

Level set methods to compute minimal surfaces in a medium with exclusions (voids)

MONICA TORRES[†]

Department of Mathematics, Northwestern University, Evanston, IL 60208, USA

DAVID CHOPP[‡]

*Department of Engineering Science and Applied Mathematics, Northwestern University,
Evanston, IL 60208, USA*

AND

TIMOTHY WALSH[§]

Sandia National Laboratories, P.O. Box 5800, M.S.0835, Albuquerque, NM 87185, USA

[Received 18 January 2004 and in revised form 11 January 2005]

In [59], periodic minimal surfaces in a medium with exclusions (voids) are constructed and in this paper we present two algorithms for computing these minimal surfaces. The two algorithms use evolution of level sets by mean curvature. The first algorithm solves the governing nonlinear PDE directly and enforces numerically an orthogonality condition that the surfaces satisfy when they meet the boundaries of the exclusions. The second algorithm involves h -adaptive finite element approximations of a linear convection-diffusion equation, which has been shown to linearize the governing nonlinear PDE for weighted mean curvature flow.

1. Introduction

Minimal surfaces occur in many areas of science and engineering. They have been extensively studied and have been a motivation for developing new theories and techniques to analyze them mathematically and to compute them numerically. Soap films, cell membranes and elastic surfaces provide mechanical examples of minimal surfaces [11], [53], [55], [32], [40]. Minimal surfaces have been shown to be important in various chemical microstructures and their corresponding phase transitions [18]. Numerical methods have been developed for these types of problems using level set methods [48], [11], [14], [1], [54]. Computer graphics and image analysis use minimal surfaces frequently for boundary detection, and to construct surfaces that are visually appealing [38], [37], [15], [56]. A recent tool to model minimal surfaces is *The Surface Evolver* [6], [7]. The *Evolver* has been applied in several domains such as capillary surfaces [41], material science [45], [34] and many others [44], [28], [8], [9], [58].

This work is motivated by [59], where periodic minimal surfaces of least area *in media with exclusions (voids)* are constructed. In each period, these surfaces provide an absolute minimum in terms of area among all surfaces with periodic boundary conditions. Even though it is possible to

[†] E-mail: torres@math.northwestern.edu

[‡] E-mail: chopp@northwestern.edu

[§] E-mail: tfwalsh@sandia.gov

prove mathematically that such surfaces exist, to the knowledge of the authors, it is not known in general how to compute numerically (guaranteed) minimal surfaces that provide an absolute (or even local) minimum in terms of area. However, we know that the surfaces in [59] satisfy the following properties: (a) they have zero mean curvature outside the exclusions and, as will be explained in the next section, (b) they must enter the exclusions *orthogonally* (see [59], [36]). Therefore, we present in this paper two numerical algorithms that reproduce these two features. In general, it has become customary to use the term **minimal surface** to denote surfaces of vanishing mean curvature (see [42]), even though such surfaces often do not provide a minimum (absolute or even local) for the surface area. We note that, although theoretically it may be possible for our algorithms to evolve to an unstable minimal surface, numerically the probability of this happening is almost zero. For an example of the difficulties in finding, numerically, unstable equilibrium surfaces see [12]. Therefore, in this paper we will not attempt to prove the stability or instability of the computed surfaces. Instead, we will focus on the development of algorithms that satisfy the two properties discussed above.

In particular, we seek answers to several interesting questions related to the minimal surfaces in media with exclusions:

1. How does one adapt standard level set methods to handle the presence of exclusions?
2. Do the exclusions act as *attractors* for the minimal surfaces?
3. Do the numerical results produce minimal surfaces that enter the exclusions orthogonally?

The first issue encompasses the novelty of this paper. The second issue is simply a conjecture. For the third issue the intent is to develop algorithms that reproduce this feature, since the orthogonality property has already been proven mathematically ([31], [36]). We present in this paper two different algorithms, which can compute the desired surfaces, that use **evolution by mean curvature**.

