 11. Smooth level set solutions: (a) coarsest mesh and (b) crossing time solution.



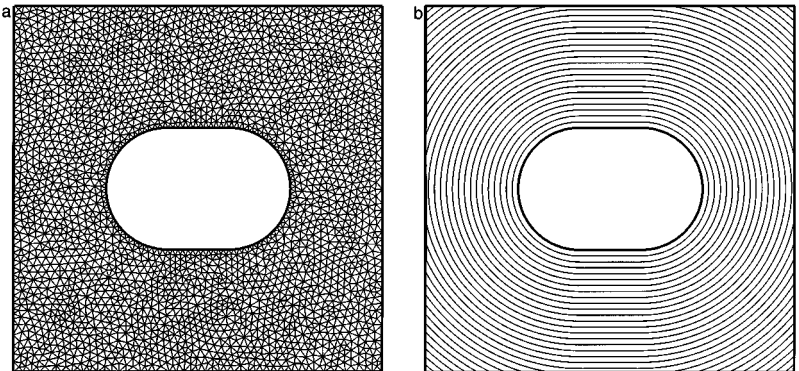


**FIG. 12.** Smooth level set solutions: absolute crossing time solution error. (a) Positive coefficient scheme, (b) explicit Petrov–Galerkin, and (c) implicit Petrov–Galerkin.

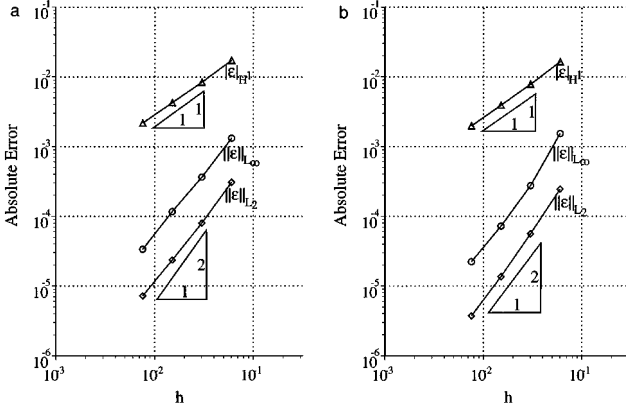
with imposed boundary condition  $u = 0$ . The Eikonal equation does not depend on time so that the explicit and implicit Petrov–Galerkin schemes produce identical discretizations. Consequently, absolute solution errors are graphed in Fig. 14 for the positive coefficient scheme and the Petrov–Galerkin scheme. Both schemes perform optimally: second order accurate in  $L_2$  and  $L_\infty$  and first order accurate in an  $H^1$  semi-norm.

### 5.3.3. Problem 2A. Non-smooth Solutions to Level Set Equation

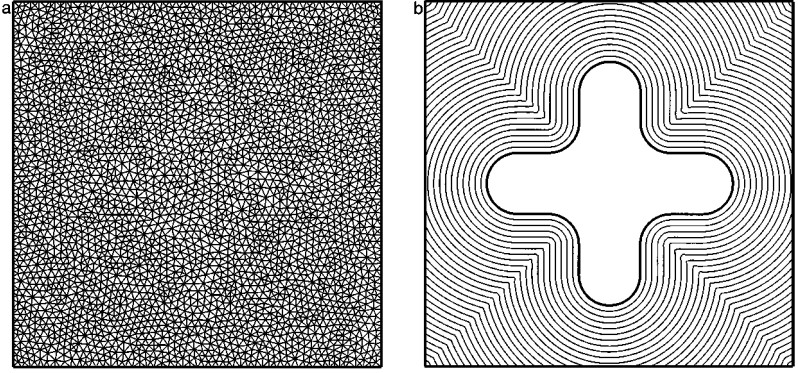
As a second example, the level set equation is solved starting from smooth initial solution data corresponding to an star-shaped curve which remains only Lipschitz smooth during time evolution. Figure 15 shows the mesh and crossing time solution computed using the positive coefficient scheme. Note the non-oscillatory resolution of the slope-discontinuous corner. Graphs for absolute error in the crossing time solution are shown in Fig. 16. The two explicit schemes again show slightly less than first order accuracy in  $L_2$  and  $L_\infty$  with further degradation in the  $H^1$  semi-norm. The implicit Petrov–Galerkin with discontinuity capturing term parameter  $\sigma_{dc} = 0.1$  retains second order accuracy in  $L_2$ , first order accuracy in  $L_1$ , and one-half order accuracy in the  $H^1$  semi-norm.



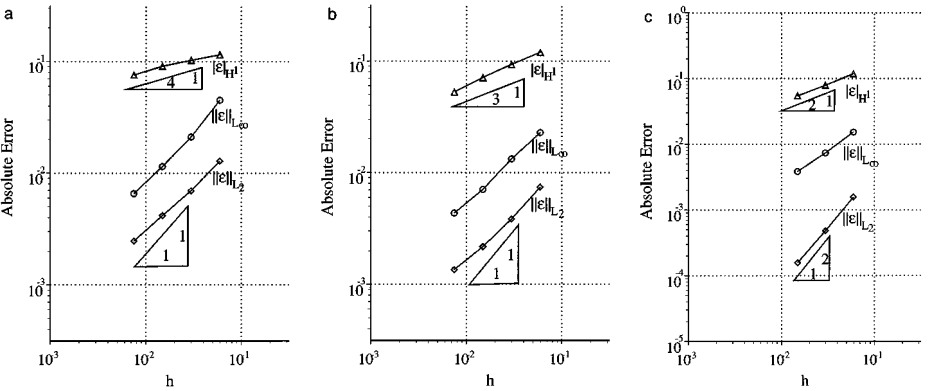
**FIG. 13.** Smooth Eikonal solutions: coarsest mesh (left) and solution (right).



**FIG. 14.** Smooth Eikonal solutions: absolute solution error. (a) Positive coefficient scheme, (b) explicit Petrov–Galerkin.



**FIG. 15.** Non-smooth level set solutions: (a) coarsest mesh and (b) crossing time solution.



**FIG. 16.** Non-smooth level set solutions: absolute crossing time solution error. (a) Positive coefficient scheme, (b) explicit Petrov–Galerkin, and (c) implicit Petrov–Galerkin.

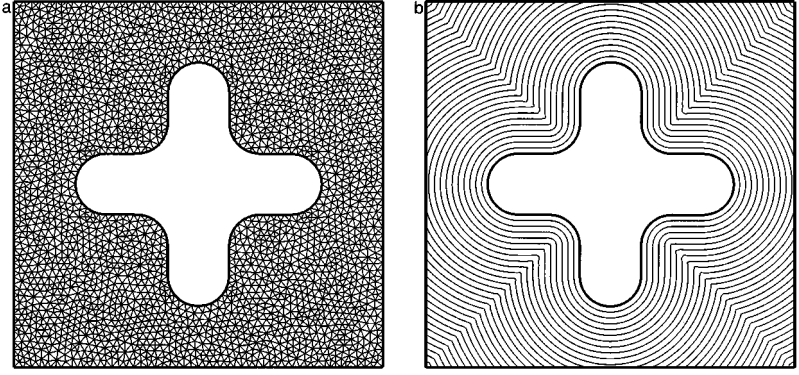


FIG. 17. Non-smooth Eikonal solutions: (a) coarsest mesh and (b) solution.

