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

Timothy J. Barth *
Information Sciences Directorate
NASA Ames Research Center

James A. Sethian †‡
Department of Mathematics
University of California, Berkeley

Abstract

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 certain forms of the H-J equations. Unfortunately, the basic scheme lacks proper Lipschitz continuity of the numerical Hamiltonian. By employing a "virtual" edge flipping technique, 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 naturally exhibit proper Lipschitz continuity of the numerical Hamiltonian. Finally, a class of Petrov-Galerkin approximations are 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

*barth@nas.nasa.gov
†sethian@math.berkeley.edu
‡Supported in part by the Office of Naval Research

also requires the numerical approximation of a mean curvature term. A simple lumped-Galerkin approximation is presented 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.

1 Introduction

In this paper, three related equations are considered

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

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

2. The general level set equation

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

3. The Eikonal equation

$$F(x)\ |\nabla u| = \pm 1, \qquad F(x) > 0, \qquad x \in \Omega \tag{3}$$

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

$$\kappa = \nabla \cdot \frac{\nabla u}{|\nabla u|}. \tag{4}$$

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

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

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

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

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 (5) then u is a entropy solution of the conservation law (6). 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-difference 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 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 built in rectangular finite difference settings, see [21], such methods,

especially in the presence of parabolic curvature terms have additional complications.

- **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 nodes, 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. Two simple space-time formulations for the Petrov-Galerkin scheme are developed.

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 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)

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

so that the Eikonal equation (3) can also be modeled by dropping the time derivative term. Let $\mathcal{T}$ denote a triangulation set in $\mathbf{R}^d$, $\mathcal{T} = \{T_1, T_2, \ldots, T_{|\mathcal{T}|}\}$, composed of simplices covering Ω 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 $V = \{v_1, v_2, \ldots, v_{|V|}\}$. Unless otherwise stated, the solution on $\mathcal{T}$ is approximated using a standard piecewise linear finite element subspace denoted by $\mathcal{V}_h^1$. For $(x,t) \in \mathcal{V}_h^1 \times \mathbf{R}^+$, the numerical solution at vertex v_j at a time $n\Delta t$ is denoted by u_j^n. Next, consider numerical approximations of Eqn. (7) of the form

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

with

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

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:

   $$u(x_j, n\Delta t) = u_0(x_j) - n\Delta t \; H(\nabla u_0). \tag{10}$$

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

   $$\mathcal{H}(\nabla u, \nabla u, \ldots, \nabla u) = H(\nabla u) \tag{11}$$

   and it can be shown that at least one nonzero α_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.

$$u^n \geq v^n \qquad \text{implies} \qquad u^{n+1} \geq v^{n+1}. \tag{12}$$

For self-maps of the form

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

it is sufficient to show that

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

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

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

with

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

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

$$\min_{i=1,2,\ldots,|V|} u_i^n \leq u_j^{n+1} \leq \max_{i=1,2,\ldots,|V|} u_i^n, \quad \forall \; j = 1,2,\ldots,|V| \tag{17}$$

as well as the local maximum principle

$$\min_{i \in \text{supp}_j} u_i^n \leq u_j^{n+1} \leq \max_{i \in \text{supp}_j} u_i^n, \quad \forall \; j = 1,2,\ldots,|V| \tag{18}$$

where 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 α_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 $N_i(x_j) = \delta_{ij}$

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

The gradient ∇u in simplex T is readily computed in terms of gradients of the basis functions

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

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

$$\nabla u = \frac{1}{d\ \mathrm{meas}(T)} \sum_{i=1}^{d+1} \overrightarrow{n}_i\ u_i. \tag{21}$$

Note that

$$\sum_{i=1}^{d+1} \overrightarrow{n}_i = 0 \tag{22}$$

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

$$\nabla H \equiv \begin{pmatrix} \frac{\partial H}{\partial u_x} \\ \frac{\partial H}{\partial u_y} \\ \vdots \end{pmatrix}. \tag{23}$$

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

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

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

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

When combined with the monotonicity conditions (14), certain orientations of ∇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.

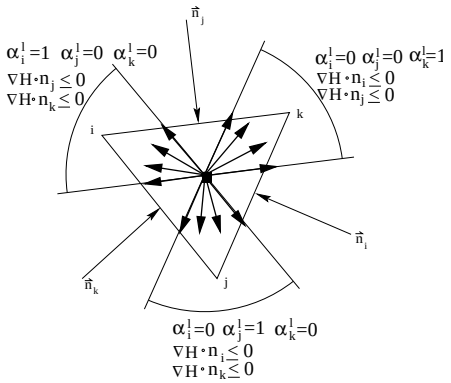


Figure 1: Typical 2-D simplex T_l showing inward pointing normals, index convention, and sector demarcation.

$\underline{\nabla H \cdot n_j \leq 0, \nabla H \cdot n_k \leq 0:}$

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

$\underline{\nabla H \cdot n_i \leq 0, \nabla H \cdot n_k \leq 0:}$

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

$\underline{\nabla H \cdot n_i \leq 0, \nabla H \cdot n_j \leq 0:}$

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

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

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

where

$$\alpha_j^l = \begin{cases} 1 & \nabla H \cdot n_k \leq 0, \;\; k = 1, 2, \ldots, d+1, j \neq k \\ 0 & \text{otherwise} \end{cases}, \qquad x \in T_l \tag{30}$$

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 α_j^l coefficient when $u(x)$ varies linear over the entire support of the scheme. From Fig. 2 it is clear that this must always happen. Consider a constant gradient vector ∇H located at a vertex 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 α_0^m. Unfortunately, the scheme suffers from a serious flaw. The numerical Hamil-

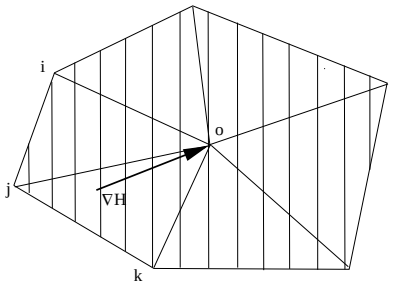


Figure 2: Triangles surrounding v_0 with linearly varying $u(x)$. Isolevels of constant $u(x)$ are shown as dotted lines and the arrow denotes the constant gradient vector ∇H at vertex v_0.

tonian fails to be a Lipschitz continuous function. This non-Lipschitzian behavior is shown in Fig. 3a-b. 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

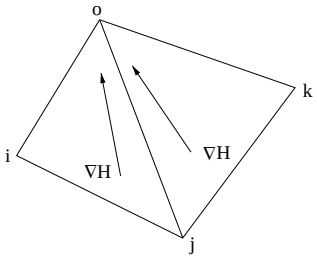


Fig. 3a. Two v_0 updates.

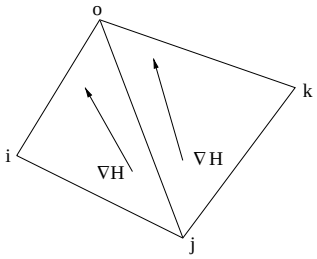


Fig. 3b. No v_0 update.

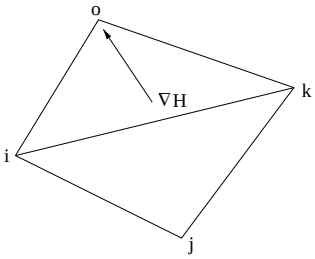


Fig. 3c. One v_0 update.

Figure 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.

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

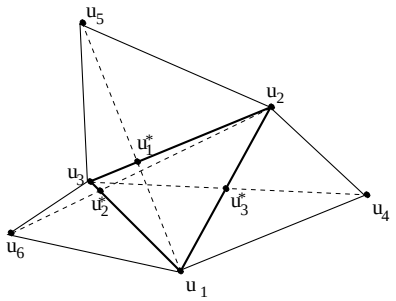


Figure 4: Virtual edge flipping strategy. Given the triangle $T(v_1, v_2, v_3)$ with vertex solution values u_1, u_2, and u_3, additional solution values u_1^*, u_2^*, u_3^* and positions are computed based on (virtual) diagonal flipped edges $e(v_3, v_4), e(v_1, v_5), e(v_2, v_6)$.