2. Minimal surfaces in media with exclusions and evolution according to mean curvature

The recent results in [10] concern a generalization of the problem of minimal surfaces in periodic media and show that, given a metric with periodic coefficients, there exists a number M so that one can find a minimizer in any strip of width M . The width M is independent of the orientation of the strip. Moreover, the minimizers constructed in [10] have the property that, when folded to the fundamental domain, they are laminations.

In [59], the results of [10] are extended to a situation where the medium has **exclusions**, i.e., **regions for which the metric vanishes**. In both papers, the surfaces are considered as boundaries of sets and the perimeter is defined in a weak sense (see [27]). This approach is advantageous because the fundamental domain in [59] is a manifold with boundary and the theory of homologically minimizing currents in manifolds with boundary is not readily available to our knowledge. In [59], the existence of plane-like periodic minimal surfaces in a medium with exclusions (voids) is proven. Similar situations of media with exclusions are standard in the theory of homogenization.

More precisely, we consider the following problem. We work in \mathbb{R}^n ($n = 2, 3$) with exclusions, which can be thought of as holes or voids, and are compact sets with Lipschitz boundary. Considering a parallelepiped embedded in \mathbb{R}^n , we look at all the surfaces with periodic boundary lying on the lateral faces of the parallelepiped. The area of a surface is measured by neglecting the parts that are inside the exclusions. We search for the surface with least area measured in this way. Figure 1 gives an illustration of the physical situation, in two dimensions. In this case, two candidate

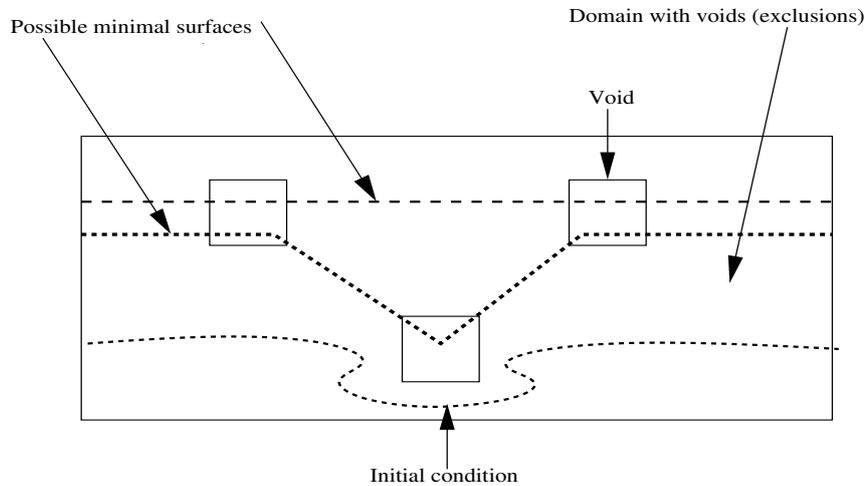


FIG. 1. Diagram showing the physical setup of the problem. Two surfaces with zero mean curvature (outside the exclusions) are drawn, going from the left side to the right side of the domain, along with an arbitrary initial condition.

minimal surfaces are drawn in a medium with exclusions, along with an arbitrary initial condition. More will be given later on the issue of initial conditions.

The existence of this minimal surface is proven in [59]. The degeneracy of the metric inside the exclusions increases the difficulty of the minimal surface problem, both analytically and numerically, and in this degeneracy resides the novelty of the results presented in [59]. The surface of least area constructed in [59] is not unique but it is a global minimum (in terms of surface area) among all candidate surfaces. As explained in the introduction, we compute in this paper surfaces with zero mean curvature outside the exclusions that enter the exclusions *orthogonally*. We note, however, that the algorithms cannot guarantee a global minimum or even local minimum in terms of area.

The two algorithms presented in this paper use evolution of level sets according to mean curvature. The basic principle behind these algorithms can be explained as follows. If we consider \mathbb{R}^2 with no exclusions, then the smallest distance between two points is a line. This line could be approximated by evolving any curve joining the two points in such a way that any location on the curve moves in the normal direction with velocity proportional to the curvature at that location. Then a point with more curvature will move faster than a point with less curvature. If we continue this evolution over time, the result is a curve having zero mean curvature, i.e., a line.