### 5.3.4. Problem 2B. Non-smooth Solutions to Eikonal Equation

As a last test problem, the crossing time solution from Problem 2A is computed as a field solution of the Eikonal equation. The specified star-shaped curve is a boundary for the domain with imposed boundary condition  $\phi = 0$ . The mesh and field solution using the positive coefficient scheme are shown in Fig. 17. Absolute solution errors are graphed in Fig. 18 for the positive coefficient scheme and the Petrov–Galerkin scheme. Again, both the positive coefficient and Petrov–Galerkin schemes with discontinuity capturing term perform very similarly. Second order accuracy is obtained in  $L_2$ , first order accuracy in  $L_\infty$ , and one-half order accuracy in the  $H^1$  semi-norm.

## 6. SCHEMES FOR CURVATURE FLOW

Many physically relevant problems are accurately modeled by the level set equation with added second order curvature flow term  $\kappa |\nabla \phi|$  of the form

$$\phi_t + (F(\nabla \phi) - \epsilon \kappa) |\nabla \phi| = 0 \quad (81)$$

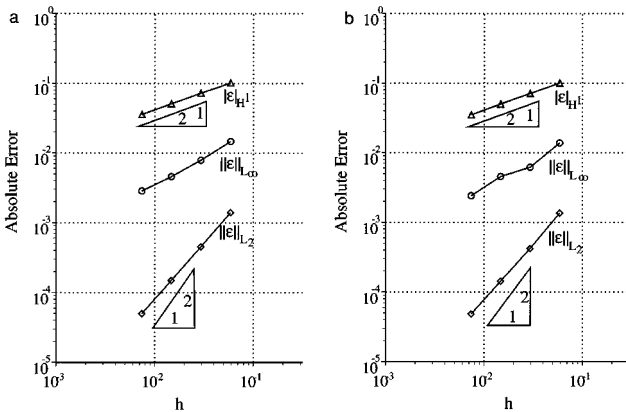


FIG. 18. Non-smooth Eikonal solutions: absolute solution error. (a) Positive coefficient scheme, (b) Petrov–Galerkin.

for locally constant  $\epsilon$ . Recall that  $\kappa$  is the mean curvature of the level set function  $\phi$

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}. \quad (82)$$

In discretizing the curvature term, we follow the same reasoning given by Osher and Sethian [22], namely that numerical stability will be dictated by the highest order differential term. This motivates a study of the following model equation on  $\Omega \subset \mathbf{R}^2$  with boundary  $\Gamma$ ,

$$\begin{aligned} \phi_t - \epsilon \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} &= 0, & x \in \Omega \\ \phi(x, 0) &= \phi_0(x), & x \in \Omega \\ \phi(x, t) &= g(x), & x \in \Gamma, \end{aligned} \quad (83)$$

which can be restated in variational form. Let  $\mathcal{S}^h$  denote the space of finite-dimensional functions with bounded total variation energy,  $\int_{\Omega} |\nabla \phi| dx$ , satisfying the Dirichlet boundary condition and  $\mathcal{V}^h$  the same space of finite-dimensional functions which satisfy homogeneous Dirichlet boundary conditions. Given these spaces, find  $\phi \in \mathcal{S}^h$ , such that for all  $w \in \mathcal{V}^h$

$$\int_{\Omega} w \phi_t dx + \epsilon \int_{\Omega} \frac{\nabla w \cdot \nabla \phi}{|\nabla \phi|} dx = 0$$

with

$$\begin{aligned} \phi(x, 0) &= \phi_0(x), & x \in \Omega \\ \phi(x, t) &= g(x), & x \in \Gamma. \end{aligned} \quad (84)$$

Examining Eq. (84) with linear elements yields the following simple lemma.

**LEMMA.** *The model equation (84) with  $C^0$  linear triangular elements exhibits a discrete maximum principle at steady-state for arbitrary solution data if the triangulation is non-obtuse.*

*Proof.* Consider a single arbitrary simplex  $T = \text{simplex}(x_1, x_2, x_3)$  and the discretization in terms of the local linear shape functions  $N_j(x)$ ,  $\phi(x) = \sum_{j=1}^3 N_j(x) \phi_j$ ,  $x \in T$

$$\int_T \frac{\nabla w \cdot \nabla \phi}{|\nabla \phi|} dx = \sum_{i=1}^3 \sum_{j=1}^3 w_i \phi_j \frac{\nabla N_i \cdot \nabla N_j}{|\nabla \phi|} \text{meas}(T). \quad (85)$$

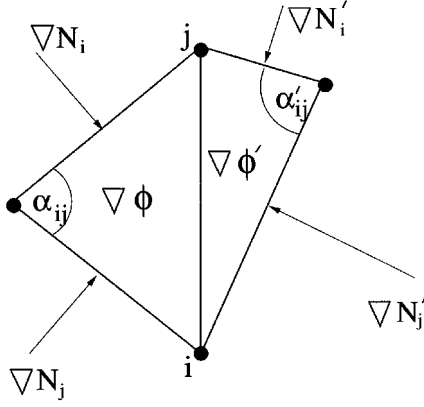
After some straightforward manipulation, the following global discretization with time term set to zero is obtained

$$\int_{\Omega} \frac{\nabla w \cdot \nabla \phi}{|\nabla \phi|} dx = \sum_{i=1}^{|V|} w_i \sum_{j \in \mathcal{N}_i} W_j^i (\phi_i - \phi_j) = 0, \quad (86)$$

where  $\mathcal{N}_i$  denotes the set of vertices adjacent to vertex  $v_i$  (see Fig. 19) with weights

$$\begin{aligned} W_j^i &= \left[ \frac{\nabla N_i \cdot \nabla N_j}{|\nabla \phi|} \text{meas}(T) + \frac{\nabla N_i' \cdot \nabla N_j'}{|\nabla \phi|'} \text{meas}(T') \right] \\ &= \frac{1}{2} \left[ \frac{\cotan(\alpha_{ij})}{|\nabla \phi|} + \frac{\cotan(\alpha'_{ij})}{|\nabla \phi|'} \right]. \end{aligned} \quad (87)$$

Since the discretization formula must hold for arbitrary values of  $w_i$  at interior vertices, it