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 $T(v_1, v_2, v_3)$, 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

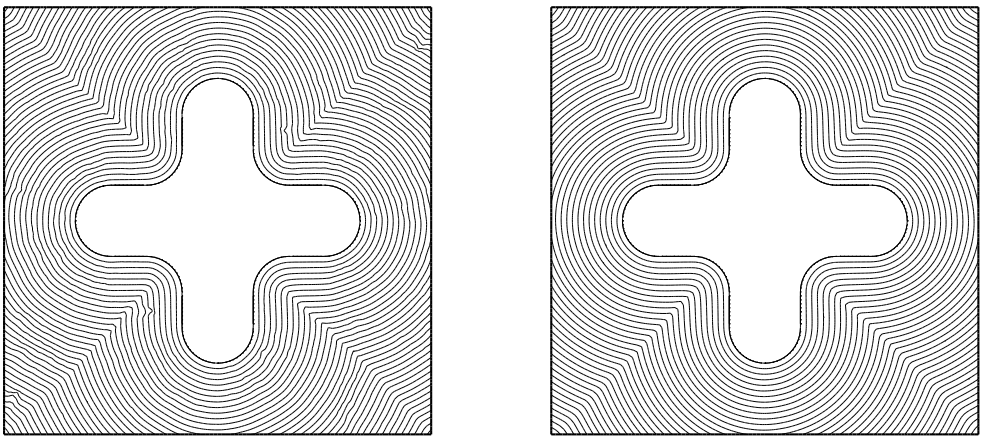


Figure 5: Level set solutions obtained without (left) and with (right) virtual edge flipping.

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. 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 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 flip-

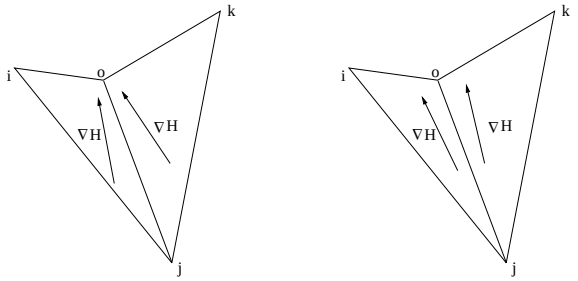


Fig. 6a. Two v_0 updates. Fig. 6b. No v_0 update.

Figure 6: Non-Lipschitzian behavior of the monotone scheme which cannot be repaired by edge flipping.

ping 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. In this new framework, it will be sufficient to show that the $d+1$ coefficients α_j^l for the simplex T_l are continuous functions of the gradient vector ∇u. For accuracy reasons α_j^l should also be bounded and positive. Note that since monotone schemes depend fundamentally on the quantity

$$\frac{\partial\Big(H(\nabla u)\Big)_T}{\partial u_i} = \frac{1}{d}\nabla H \cdot \overrightarrow{n_i},$$

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

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

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

$$\Big(H(\nabla u)\Big)_T = \frac{1}{p}\nabla H \cdot \nabla u = \frac{1}{d\ p}\sum_{i=1}^{d+1}(\nabla H \cdot \overrightarrow{n_i})u_i. \tag{32}$$

Next define

$$K_i = \frac{\nabla H \cdot \overrightarrow{n_i}}{d\ p} \tag{33}$$

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

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

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

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

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

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

with

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

When written in matrix form

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

it becomes clear that the decomposition produces a sign pattern useful in constructing a positive coefficient schemes satisfying Eqn. (15). To do so simply let

$$\alpha_l^i = \frac{\delta_i}{\Big(H(\nabla u)\Big)_{T_l}} \tag{39}$$

and insert into our prototype numerical Hamiltonian

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

Since the coefficient terms depend continuously on ∇H and the simplex geometry, Lipschitz continuity of the numerical Hamiltonian is obtained. 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 *non-obtuse* triangulations, α_j^l are nonnegative and hence bounded. In this case the conditions for a positive coefficient scheme are obtained:

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

with

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

For general (obtuse) triangulations and homogeneous Hamiltonians, additional nonlinearity must be introduced into the formulation so that a positive

coefficient scheme is obtained. Deconinck [13] in the context of conservation law equation discretization, suggests the a simple nonlinear modification of α_j^l to obtain a positive coefficient scheme for general triangulations:

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

The inclusion of this modification produces a robust scheme for the H-J

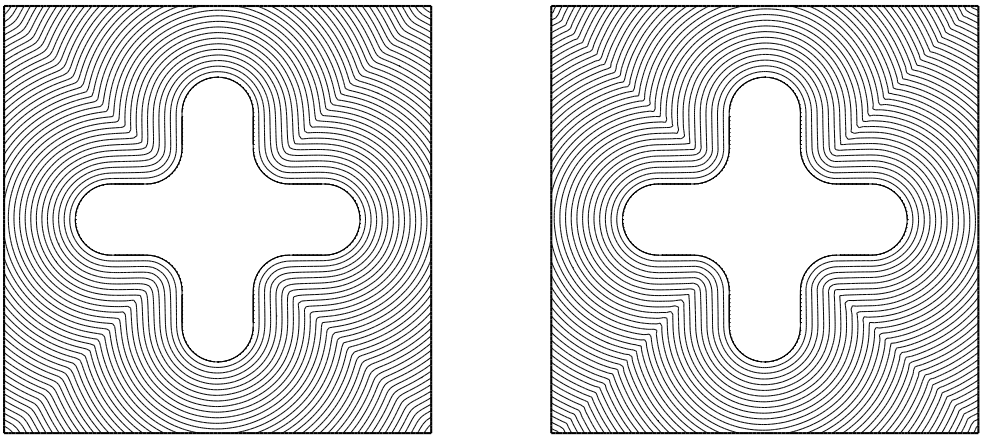


Figure 7: Level set solution obtained using monotone scheme with virtual edge flipping (left) and positive coefficient scheme (right).

equations (with source term), i.e.

$$\mathcal{H}_j(\nabla u_1, \nabla u_2, \ldots, \nabla u_{|\mathcal{T}|}, x) = \frac{\sum_{l=1}^{|\mathcal{T}|} \tilde{\alpha}_j^l \left(H(\nabla u) - f(x)\right)_{T_l}}{\sum_{l=1}^{|\mathcal{T}|} \tilde{\alpha}_j^l \ \mathrm{meas}(T_l)} \tag{44}$$

Note that this nonlinear modification still permits the scheme to be written in the positive coefficient form (41). Repeating the example calculation of the previous section shown in Fig. 4, Fig. 7 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

$$u_t + H(\nabla u) = 0. \tag{45}$$

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

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

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

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

i.e. $\mathcal{V}^h$ 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 $\mathcal{V}^n$ 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 Γ:

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

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 $w^h \in \mathcal{V}_n^h$

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

with

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

for $\tau > 0$ and $\nabla H = \nabla H(\nabla u)$ everywhere in (50). The motivation for the least-squares stabilization $B(u,w)_{ls}$ comes from looking at the variation of the quadratic potential

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

for constant ∇H

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

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

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

and

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

Removing the assumption of constant ∇H in the calculation of the first variation yields

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

where $\mathrm{Hess}(H)$ 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

$$\mathrm{Hess}(H) \ \nabla u = p \ (p-1) \ \nabla H \tag{56}$$

so that

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

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 Eqn.(51) is recovered. For this reason, we defer further consideration of this new term.