When the exclusions are present, the shortest distance between two points is no longer a straight line, and one can check easily that the optimal path meets the exclusions orthogonally. These two facts greatly complicate the development of numerical algorithms for computing the surfaces. The same orthogonality phenomena occur in higher dimensions; i.e., the intersection of the minimal surface and the exclusions locally looks like two perpendicular hyperplanes. A proof of this fact, using the theory of currents, can be found in [31]. We can also obtain the orthogonality property by studying the first variation of the area. This analysis is done in [36], where a generalized Snell's law is derived, for the case of two different media with nonzero weight. In [36], the surfaces are treated as graphs of functions while in this paper we use a level set approach, since the surfaces we

want to compute are boundaries of sets, and not necessarily graphs of functions. Moreover, the first algorithm we present enforces numerically the orthogonality condition.

One important issue with regard to exclusions is the smoothness of their boundaries. The theoretical basis of the algorithms presented in this paper only covers the case of exclusions with C^2 boundaries. This, of course, does not allow for exclusions with corners or edges, which are quite common. In some of the numerical experiments, however, we study the case of exclusions with corners, although we recognize that the algorithms are not guaranteed to behave as expected in these cases. Interestingly, in the numerical results we consistently observe the computed minimal surfaces entering at 45° at corners that involve right angles.

The evolution of hypersurfaces in \mathbb{R}^n according to their mean curvature has been investigated thoroughly in the literature. There are many results in this area using parametric methods of differential geometry (see e.g. [30], [29], [33], and the references therein), in the setting of varifold theory from geometric measure theory (see e.g. [6], and the references therein), and through the level set representation (see e.g. [43], [48], [24], [39], and the references therein).

A level set approach for motion by mean curvature was given by Osher and Sethian in [43]. Given the initial hypersurface Γ_0 , they select some continuous function g so that

$$\Gamma_0 = \{x \in \mathbb{R}^n : g(x) = 0\}.$$

They show that the PDE

$$\begin{aligned} u_t &= |\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) && \text{in } \mathbb{R}^n \times [0, \infty), \\ u &= g && \text{on } \mathbb{R}^n \times \{t = 0\} \end{aligned} \quad (1)$$

evolves the level sets of u according to their mean curvature.

This equation can be extended to the case with exclusions as follows:

$$\begin{aligned} u_t &= |\nabla u| \operatorname{div} \left(w \frac{\nabla u}{|\nabla u|} \right) && \text{in } \mathbb{R}^n \times [0, \infty), \\ u &= g && \text{on } \mathbb{R}^n \times \{t = 0\}, \end{aligned} \quad (2)$$

where w is a smooth weight that takes value $\epsilon \ll 1$ inside the exclusions and 1 outside. In this paper, we present two algorithms, which approximate the PDE's (1) and (2) respectively.

In the first algorithm, we solve the nonlinear equation (1) outside the exclusions, while imposing the condition that the surface must be orthogonal to the exclusions. This algorithm uses a finite difference scheme similar to that used in [11].

In the second approach, we solve equation (2) by adapting the BMO (Bence, Merriman and Osher) algorithm to the case with exclusions since the original BMO algorithm, as introduced in [39], does *not* directly carry over to the case with exclusions. The original BMO uses a linearization of equation (1), which leads to the heat equation. However, in the case with exclusions we must instead linearize equation (2), and this leads to a convection term in addition to the standard diffusion term. This complicates the numerical implementation, but still leads to an algorithm that is similar to BMO. Our implemented algorithm uses h -adaptive finite element approximations to solve the linear convection-diffusion equation that linearizes the governing nonlinear PDE (2).

There are three primary differences between the two algorithms. First, in the first algorithm, mean curvature is computed explicitly for purposes of evolving the surface, while the second algorithm uses a linearized approximation of the mean curvature. Second, the first algorithm

enforces an orthogonality condition at the exclusion surfaces, while the second algorithm uses an additional weight function which produces the same orthogonality condition at equilibrium. Third, the first algorithm evolves the surface only outside the exclusions, while the second algorithm evolves the surface both inside and outside the exclusions.