**FIG. 19.** Discretization weight geometry for the edge  $e(v_i, v_j)$ .

can be concluded that for all interior vertices  $v_i$

$$\sum_{j \in \mathcal{N}_i} W_j^i (\phi_i - \phi_j) = 0. \quad (88)$$

Written in this form, a discrete maximum principle exists if all weights are nonnegative,  $W_j^i \geq 0$ , since  $\phi_i$  is then a convex combination of adjacent neighbors, i.e.,

$$\phi_i = \frac{\sum_{j \in \mathcal{N}_i} W_j^i \phi_j}{\sum_{j \in \mathcal{N}_i} W_j^i} \quad (89)$$

so that

$$\min_{j \in \mathcal{N}_i} \phi_j \leq \phi_i \leq \max_{j \in \mathcal{N}_i} \phi_j. \quad (90)$$

From Eq. (87) it follows directly that a sufficient condition for  $W_j^i \geq 0$  is that the triangulation be non-obtuse. ■

Our strategy for the general level set curvature flow term

$$|\nabla \phi| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \quad (91)$$

is to obtain a pointwise estimate at each vertex  $v_i$  of the form

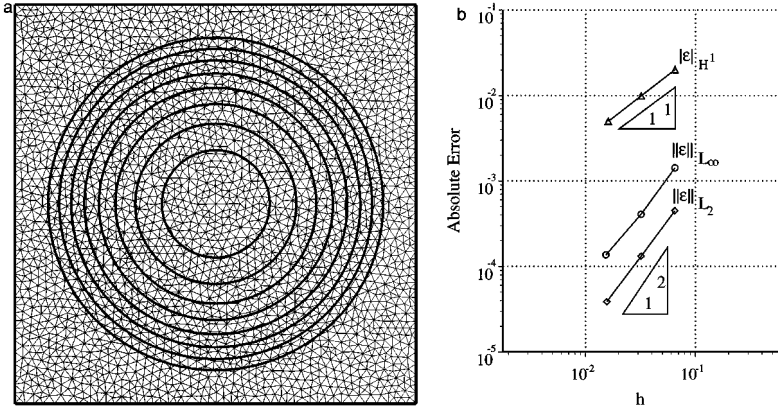
$$|\nabla \phi| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \Big|_i \approx |\nabla \phi|_i \frac{\sum_{j \in \mathcal{N}_i} W_j^i (\phi_j - \phi_i)}{\sum_{T \in \mathcal{N}_i} \text{meas}(T)}, \quad (92)$$

where the  $W_j^i$  are the weights described earlier,  $\mathcal{N}_i$  denotes the triangle neighbor set incident to  $v_i$ , and  $(\nabla \phi)_i$  is the pointwise (mass-lumped) Galerkin approximation, i.e.,

$$(\nabla \phi)_i = \frac{\sum_{T \in \mathcal{N}_i} \int_T \nabla \phi \, dx}{\sum_{T \in \mathcal{N}_i} \text{meas}(T)}. \quad (93)$$

This term then incorporates easily into the previously discussed explicit schemes. We have not yet considered the addition of this term into the implicit Petrov–Galerkin scheme.

To assess the accuracy of the curvature approximation, two numerical experiments have been performed. In the first experiment, curvature flow collapse of a 2D circular level set is numerically simulated on a sequence of meshes of increasing mesh density using the



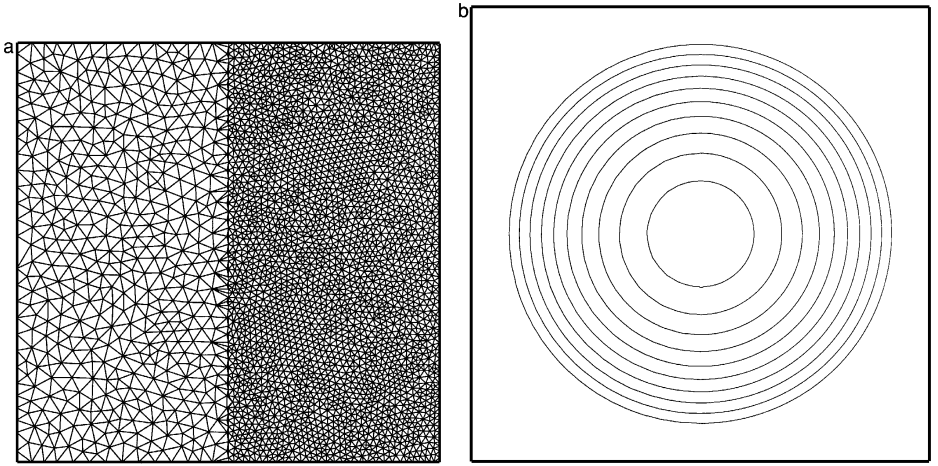
**FIG. 20.** Curvature flow collapse of a unit circle. (a) Coarsest mesh and crossing time solution sampled at equal time intervals and (b) absolute solution error using two-stage Runge–Kutta time integration.

two-stage Runge–Kutta time advancement with  $\Delta t \approx h^2/8$ . The crossing time solution for radii values  $[1/10, 1]$  is computed and compared with the known exact solution. Figure 20(a) shows the coarsest mesh with superimposed crossing time solution contours corresponding to equal time increments. Figure 20(b) graphs the  $H^1$ ,  $L_2$ , and  $L_\infty$  norms of the crossing time solution error versus the mesh spacing parameter  $h$  for each mesh. Despite the mass-lumped Galerkin approximation used in the curvature flow calculation, near optimal order accuracy is observed, i.e., second order accuracy in  $L_2$  and  $L_\infty$  norms and first order accuracy in the  $H^1$  semi-norm. This unexpected result is attributable to the nearly uniform distribution of simplices. More generally, we expect the results to deteriorate unless an implicit mass-like matrix is included in the calculation.

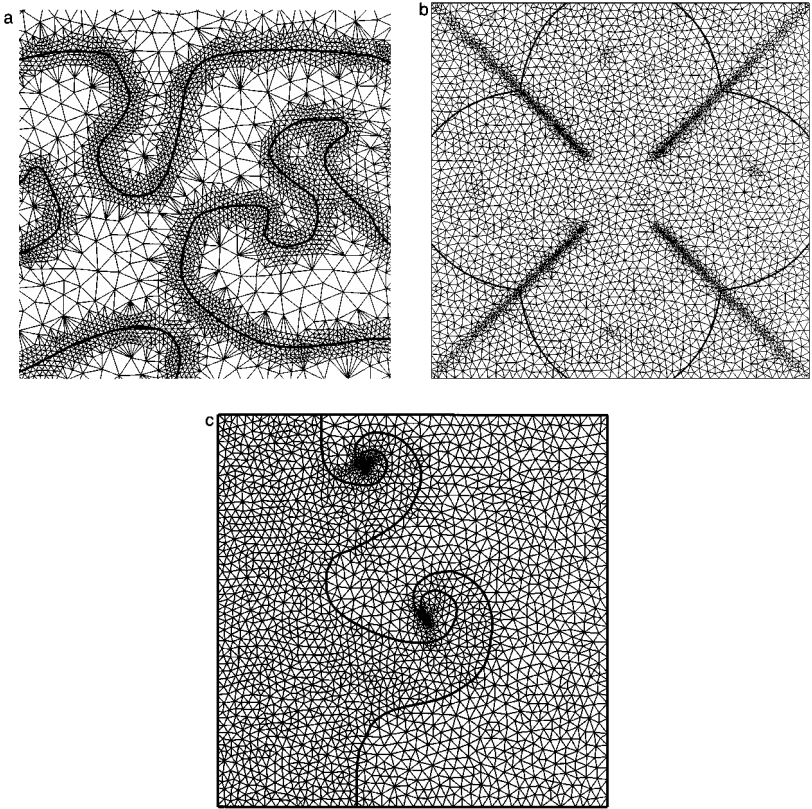
The second numerical experiment computes the same curvature flow problem on a mesh that contains 500 vertices in the left half of the mesh and 2800 vertices in the right half of the mesh. The mesh density jumps abruptly at the  $x = 0$  axis. This test problem was devised by Milne [21] to assess the robustness and accuracy of curvature flow calculations on adaptive Cartesian meshes when interpolating the solution from a coarse mesh boundary onto a fine mesh boundary. Specifically, Milne observed a noticeable anomaly in the solution at the location of the interpolation. Figure 21 shows the results of a similar calculation carried out on the triangulated domain. The two-stage Runge–Kutta scheme with  $\Delta t = 0.0004$  was used for time advancement. Crossing time solution contours were computed corresponding to equal time intervals. The crossing time solution contours shown in this figure do not reveal any noticeable defects in solution contours along the  $x = 0$  axis.