Our choice of the parameter τ is motivated from standard practice for conservation laws:

$$\tau = \left(\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2|\nabla H|}{ph} \right)^2 \right)^{-1/2} \tag{58}$$

with the Δt 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:

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

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

$$\int_\Omega \int_{I^n} u u_t\; d\,t\; d\,x = \frac{1}{2}\int_\Omega [u^2(t_-^{n+1}) - u^2(t_+^n)]\; d\,x \tag{60}$$

and the jump integral

$$\int_\Omega u(t_+^n)[u(t_+^n) - u(t_-^n)]\; d\,x = \frac{1}{2}\int_\Omega [u(t_+^n) - u(t_-^n)]^2\; d\,x + \frac{1}{2}\int_\Omega [u^2(t_+^n) - u^2(t_-^n)]\; d\,x \tag{61}$$

yields

$$\begin{aligned}
\int_\Omega \int_{I^n} u u_t\; d\,t\; d\,x + \int_\Omega u(t_+^n)[u(t_+^n) - u(t_-^n)]\; d\,x &= \frac{1}{2}\int_\Omega [u(t_+^n) - u(t_-^n)]^2\; d\,x \\
&+ \frac{1}{2}\int_\Omega [u^2(t_-^{n+1}) - u^2(t_-^n)]\; d\,x
\end{aligned} \tag{62}$$

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). ■

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 $\nabla H =$ 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_-.

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:

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

In this formula, ϵ 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

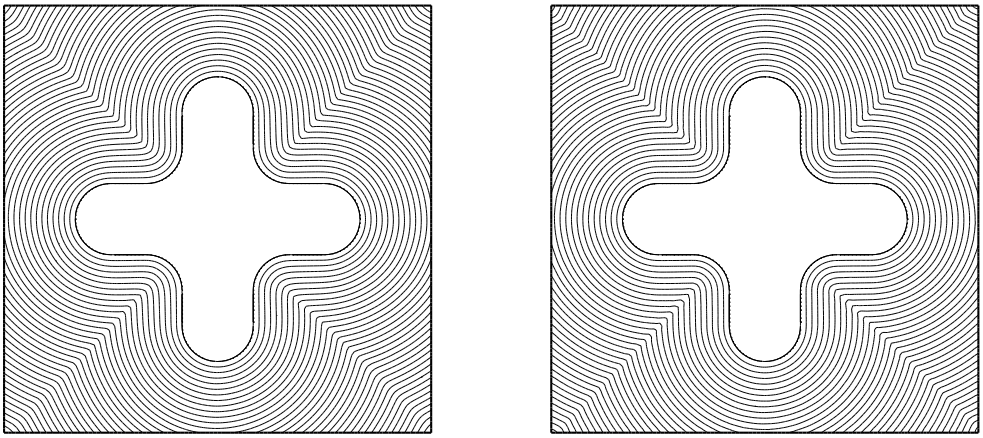


Figure 8: Level set solution obtained using monotone scheme with virtual edge flipping (left) and Petrov-Galerkin scheme (right).

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 7 is repeated for the Petrov-Galerkin discretization with discontinuity capturing term ($\sigma_{dc} = .1$). Figure 8 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

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

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 Δt

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

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

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

positivity of coefficients in time, i.e. that u_j^{n+1} is a positive weighted combination of u^n, yields the following CFL-like time step restriction similar to the monotone scheme:

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

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

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

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:

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

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 $\mathcal{G}(u) : \mathbf{R}^{|V|} \to \mathbf{R}^{|V|}$ 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 Eqn. (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

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

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 $\mathcal{G}(u)$:

$\mathbf{R}^{|V|} \to \mathbf{R}^{|V|}$ is an order preserving self-map. From order preservation, it follows that

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

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

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

Considering the composition term

$$\mathcal{G}(\mathcal{G}(u)) = T(T(u)u)T(u)u. \tag{75}$$

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 Sec. 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

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

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

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

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

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

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.

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

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 $x \in T, t \in [t_+^n, t_-^{n+1}]$

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

as depicted in Fig. 9. 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 scheme. Using midpoint quadrature formula to evaluate Petrov-Galerkin integrals, let $\overline{u}$ denote a reference state at the quadrature point location. Define $\nabla\overline{H} = \nabla H(\overline{u})$ and $\overline{\tau} = \tau(\overline{u})$ and the following bilinear forms:

$$B(u,w;\overline{u})_{gal} = \int_{I^n}\int_\Omega w\left(u_t + p^{-1}\nabla\overline{H}\cdot\nabla u\right) d\Omega\, dt$$

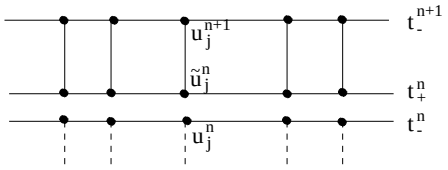


Figure 9: One-dimensional space-time slab depicting multiple data values from t_-^n to t_+^n.

$$B(u,w;\overline{u})_{ls} = \int_{I^n}\int_{\Omega}\Big(w_t + p^{-1}(\nabla\overline{H}\cdot\nabla w)\Big)\,\overline{\tau}\,\Big(u_t + p^{-1}(\nabla\overline{H}\cdot\nabla u)\Big)\,d\Omega\,dt \tag{78}$$

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

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

2. For s = 1, m

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

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

   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

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

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

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

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

$$
\begin{aligned}
& T \xrightarrow{local} \text{simplex}(x_1, x_2, \ldots, x_{d+1}) \\
& N_i(x) = \{N_i(x) \in \mathcal{P}_1 \mid N_i(x_j) = \delta_{ij},\ j = 1, 2, \ldots, d+1,\ x \in T\} \\
& \overline{F} = \frac{1}{\text{meas}(T)} \int_T F(\nabla\phi)\ dx \\
& \overline{f} = \frac{1}{\text{meas}(T)} \int_T f(x)\ dx \\
& \overrightarrow{n}_i = d\ \text{meas}(T)\ \nabla N_i \\
& \nabla\phi = \sum_{j=1}^{d+1} \nabla N_j\ \phi_j \\
& K_i = \frac{\overline{F}\ \nabla\phi \cdot \overrightarrow{n}_i}{d\ |\nabla\phi|} \\
& \delta\phi = \sum_{l=1}^{d+1} K_l\ \phi_l \\
& \delta\phi_i = K_i^+ \left(\sum_{l=1}^{d+1} K_l^- \right)^{-1} \sum_{l=1}^{d+1} K_l^- (\phi_i - \phi_l) \\
& \tilde{\alpha}_i = \frac{\max\left(0, \delta\phi_i / \delta\phi\right)}{\sum_{l=1}^{d+1} \max\left(0, \delta\phi_l / \delta\phi\right)} \\
& \phi_i^* = \phi^* + \tilde{\alpha}_i\ (\delta\phi - \overline{f}\ \text{meas}(T)) \\
& w_i = w_i + \tilde{\alpha}_i\ \text{meas}(T)
\end{aligned}
$$

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 $v_i \in V$

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

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

Endfor

5.2 The Explicit Petrov-Galerkin Scheme for the Level Set Equation

Once again consider the level set equation

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

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

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

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

$$\begin{aligned}
&T \xrightarrow{local} \text{simplex}(x_1, x_2, \ldots, x_{d+1}) \\
&N_i(x) = \{N_i(x) \in \mathcal{P}_1 \mid N_i(x_j) = \delta_{ij},\ j = 1, 2, \ldots, d+1,\ x \in T\} \\
&\overline{F} = \frac{1}{\text{meas}(T)} \int_T F(\nabla\phi)\ dx \\
&\overline{f} = \frac{1}{\text{meas}(T)} \int_T f(x)\ dx \\
&\overrightarrow{n}_i = d\ \text{meas}(T)\ \nabla N_i \\
&\nabla\phi = \sum_{j=1}^{d+1} \nabla N_j\ \phi_j \\
&K_i = \frac{\overline{F}\ \nabla\phi \cdot \overrightarrow{n}_i}{d\ |\nabla\phi|} \\
&\tau = \tau(\phi) \\
&\delta\phi = \sum_{l=1}^{d+1} K_l\ \phi_l \\
&\alpha_i = \frac{1}{d} + \tau\ K_i \\
&\phi_i^* = \phi^* + \alpha_i\ (\delta\phi - \overline{f}\ \text{meas}(T)) \\
&w_i = w_i + \alpha_i\ \text{meas}(T)
\end{aligned}$$

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 $v_i \in V$

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

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 Ω 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 θ 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 Ω 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_∞.