3. The first algorithm

Following the original work on the level set method by Osher and Sethian [43], a numerical study of mean curvature flow using the level set method was conducted in [14]. Minimal surfaces attached to fixed one-dimensional curves were also computed using the level set method in [11]. The algorithm used in this paper is a variation of those earlier works.

For simple curvature flow using the level set method, the PDE (1) is rewritten for two dimensions as

$$u_t = \frac{u_{xx}u_y^2 + u_{yy}u_x^2 - 2u_{xy}u_xu_y}{u_x^2 + u_y^2}, \quad (3)$$

for three dimensions as

$$u_t = \frac{u_{xx}(u_y^2 + u_z^2) + u_{yy}(u_x^2 + u_z^2) + u_{zz}(u_x^2 + u_y^2) - 2u_{xy}u_xu_y - 2u_{xz}u_xu_z - 2u_{yz}u_yu_z}{u_x^2 + u_y^2 + u_z^2}, \quad (4)$$

and solved using standard central differences. The details of the discretization can be found in [14].

3.1 Implementation of the orthogonality condition

The introduction of exclusions, where additional boundary conditions are imposed on the evolving surface, requires a modification to the simple flow. To implement the boundary conditions, the exclusions are also represented using a level set function $\omega(x)$, where $\omega(x)$ is the signed distance to the boundary of the exclusions with $\omega(x) > 0$ inside, and $\omega(x) < 0$ outside.

By representing the exclusions using a signed distance function, imposing the orthogonality condition on the boundary of the exclusion becomes simpler. The normal to the exclusion is now given by $\nabla\omega$, while the normal to the evolving surface is ∇u . Thus, the orthogonality condition is enforced by requiring

$$\nabla\omega \cdot \nabla u = 0 \quad \text{on } \{x \in \mathbb{R}^n : \omega(x) = 0\} \quad (5)$$

at the exclusion surface.

Interestingly, (5) is the same equation used to construct velocity extensions in modern level set methods as described in [3]. In that context, a speed function F , defined on the surface $v = 0$, is to be extended off the surface orthogonally. This results in solving for F the equation

$$\begin{aligned} \nabla F \cdot \nabla v &= 0 & \text{in } \mathbb{R}^n, \\ F &= F_0 & \text{on } \{x \in \mathbb{R}^n : v(x, t) = 0\}, \end{aligned} \quad (6)$$

where F_0 is given.

In [3], an efficient algorithm for solving (6) is presented, which is based on the fast marching method [47], [46], [49], [50], [51], [52]. The fast marching method is an optimally fast algorithm for solving equations of the form

$$\begin{aligned} G|\nabla v| &= 1 & \text{in } \mathbb{R}^n, \\ v &= 0 & \text{on } \Gamma, \end{aligned} \quad (7)$$

where G is a prescribed scalar speed function defined in \mathbb{R}^n and Γ is a manifold in \mathbb{R}^n with co-dimension one. Within the context of level set methods, this algorithm was modified to allow higher accuracy in [13]. The fast marching method uses a principle of causality to compute the solution to (7), using upwind finite difference approximations, in order from nearest to the surface Γ to the farthest. The precise details of the implementation of the fast marching method are beyond the scope of this paper; the interested reader is referred to [46], [13].

Velocity extensions are computed by first solving (7) with $G \equiv 1$, $\Gamma = \{x : u(x) = 0\}$ and then solving (6) in the same order as used in solving (7). Note that in practice, (6) and (7) are solved simultaneously.

Returning to the algorithm for this paper, the function ω representing the exclusions is constructed initially so that it solves (7), with $\Gamma = \{x : \omega(x) = 0\}$. Thus, all that remains is to solve (6) with $v = \omega$, $F = u$, and $F_0 = u|_{\{x : \omega(x)=0\}}$, only *inside* the exclusions, i.e. where $\omega(x) > 0$. The resulting function u is unchanged outside the exclusions and satisfies (5) inside the exclusions.