## 7. MESH ADAPTIVITY

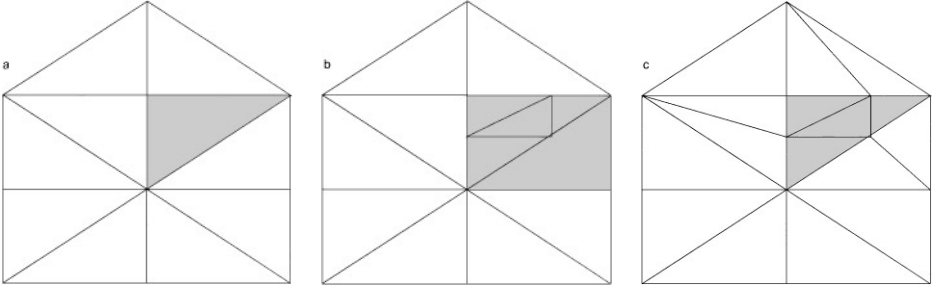
One of the central virtues of a triangulated domain formulation, as opposed to a Cartesian mesh framework, is the ease in implementing mesh adaptivity. The triangulated mesh approach avoids the significant problems that exist in Cartesian mesh schemes when interpolating between coarse and fine meshes to resolve nonconforming (hanging) nodes in curvature approximation. Mesh adaptation can either be part of the initial mesh construction, or can occur dynamically as the solution evolves. Two basic adaptation strategies have been considered: (1) conformal adaptation and (2) Steiner/Delaunay adaptation. The next two sections will elaborate on these techniques. As examples of these techniques taken from level set computations, Fig. 22 shows three different uses of mesh adaptation for evolving front



**FIG. 21.** Curvature flow collapse of a unit circle. (a) Triangulated domain and (b) crossing time solution contours at equal time intervals.



**FIG. 22.** Various mesh adaptation techniques for evolving interface problems. (a) Interface proximity adaptation; (b) curvature adaptation; (c) interface conforming adaptation.



**FIG. 23.** Conformal refinement steps for a 2D triangle. (a) candidate triangle, (b) conformal subdivision, (c) resolution of subdivision.

problems. Figure 22a shows conformal adaptation around the front itself; Fig. 22b shows Steiner adaptation around regions of high curvature in the evolving front; Fig. 22c shows interface conforming Steiner adaptation around regions of high curvature in the evolving solution.

### 7.1. Conformal Adaptation

The term conformal adaptation comes from the fact that a two-dimensional triangle can be divided into four sub-triangles, each having the same angles as the original triangle. Using this idea, a natural two-step algorithm exists to sub-divide any triangle. In step one, the selected triangle (shown in light gray) is conformally sub-divided by connecting the midpoints of each side (see Fig. 23). In step two, these side “hanging nodes” are then resolved by connecting them to the far vertex. The result is again a triangulated grid with no hanging nodes. Algorithmically, one can proceed as follows. The initial grid is made up of “parent” and “child” triangles; we insist that at no point will parent triangles ever be removed. This grid is stored as a list of triangles; each triangle contains a pointer to its parent, as well as to its children (which are the first level subdivisions within that parent). Addition of triangles corresponds to adding children to parents in the list. Removal of triangles can easily occur by removing all children (and grandchildren, etc.) from the list. This can easily be performed dynamically as the solution evolves.

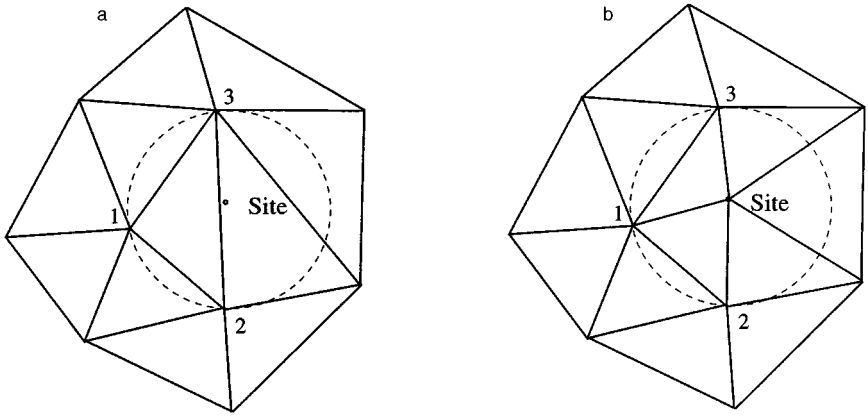
The technique extends to three dimensions, see, for example, [20, 31], although the method ceases to be angle preserving. In three dimensions, a single tetrahedron is decomposed into eight sub-tetrahedron. The lack of angle preservation introduces a *stability* question associated with repeated application of refinement refinement, i.e., that angles do not degenerate too quickly. It is known that stability is obtained if the shortest of two possible interior edges is used in the 1:8 refinement. The handling of hanging nodes requires 1 : 2 and 1 : 4 refinement as well.

### 7.2. Steiner/Delaunay Adaptation

**DEFINITION.** A Steiner triangulation algorithm is any triangulation algorithm that adds additional sites to an existing triangulation to improve some measure of grid quality, Bern and Epstein [6].