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 10 shows the mesh

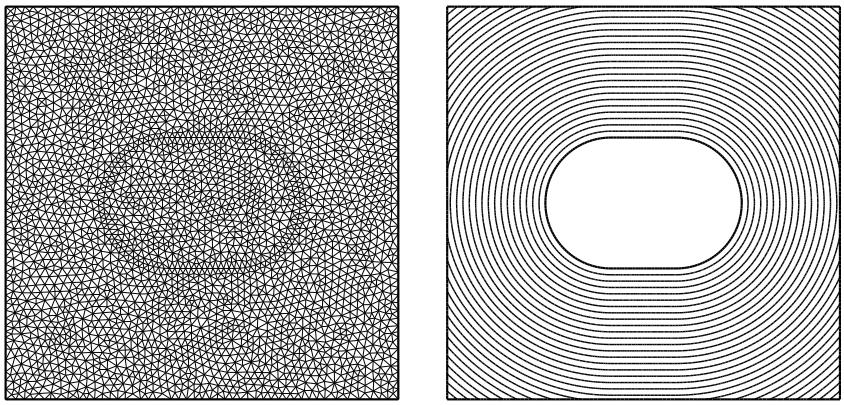


Figure 10: Smooth Level Set Solns: Coarsest Mesh (left) and Crossing Time Solution (right).

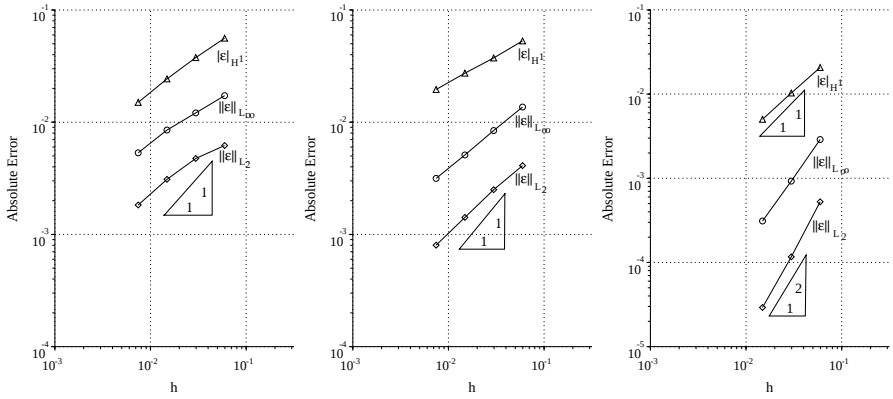


Figure 11: Smooth Level Set Solns: Absolute crossing time solution error. Positive coefficient scheme (left), explicit Petrov-Galerkin (middle), and implicit Petrov-Galerkin (right).

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. 11. 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_∞ 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 now computed as a field solution of the Eikonal equation. The mesh and

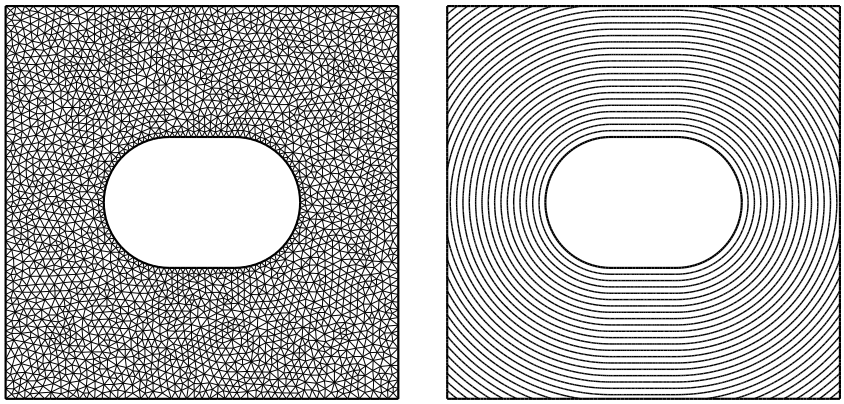


Figure 12: Smooth Eikonal Solns: Coarsest Mesh (left) and Solution (right).

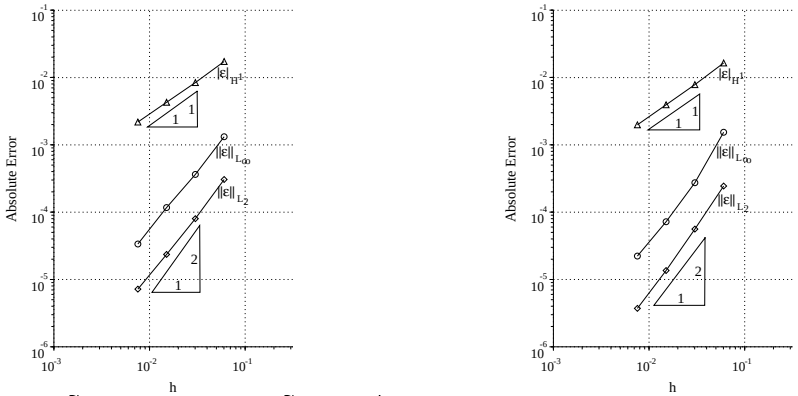


Figure 13: Smooth Eikonal Solns: Absolute solution error. Positive coefficient scheme (left), explicit Petrov-Galerkin (right).

field solution using the positive coefficient scheme are shown in Fig. 12. The specified oval curve is now a *boundary* for the domain with imposed boundary condition $\phi = 0$. 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. 13 for the positive coefficient scheme and the Petrov-Galerkin scheme. Both schemes perform optimally; second order accurate in L_2 and L_∞ 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 14 shows the mesh and

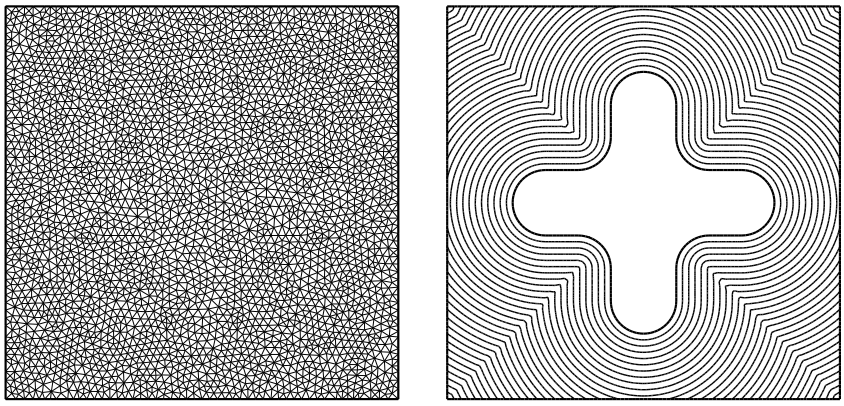


Figure 14: Non-Smooth Level Set Solns.: Coarsest Mesh (left) and Crossing Time Solution (right).

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. 15. The two explicit schemes again show slightly less than first order accuracy in L_2 and L_∞ with further degradation in the H^1 semi-norm. The implicit Petrov-Galerkin with discontinuity capturing term parameter $\sigma_{dc} = .1$ retains second order accuracy in L_2, first order accuracy in L_1, and one-half order accuracy in the H^1 semi-norm.