The Steiner/Delaunay adaptation technique exploits the well-known circumcircle/circumsphere characterization of unconstrained Delaunay triangulations, i.e., that the circumcircle/circumsphere of any triangle/tetrahedron does not contain any other vertex in the





**FIG. 24.** Steiner refinement: (a) Site insertion at circumcenter location and (b) edge reconfiguration.

triangulation. A number of researchers have independently discovered the benefits of inserting sites at circumcenters of Delaunay triangles to refine the triangulation and improve measures of grid quality [7, 25]. For example, in Fig. 24, the triangle  $T(v_1, v_2, v_3)$  is refined by adding a site at the circumcenter location and reconfiguring edges using an edge flipping procedure to restore the Delaunay characterization. Most algorithms follow a procedure similar to that proposed by Chew:

**ALGORITHM.** Steiner Triangulation and Adaptation, Chew [7].

1. Construct a constrained Delaunay triangulation of the boundary points and edges.
2. Compute a measure of shape and size for each element. A triangle is excluded from refinement consideration if: (1) it is well-shaped, i.e., the smallest angle is greater than  $30^\circ$ , (2) it is well-sized, i.e., the triangle passes a user defined sizing measure. Any sizing measure can be specified as long as it can be achieved by making the triangle smaller. Some sizing measures might include solution adaptive gradients (properly scaled), solution curvature, cell area, etc.
3. If all triangles are well-sized and well-shaped then halt. Otherwise choose the largest triangle,  $\Delta$ , which fails and determine its circumcenter,  $c$ .
4. Traverse from  $\Delta$  toward  $c$  until either a constraining boundary edge is encountered or the triangle containing  $c$  is found.
5. If a triangle is found containing  $c$  then insert  $c$  into the triangulation and proceed to Step 2.
6. If a boundary edge is encountered during the traversal then split the boundary edge into halves and update the triangulation. Let  $l$  be the length of the new edges and consider the new vertex located on the boundary. Delete each interior vertex of the triangulation which is closer than  $l$  to this boundary site. Proceed to Step 2.

Using this algorithm it is proven in [7, 8] that guaranteed-quality meshes are obtained:

1. All angles in the triangulation lie between  $30^\circ$  and  $120^\circ$ . (Smaller angles are required if boundary angles less than  $30^\circ$  are allowed.)
2. All triangles will pass the user specified measure.
3. All boundary edge constraints will be preserved by the final triangulation.

Within the context of the level set method, the interface can be treated as a constraining boundary and Chew's algorithm applied in a two-sided sense since additional vertices may

be placed on the interface which impacts the triangulation on both sides of the interface. Note that circumcenter insertion extends naturally to Delaunay triangulations in three space dimension although the technique does not yield guaranteed angle bounds of any known type.

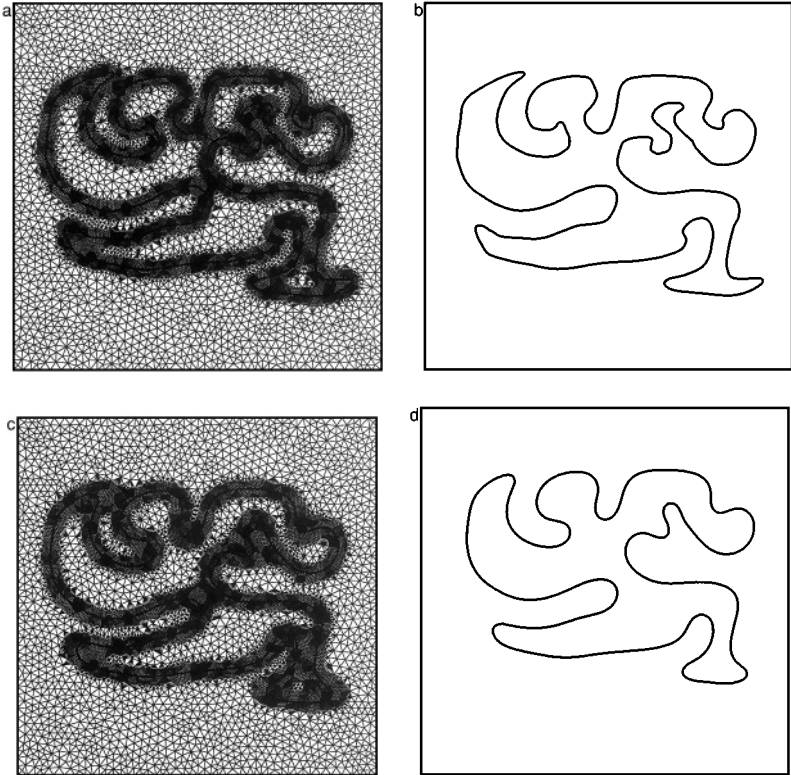
## 8. EXAMPLE CURVATURE FLOW CALCULATIONS

### 8.1. Grayson's Problem

To demonstrate the performance of curvature flow discretization as well as the use dynamic mesh adaptation, the motion of a simple closed curve collapsing with speed proportional to local curvature is numerically computed. In considering this problem, Grayson [14] proved the remarkable theorem that *all* simple closed curves eventually collapse to a round point. It serves as a challenging test problem to numerically verify Grayson's theorem. In the context of the level set equation, simulating Grayson's problem amounts to solving the following level set equation:

$$\phi_t - \kappa |\nabla \phi| = 0. \quad (94)$$

Figures 25a–25f show a numerical calculation of 2-D curvature flow using the curvature discretization of Section 6. In this calculation, the two-stage Runge–Kutta time advancement



**FIG. 25.** Grayson's Problem. Time evolution of simple closed curve with curvature dependent speed. (a) Initial adopted mesh; (b) zero level set ( $T = 0$ ); (c) adapted mesh; (d) zero level set ( $T = 0.1$ ); (e) adapted mesh; (f) zero level set ( $T = 0.5$ ); (g) adapted mesh; (h) zero level set ( $T = 1.5$ ).

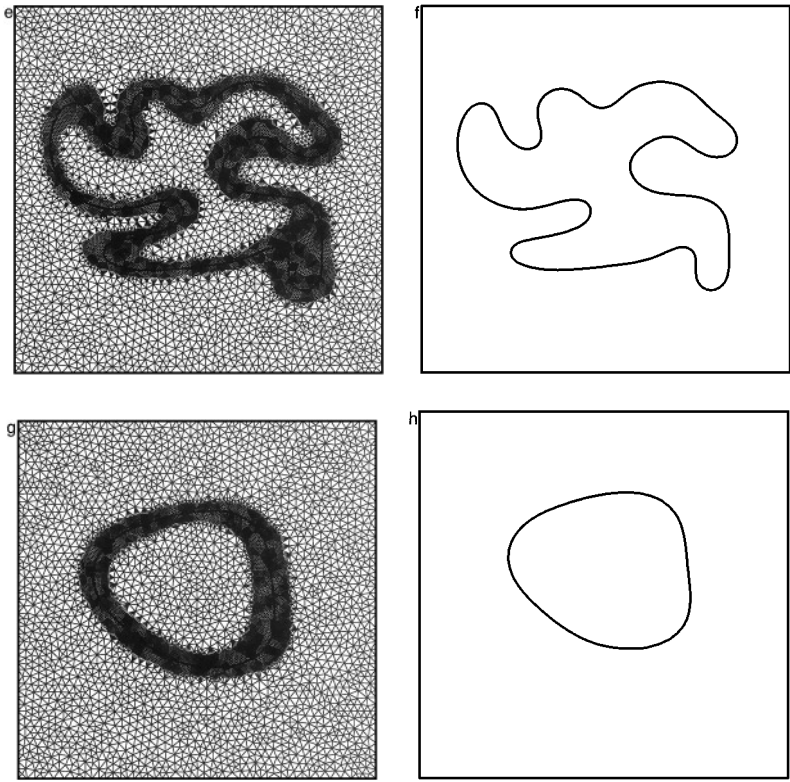


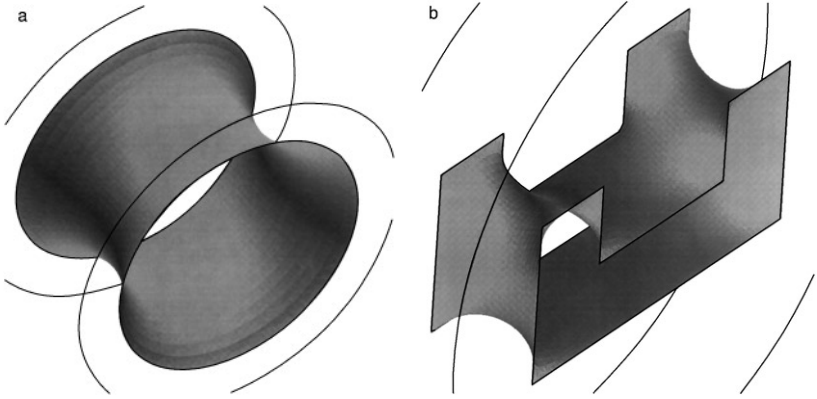
FIG. 25—Continued

has been used with  $\Delta t = 0.001$  and three levels of conformal adaptive refinement. Simplices with level set function  $|\phi| < 0.05$  were identified for refinement. As Figs. 25a–25f demonstrate, the calculation eventually evolves to a convex shape which eventually becomes a round point thereby verifying Grayson’s theorem.

## 8.2. Construction of Minimal Surfaces

One application of curvature flow is the construction of minimal surfaces. The first application of level set methods to the construction of minimal surfaces is due to Chopp [9]; his approach is as follows. Imagine a wire frame, with the goal of constructing a minimal surface that passes through that wire frame. One possible way to construct that minimal surface is to start with *any* surface that passes through that wire frame, and allow that surface to evolve under curvature flow, that is, with speed  $F = -\kappa$ , where  $\kappa$  is the curvature. If the surface remains attached to the wire frame as it flows, the end limiting result will be a surface of minimal curvature.

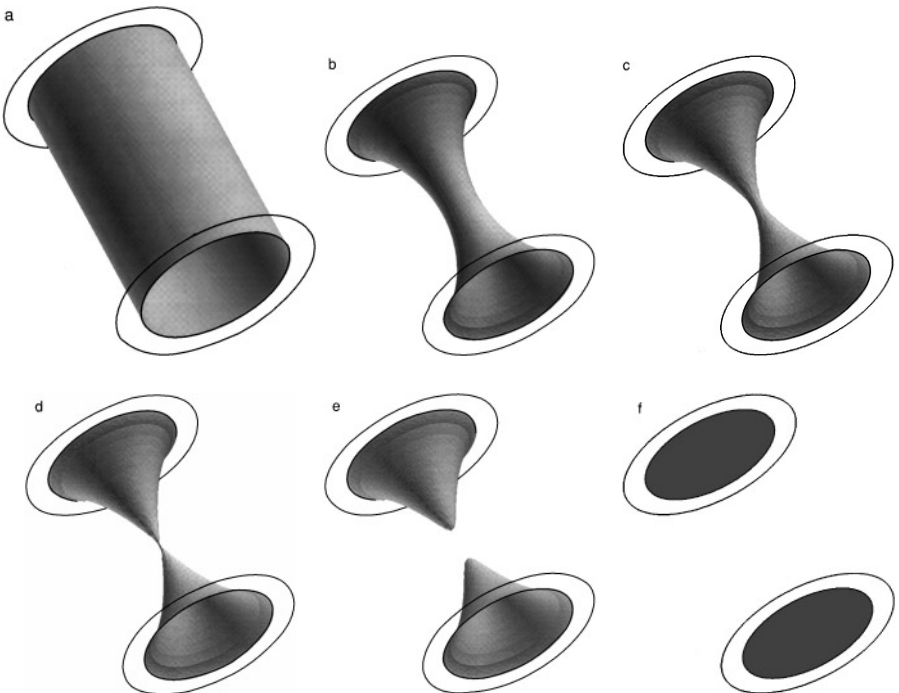
Chopp employed a level set method on a Cartesian mesh utilizing a finite difference scheme to advance the level set equation. Chopp restricted the movement of the zero level set so that it always remained attached to the surface; since a rectangular coordinate system was used, considerable care was applied to constructing interpolating boundary conditions between the wire frame and the neighboring grid points in order to ensure that the front remained “attached” to the frame as it evolved. In addition, this internal boundary condition on the level set equation caused considerable distortion in the gradient of the level set



**FIG. 26.** Minimal surfaces constructed through curvature flow.

function around the zero level set; a renormalization procedure was developed to circumvent this difficulty. The advantage of using a level set method to construct minimal surfaces is that one need not know in advance the final topology of the minimal surface; breaking and splitting is handled automatically as the surface evolves from its initial guess to the final state. For details, see [9].

Using a triangulated domain approach to this problem has the advantage that the triangulation can be arranged so that vertices fall on the wire frame, thus automatically constraining the front. Figure 26 shows calculated minimal surfaces for two ring pairs, one circular and one rectilinear. Both ring geometries are located less than a critical distance apart so that