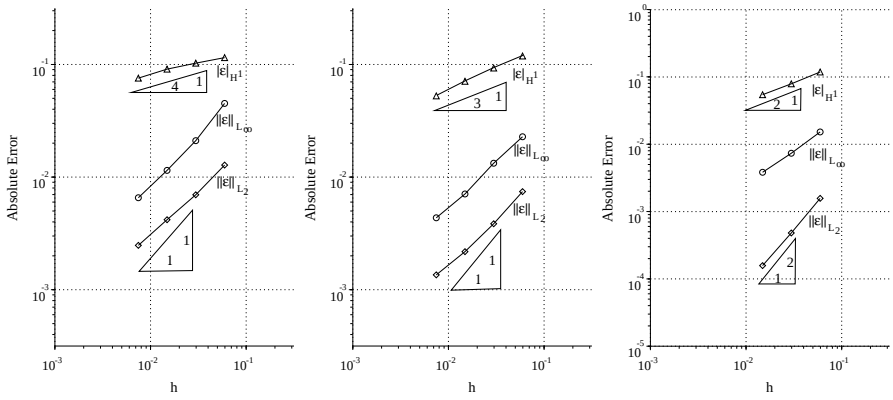


Figure 15: Non-Smooth Level Set Solns: Absolute crossing time solution error. Positive coefficient scheme (left), explicit Petrov-Galerkin (middle), and implicit Petrov-Galerkin (right).

5.3.4 Problem 2B: Non-Smooth Solutions to Eikonal Eqn.

As a last test problem, the crossing time solution from Problem 2A is now computed as a field solution of the Eikonal equation. The specified star-

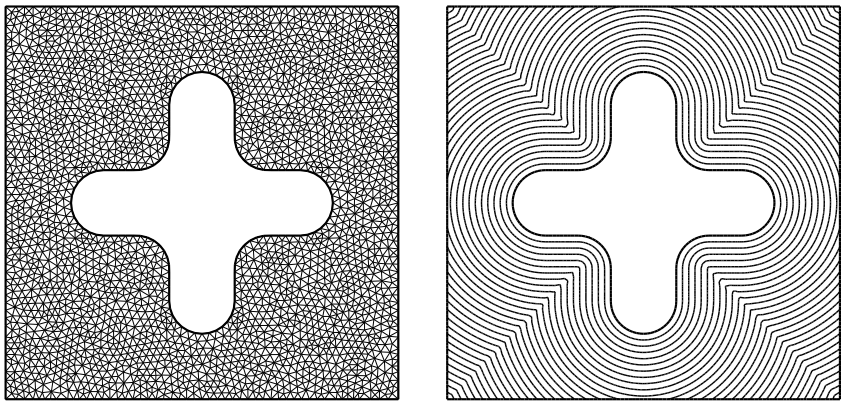


Figure 16: Non-Smooth Eikonal Solns.: Coarsest Mesh (left) and Solution (right).

shaped curve is now 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. 16. Absolute solution errors are graphed in Fig. 17 for the positive coefficient scheme and the Petrov-Galerkin scheme. Again,

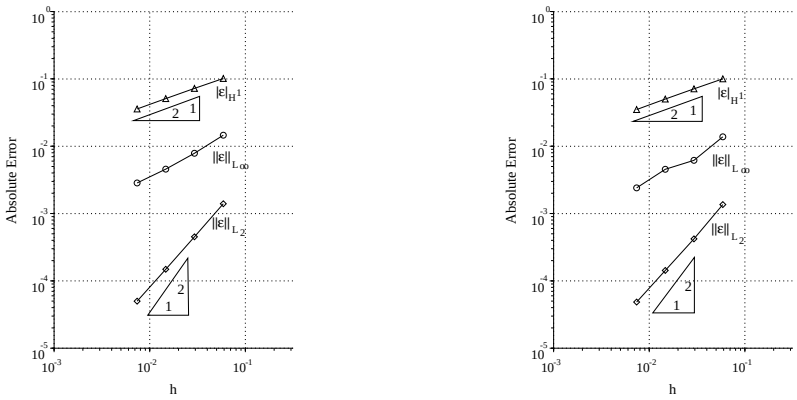


Figure 17: Non-Smooth Eikonal Solns.: Absolute solution error. Positive coefficient scheme (left), Petrov-Galerkin (right).

both the positive coefficient and Petrov-Galerkin schemes with discontinuity capturing term perform very similarly. Second order accuracy is obtain in L_2, first order accuracy in L_∞, 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 $\kappa\,|\nabla\phi|$ for the form

$$\phi_t + (F(\nabla\phi) - \epsilon\kappa)\;|\nabla\phi| = 0 \tag{81}$$

for locally constant ϵ. Recall that κ is the mean curvature of the level set function ϕ

$$\kappa = \nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}. \tag{82}$$

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 Γ:

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

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

$$\begin{aligned}
&\int_\Omega wu_t\;dx + \epsilon\int_\Omega \frac{\nabla w\cdot\nabla\phi}{|\nabla\phi|}\;dx = 0\\
&\text{with}\\
&\phi(x,0) = \phi_0(x), \qquad x\in\Omega\\
&\phi(x,t) = g(x), \qquad x\in\Gamma.
\end{aligned} \tag{84}$$

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

Lemma: The model equation (84) with C^0 linear triangular elements $\mathcal{V}^h\subset\mathcal{S}^h\in\mathcal{P}_1(T)$ 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 $N_i(x)$, $\phi(x) = \sum_{j=1}^{3} N_j(x)\phi_j, \quad x \in T$

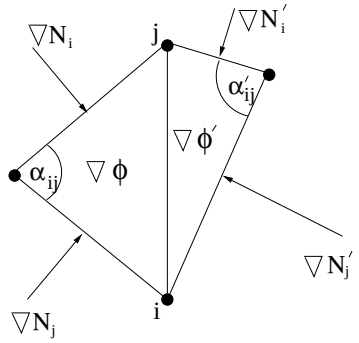
$$\int_T \frac{\nabla w \cdot \nabla \phi}{|\nabla \phi|}\, dx = \sum_{i=1}^{3}\sum_{j=1}^{3} w_i\ \phi_j\ \frac{\nabla N_i \cdot \nabla N_j}{|\nabla \phi|}\text{meas}(T). \tag{85}$$

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

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

where $\mathcal{N}_i$ denotes the set of vertices adjacent to vertex v_i with weights

$$\begin{aligned} W_j^i &= \left[\frac{\nabla N_i \cdot \nabla N_j}{|\nabla \phi|} + \frac{\nabla N_i' \cdot \nabla N_j'}{|\nabla \phi|'}\right] \\ &= \frac{1}{2}\left[\frac{\text{cotan}(\alpha_{ij})}{|\nabla \phi|} + \frac{\text{cotan}(\alpha_{ij}')}{|\nabla \phi|'}\right]. \end{aligned} \tag{87}$$



Discretization weight geometry for the edge $e(v_i, v_j)$

Since the discretization formula must hold for arbitrary values of w_i at interior vertices, it can be concluded that for all interior vertices v_i

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

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

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

so that

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

From Eqn. (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

$$|\nabla \phi| \ \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \tag{91}$$

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

$$|\nabla \phi| \ \nabla \cdot \left. \frac{\nabla \phi}{|\nabla \phi|} \right|_i \approx |\nabla \phi|_i \ \frac{\sum_{j \in \mathcal{N}_i} W_j^i \ (u_j - u_i)}{\sum_{T \in N_i} \text{meas}(T)} \tag{92}$$

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

$$(\nabla \phi)_i = \frac{\sum_{T \in N_i} \int_T \nabla u \ dx}{\sum_{T \in N_i} \text{meas}(T)}. \tag{93}$$

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.

7 Mesh Adaptivity

One of the central virtues of a triangulated domain formulation, as opposed to a rectilinear framework, is the ease in implementing mesh adaptivity. 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. 18 shows three different uses of mesh adaptation for evolving front problems. Fig. 18a shows conformal adaptation around the front itself; Fig. 18b shows Steiner adaptation around regions of high curvature in the evolving front; Fig. 18c shows interface conforming Steiner adaptation around regions of high curvature in the evolving solution.

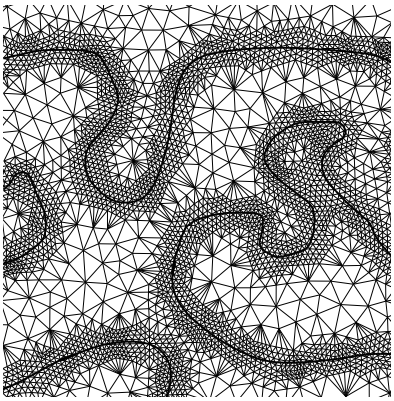


Fig. 18a. Interface proximity adaptation.

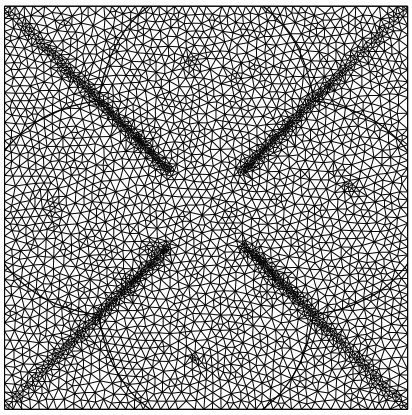


Fig. 18b. Curvature adaptation.

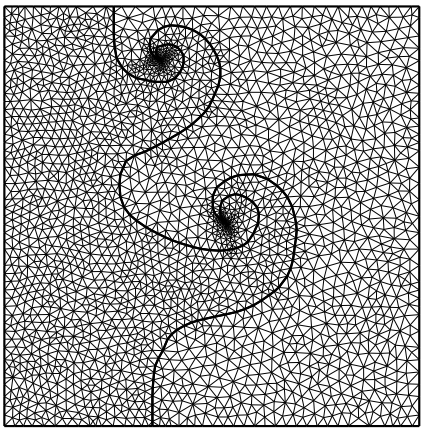
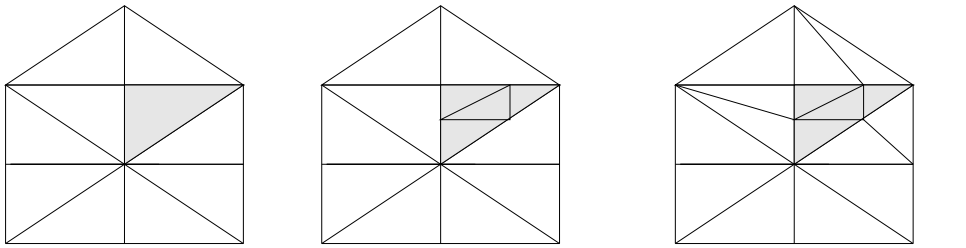


Fig. 18c. Interface Conforming Adaptation.

Figure 18: Various mesh adaptation techniques for evolving interface problems.

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 19). 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.



(a) Candidate triangle. (b) Conformal subdivision. (c) Resolution of subdivision.

Figure 19: Conformal refinement steps for a 2D triangle.

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 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. 20, 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

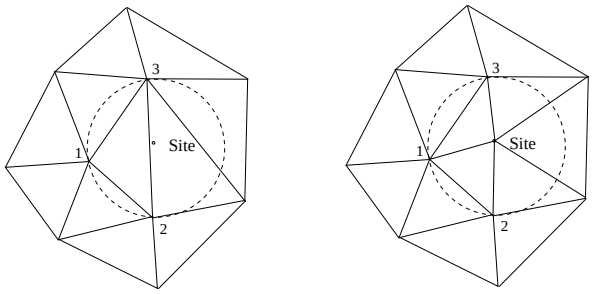


Figure 20: Steiner refinement: Site insertion at circumcenter location (left) and edge reconfiguration (right).

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 passes only if: (1) it is well-shaped, i.e. the smallest angle is greater than $30°$, (2) it is well-sized, i.e. the triangle passes a user defined size 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 pass then halt. Otherwise choose the largest triangle, $\triangle$, which fails and determine its circumcenter, c.

4. Traverse from $\triangle$ 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° and 120°. (Smaller angles are required if boundary angles less that 30° are allowed).

2. All triangles will pass the user specified measure.

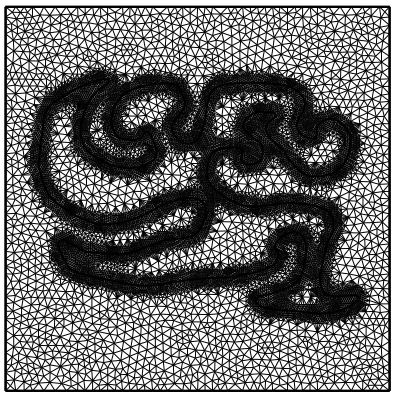
3. All boundary edge constraints will be preserved by the final triangulation.

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



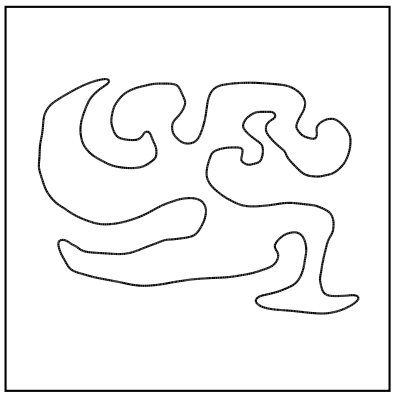
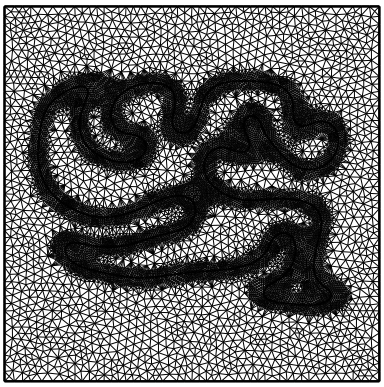


Fig. 21a – b. Initial Adapted Mesh Closed Curve (T=0)



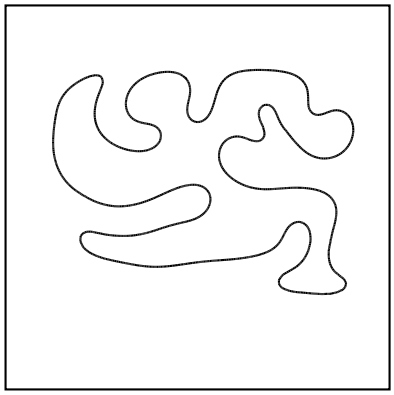
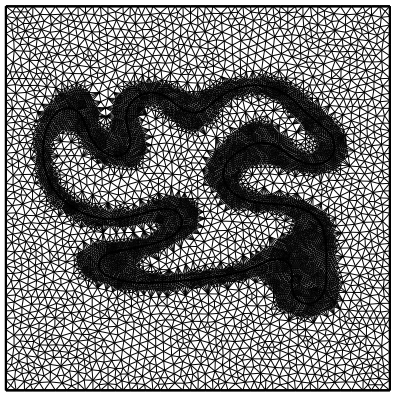