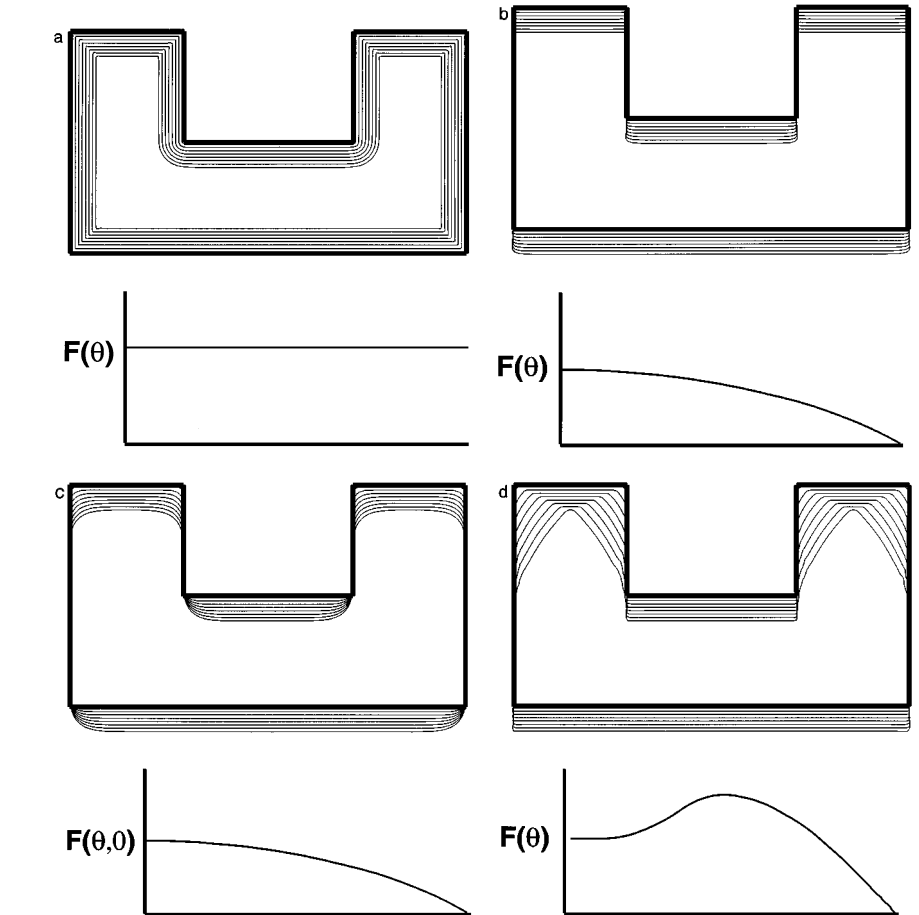


**FIG. 27.** Catenoid evolution through curvature flow undergoing topological change.

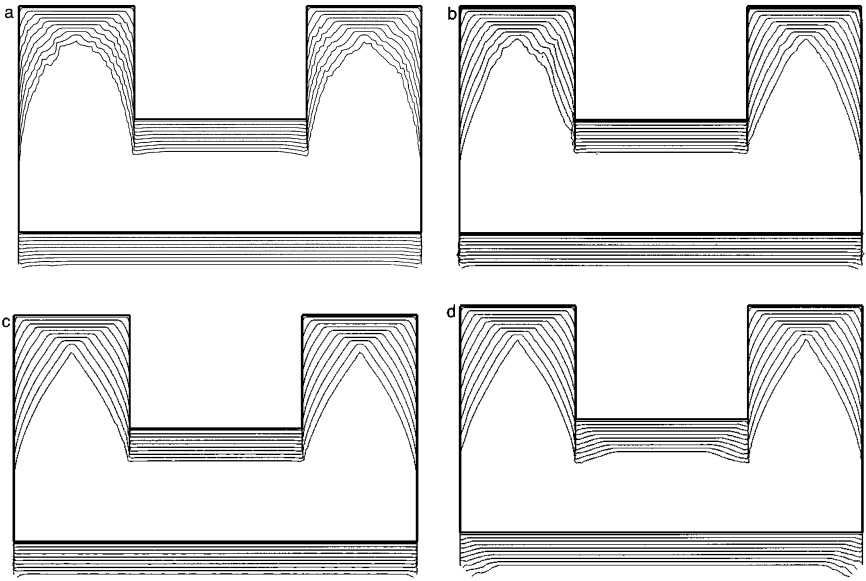
the curvature flow calculation eventually reaches an equilibrium. In the case of the circular ring geometry, the equilibrium shape is a catenoid. By lengthening the distance between the rings, the minimal surface pinches off and separates into two disks spanning the rings. Figure 27 shows the evolution of this sequence, starting from an initial guess and evolving towards the final minimal surface state.

## 9. NONCONVEX HAMILTONIANS

Next, the performance of the newly developed schemes for nonconvex Hamiltonians is studied. Specifically, nonconvex Hamiltonians are considered which model ion-milling used in the etching of a material surface during semiconductor manufacturing. In some problems, the rate at which an interface is etched away depends on the angle of incidence between the surface normal and the incoming beam. This yield function is often empirically fit from experiment and has been observed to cause such effects as faceting at corners. These problems were studied extensively using level set methods in [3–5], and the role of nonconvexity in the Hamiltonian was discussed at length. Here, those experiments are repeated in a triangulated setting using the schemes discussed earlier.



**FIG. 28.** Convex and nonconvex Hamiltonians. (a)  $F(\theta) = 1$ ; (b)  $F(\theta) = \cos(\theta)$ ; (c)  $F(\theta, \kappa) = \cos \theta - 0.1\kappa$ ; (d)  $F(\theta) = 5 \cos \theta - 4 \cos^3 \theta$ .



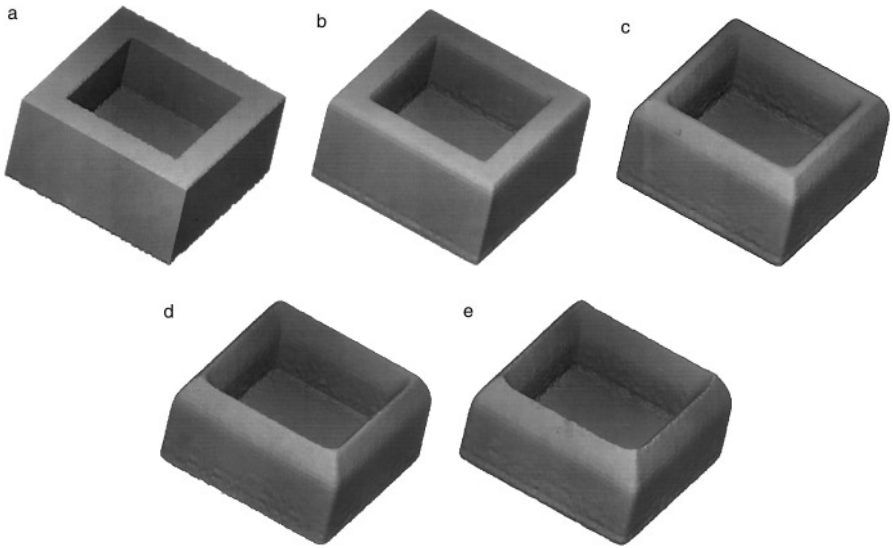
**FIG. 29.** Performance of various schemes for nonconvex etching problem. (a) Positive scheme; (b) explicit Petrov–Galerkin; (c) explicit Petrov–Galerkin with discontinuity capturing term; (d) implicit Petrov–Galerkin with discontinuity capturing term.

Figure 28 shows etching profiles for various different yield curves. Note that the last yield curve, Fig. 28d given by  $F(\theta) = 5 \cos \theta - 4 \cos^3 \theta$ , produces the faceting discussed above.