Fig. 21c – d. Adapted Mesh Closed Curve (T=.1)

the context of the level set equation, simulating Grayson's problem amounts

to solving the following level set equation:

$$\phi_t - \kappa|\nabla\phi| = 0. \tag{94}$$



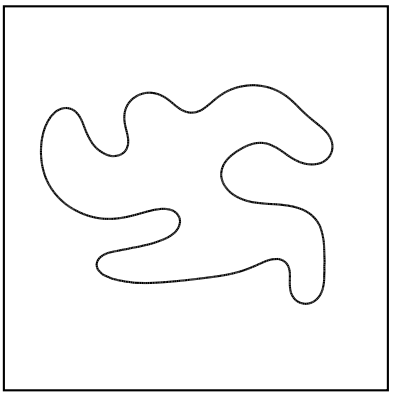
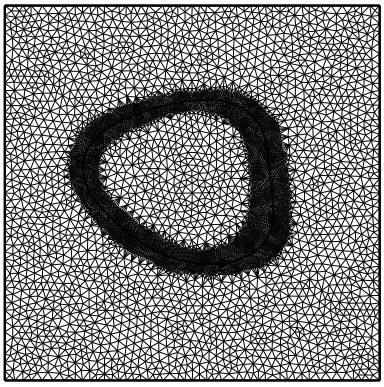


Fig. 21e – f. Adapted Mesh Closed Curve (T=.5)



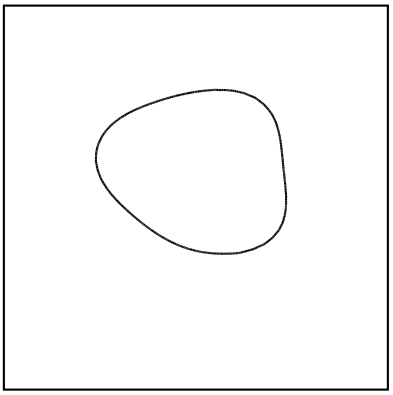


Fig. 21g – h. Adapted Mesh Closed Curve (T=1.5)

Figure 21: Grayson's Problem. Time evolution of simple closed curve with curvature dependent speed.

As Figs. 21a-f 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 κ 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, coupled to a finite difference scheme to advance the level set equation. Chopp restricted the movement of the zero level set so that it always remains 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 function around the zero level set; a renormalization procedure was developed to circumvent this difficulty. The advantage to 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].

Figure 22: Minimal Surfaces Constructed through Curvature Flow

Using a triangular-node based approach to this problem yields the advantage that the triangular discretization can be arranged so that vertices fall on the wire frame, thus automatically constraining the front.

Figure 22 shows the minimal surface that spans two rings, located less than the critical distance apart. By lengthening the distance between the rings, the minimal surface pinches off and separates into two disks spanning the rings. Figure 23 shows the evolution of this sequence, starting from an initial guess and evolving towards the final minimal surface state.

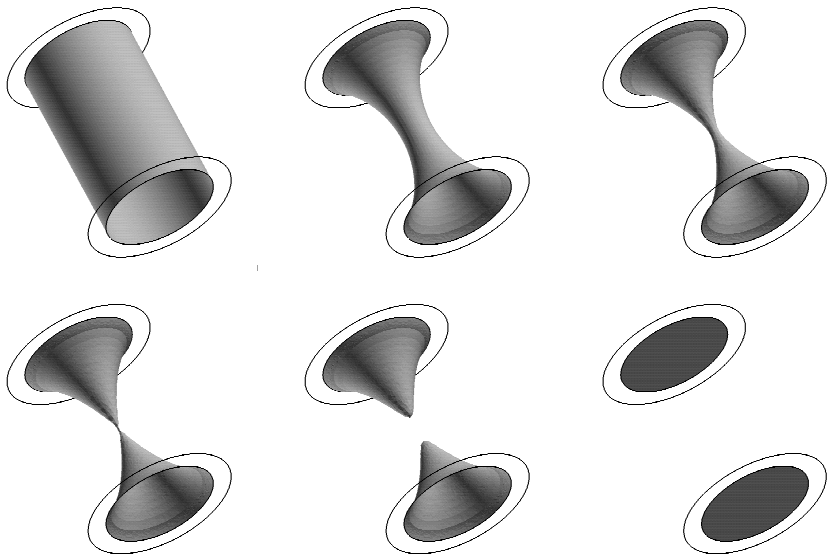


Figure 23: Catenoid evolution through curvature flow undergoing change.

9 Nonconvex Hamiltonians

Next, we study the performance of the newly developed schemes for nonconvex Hamiltonians 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, 4, 5], and the role of nonconvexity in the Hamiltonian was discussed at length. Here, those experiments are repeated in a triangulated setting using the nonconvex scheme discussed earlier.

Figure 24 shows etching profiles for various different yield curves. Note that the last yield curve, Fig. 24d given by $F(\theta) = 5\cos\theta - 4\cos^3\theta$, produces the faceting discussed above.

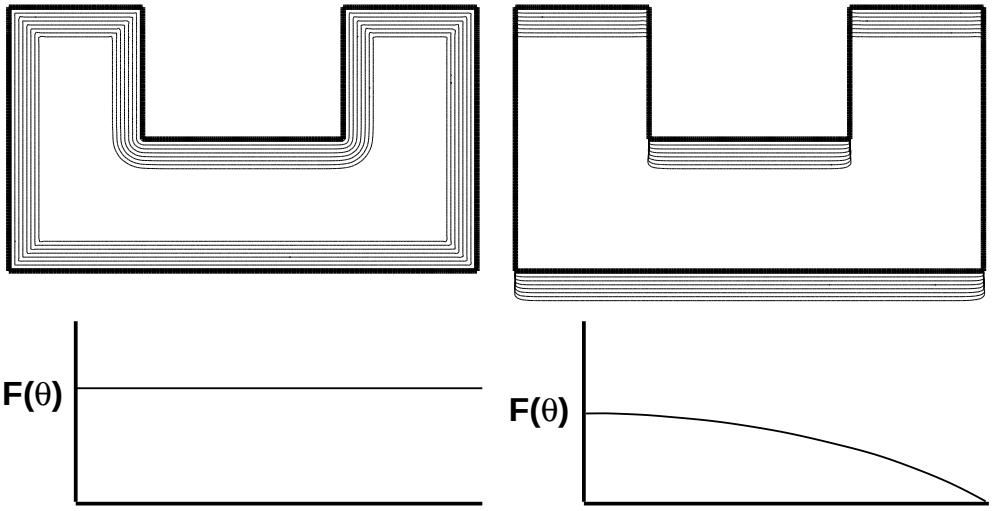


Fig. 24a. $F(\theta) = 1$.

Fig. 24b. $F(\theta) = \cos\theta$.

Figure 25 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

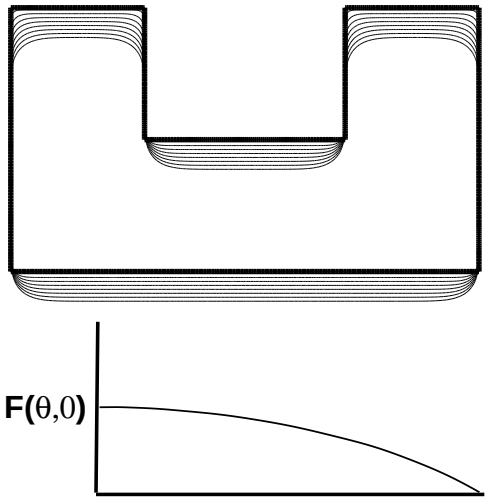


Fig. 24c. $F(\theta, \kappa) = \cos\theta - .1\kappa$.