Figure 29 shows how various schemes capture the nonconvex profile. The positive and explicit Petrov–Galerkin schemes exhibit the wiggles symptomatic of the incorrect solution, see [3]. The explicit Petrov–Galerkin scheme with discontinuity capturing correctly computes the faceting effects for this problem. The implicit Petrov–Galerkin scheme also computes the faceting effects on the upper etched surface but seems to produce unphysical faceting on the lower surface. This phenomena is not well-understood and warrants further investigation.

### 9.1. Etching and Deposition in Semiconductor Manufacturing

In a series of papers [3–5], level set methods were developed and applied to surface topography evolution in etching and deposition processes used in semiconductor manufacturing. In these simulations, the ultimate speed of the propagating front depends on such factors as the flux of material from the source to the surface, visibility, material-dependent etch rates, re-sputtering and re-emission from the surface, and surface diffusion. Those calculations were performed using a fixed rectangular finite difference approximation to the equations of motion, together with an adaptive mesh narrow band technique to greatly ease the computational labor, see [2]. Here, the most straightforward of those simulations has been repeated using a triangulated domain scheme. The test geometry is a simple 3D contact well. The etching law is the nonconvex ion-milling law considered in the previous problem. The explicit Petrov–Galerkin with discontinuity capturing term is used to compute the numerical solution on the tetrahedral mesh. The sequence of figures from left to right and top to bottom in Fig. 30 shows the time evolution of the etched surface. The characteristic surface faceting associated with the nonconvex etch law is clearly seen. The calculation quality can be further improved by using dynamic mesh adaptation.



**FIG. 30.** The 3D contact well etching evolution using the nonconvex etching law.

## 10. CONCLUSIONS AND FUTURE WORK

A general algorithmic approach has been developed for computing the solutions of Hamilton–Jacobi and front propagation problems on triangulated domains in two and three space dimensions. In later work, our plan is to employ these techniques on interface problems where mesh adaptivity is critical, and problems in which nodal points on the interface are needed for accurate representation of internal boundary conditions, for example, in some aspects of semiconductor manufacturing and materials sciences. This work will be reported elsewhere.

## REFERENCES

1. R. Abgrall, Numerical discretization of the first-order Hamilton–Jacobi equation on triangular meshes, *Comm. Pure Appl. Math.* **49**, 1339 (1996).
2. D. Adalsteinsson and J. A. Sethian, A fast level set method for propagating interfaces, *J. Comput. Phys.* **118**(2), 269 (1995).
3. D. Adalsteinsson and J. A. Sethian, A unified level set approach to etching, deposition and lithography. I. Algorithms and two-dimensional simulations, *J. Comput. Phys.* **120**(1), 128 (1995).
4. D. Adalsteinsson and J. A. Sethian, A unified level set approach to etching, deposition and lithography. II. Three-dimensional simulations, *J. Comput. Phys.* **122**(2), 348 (1995).
5. D. Adalsteinsson and J. A. Sethian, A unified level set approach to etching, deposition and lithography. III. Complex simulations and multiple effects, *J. Comput. Phys.*, in press.
6. M. Bern and D. Epstein, *Mesh Generation and Optimal Triangulation*, Technical Report CSL-92-1, Xerox PARC, March 1992.
7. L. P. Chew, *Guaranteed-Quality Triangular Meshes*, Technical Report TR 89-983, Cornell University Department of Computer Science, March 1989.
8. L. P. Chew, Guaranteed-quality mesh generation for curved surfaces, in *Proceedings of the 9th ACM Symposium on Computational Geometry*, 1993.
9. D. L. Chopp, Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.* **106**, 77 (1993).
10. L. Corrias, M. Falcone, and R. Natalini, Numerical schemes for conservation laws via Hamilton–Jacobi equations, *Math. Comp.* **64**(210), 555 (1995).

11. M. G. Crandall and P. L. Lions, Two approximations of solutions of Hamilton–Jacobi equations, *Math. Comp.* **43**(167), 1 (1984).
12. M. G. Crandall and L. Tartar, Some relations between nonexpansive and order preserving mappings, *Proc. Amer. Math. Soc.* **78**(3), 385 (1980).
13. H. Deconinck, R. Struijs, and P. L. Roe, *Compact Advection Schemes on Unstructured Grids*, Technical Report, VKI, VKI LS 1993-04, Computational Fluid Dynamics, 1993.
14. M. Grayson, The heat equation shrinks embedded plane curves to round points, *J. Differential Geom.* **26**, 555 (1989).
15. P. Hansbo, Explicit streamline diffusion finite element methods for the compressible Euler equations in conservation variables, *J. Comput. Phys.* **109**(2), 274 (1993).
16. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method* (Cambridge Univ. Press, Cambridge, 1987).
17. S. N. Kruzkov, First order quasilinear equations in independent variables, *Mat. Sb.* **81**, 228 (1970).
18. P. L. Lions, *Generalized Solutions of Hamilton–Jacobi Equations* (Pitman, Marshfield, MA, 1982).
19. P. L. Lions and P. E. Souganidis, Convergence of MUSCL and filtered schemes for scalar conservation laws and Hamilton–Jacobi equations, *Numer. Math.* **69**(4), 441 (1995).
20. R. L. Löhner and J. D. Baum, *Numerical Simulation of Shock Interaction with Complex Geometry Structures Using a New h-Refinement Scheme on Unstructured Grids*, Technical Report, 28th AIAA Aerospace Sciences Meeting, 1990.
21. B. Milne, *Adaptive Level Set Methods Interfaces*, Technical Report, University of California Berkeley Department of Mathematics, 1995.
22. S. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulation, *J. Comput. Phys.* **79**, 12 (1988).
23. P. L. Roe, *Linear Advection Schemes on Triangular Meshes*, Technical Report CoA 8720, Cranfield Institute of Technology, 1987.
24. P. L. Roe, “*Optimum*” *Upwind Advection on a Triangular Mesh*, ICASE 90-75, 1990.
25. J. Ruppert, *A New and Simple Algorithm for Quality Two-Dimensional Mesh Generation*, Technical Report UCB/CSD 92/694, University of California Berkeley, Department of Computer Science, 1992.
26. J. A. Sethian, Curvature and the evolution of fronts, *J. Math. Phys.* **101**, 487 (1985).
27. J. A. Sethian, Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservation laws, *J. Differential Geom.* **31**, 131 (1990).
28. J. A. Sethian, *Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces* (Acta Numerica, Cambridge Univ. Press, Cambridge, UK, 1995).
29. J. A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Natl. Acad. Sci. U.S.A.* **93**(4), 1591 (1996).
30. J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Sciences* (Cambridge Univ. Press, Cambridge, UK, 1996).
31. M. S. Shephard, *Parallel Automated Adaptive Procedures for Unstructured Meshes*, Technical Report AGARD Report R-907, 1995.
32. C. W. Shu and S. Osher, Efficient implementation of essentially nonoscillatory shock-capturing scheme, *J. Comput. Phys.* **77**, 439 (1988).
33. H. K. Zhao, T. Chan, B. Merriman, and S. Osher, A variational level set approach to multiphase motion, *J. Comput. Phys.* **127**, 179 (1996).