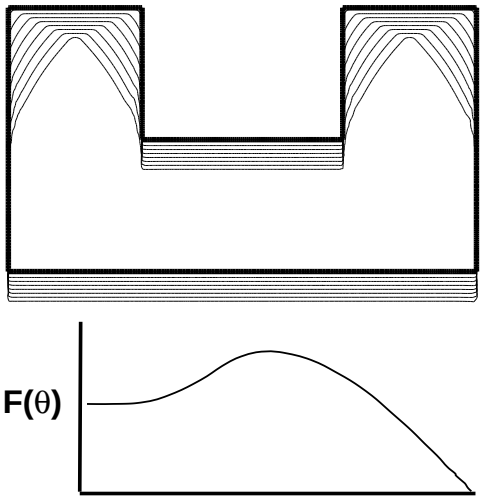


Fig. 24d. $F(\theta) = 5\cos\theta - 4\cos^3\theta$.

Figure 24: Convex and Nonconvex Hamiltonians

faceting on the lower surface. This phenomena is not well-understood and warrants further investigation.

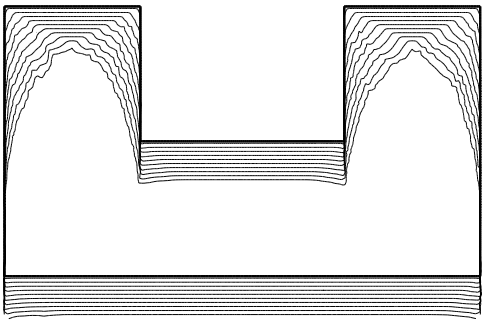


Fig. 25a. Positive Scheme.

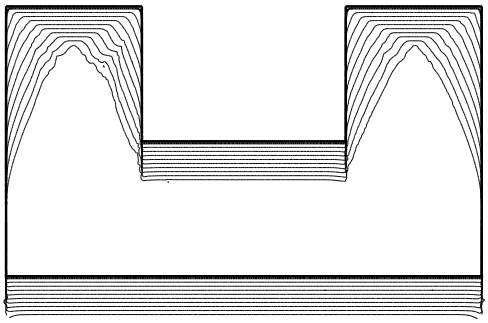


Fig. 25b. Explicit Petrov-Galerkin.

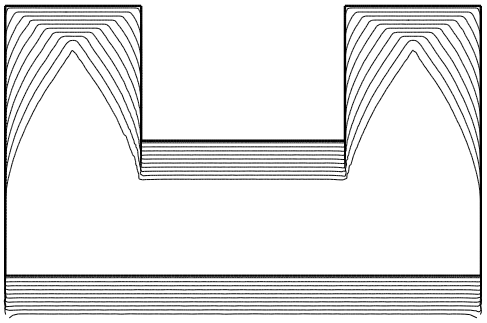


Fig. 25c. Exp. Petrov-Galerkin + D.C.

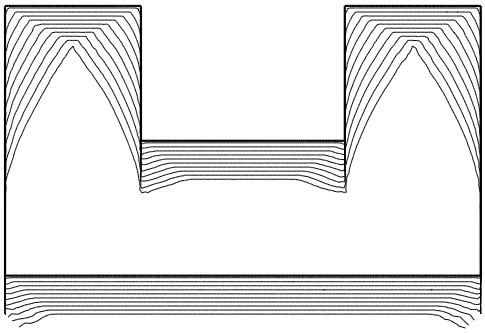


Fig. 25d. Imp. Petrov-Galerkin + D.C.

Figure 25: Performance of various schemes for nonconvex etching problem.

9.1 Etching and Deposition in Semiconductor Manufacturing

In a series of papers [3, 4, 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

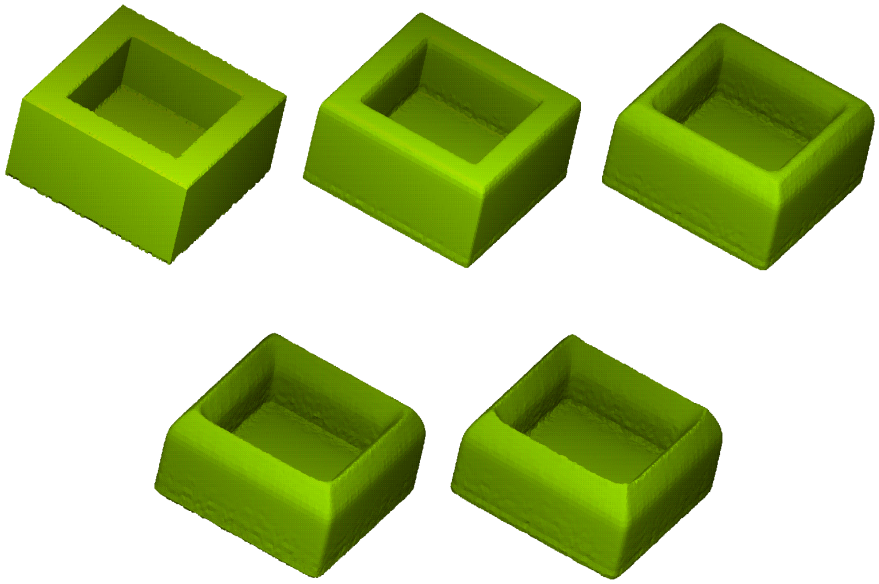


Figure 26: 3D Contact well etching evolution using nonconvex etching law.

figures from left to right and top to bottom show 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.

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.*, 49:1339–1373, 1996.

[2] D. Adalsteinsson and J.A. Sethian. A fast level set method for propagating interfaces. *J. Comp. Phys.*, 118(2):269–277, 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. Comp. Phys.*, 120(1):128–144, 1995.

[4] D. Adalsteinsson and J.A. Sethian. A unified level set approach to etching, deposition and lithography ii: Three-dimensional simulations. *J. Comp. Phys.*, 122(2):348–366, 1995.

[5] D. Adalsteinsson and J.A. Sethian. A unified level set approach to etching, deposition and lithography iii: Complex simulations and multiple effects. *Accepted for publication, J. Comp. Phys.*, 1997.

[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, 1993. Proceedings of the 9th ACM Symposium on Computational Geometry.

[9] D.L. Chopp. omputing minimal surfaces via level set curvature flow. *J. Comp. Phys.*, 106:77–91, 1993.

[10] L. Corrias, M. Falcone, and R. Natalini. Numerical schemes for conservation laws via Hamilton-Jacobi equations. *Math. Comp.*, 64(210):555–580, 1995.

[11] M.G. Crandall and P.L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43(167):1–19, 1984.

[12] M.G. Crandall and L. Tartar. Some relations between nonexpansive and order preserving mappings. *Proceedings of the American Mathematical Society*, 78(3):385–390, 1980.

[13] H. Deconinck, R. Struijs, and P. L. Roe. Compact advection schemes on unstructured grids. Technical report, VKI, 1993. VKI LS 1993-04, Computational Fluid Dynamics.

[14] M. Grayson. The heat equation shrinks embedded plane curves to round points. *J. Diff. Geom.*, 26:555, 1989.

[15] G. Holmes and D. Snyder. The generation of unstructured triangular meshes using Delaunay triangulation. pages 643–652. Pineridge Press, 1988.

[16] Claes Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, Cambridge, 1987.

[17] S.N. Kruzkov. First order quasilinear equations in independent variables. *Mat. Sb.*, 81:228–255, 1970.

[18] P.L. Lions. *Generalized Solutions of Hamilton-Jacobi Equations.* Pitman Publishing Inc., Marshfield, Mass., 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):441–470, 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 Mtg, 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. Comp. Phys.*, 79:12–49, 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.*, 101:487–499, 1985.

[27] J.A. Sethian. Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws. *J. Diff. Geom.*, 31:131–161, 1990.

[28] J.A. Sethian. *Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces.* Acta Numerica, Cambridge University Press, 1995.

[29] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sciences.*, 93(4):1591–1595, 1996.

[30] J.A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Sciences.* Cambridge University Press, Cambridge, 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 non-oscillatory shock-capturing scheme. *Journal of Computational Physics*, 77:439–471, 1988.

[33] H.K Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comp. Phys.*, 127:179–195, 1996.