3.2 Reinitialization

As observed in [11], the introduction of constraints to curvature flow can lead to instability in the numerical method. Without taking additional measures, the same would be true here. The remedy employed in [11] was to use reinitialization.

In the presence of constraints, the unmodified method will not maintain even spacing of the level sets. This leads to both steep gradients in some regions and near zero gradients over other larger regions, and both can cause a severe breakdown in the numerical approximation. Reinitialization is a process where the level set spacing is restored by solving

$$\begin{aligned} |\nabla u^*| &= 1 && \text{in } \mathbb{R}^n, \\ u^* &= 0 && \text{on } \{x \in \mathbb{R}^n : u(x) = 0\}. \end{aligned} \quad (8)$$

The solution u^* of (8) is a signed distance function to the set $\{x : u(x) = 0\}$, and so has even spacing of level sets restored while keeping the surface of interest, $\{x : u(x) = 0\}$, unchanged.

Obviously, (8) is solved in the same manner as (7), with $G = 1$ and $\Gamma = \{x : u(x) = 0\}$. However, unlike the implementation of the boundary conditions, this time we solve (8) over all \mathbb{R}^n , not just inside the exclusions. Since $\{x : u^*(x) = 0\} = \{x : u(x) = 0\}$, using reinitialization after applying the boundary conditions means that the surface $\{x : u^*(x) = 0\}$ is still orthogonal to the exclusions, but other level curves of u^* may not be.

3.3 Summary of the first algorithm

The pieces described above are now assembled into the following algorithm:

1. Initialization: Construct initial signed distance function to the exclusions, $\omega(x)$, and initial surface to be minimized, u . Here, $\omega(x) > 0$ inside the exclusions and $\omega(x) < 0$ outside. The function u is the signed distance to the initial curve/surface.
2. Move: Advance the function u one time step using (3) or (4). Here, u is advanced independent of the exclusions since the value of u in the exclusions will be recomputed in the following step.

3. Enforce Boundary Conditions: Solve (5) in the set $\{x \in \mathbb{R}^n : \omega(x) \geq 0\}$ as described in Section 3.1. This step will reset the value of u inside the exclusions, while leaving everything outside the exclusions unchanged. The function u is no longer a signed distance function after this step.
4. Reinitialize: Solve (8) on all \mathbb{R}^n and then replace $u = u^*$. This step changes u back into a signed distance function everywhere so that it can stably evolve by mean curvature.
5. Loop: Go to step 2.

4. The second algorithm

Bence, Merriman and Osher proposed in [39] a numerical algorithm for computing the mean curvature flow (1) using the heat equation and reinitializing after short time steps. This algorithm, which is a quasi-linearization of (1), proceeds as follows. The nonlinear PDE (1) is replaced by the heat equation:

$$\begin{aligned} u_t - \Delta u &= 0 && \text{in } \mathbb{R}^n \times (0, \infty), \\ u &= \varphi_{C_0} && \text{on } \mathbb{R}^n \times \{t = 0\}, \end{aligned} \quad (9)$$

where C_0 is an open bounded set with boundary Γ_0 and φ_{C_0} is the characteristic function of C_0 . The linear PDE (9) is solved to some small time t . Then the new set is defined:

$$C_t = \{x \in \mathbb{R}^n : u(x, t) \geq 1/2\} \quad (10)$$

and (9) is solved again with C_0 replaced by C_t . In [39], heuristics are given to show that the evolution of C_0 into C_t approximates, for small times, the mean curvature motion of the boundary Γ_0 of C_0 . By repeating this procedure, i.e. solving the heat equation and reinitializing after a short time, approximate mean curvature flow is obtained. This is valid even for large times. The algorithm only involves the heat equation, which is easy to implement numerically and very inexpensive to compute. This allows for long-time simulations with many time steps, at minimal computational cost. Proofs of the convergence of this algorithm can be found in Evans [19] and Barles–Georgelin [4]. We could say that the original BMO algorithm [39] simulates mean curvature flow in homogeneous domains, as can be seen by the constant coefficients in the PDE (9). In this paper we extend this to domains with exclusions. This makes the computations much more difficult in many ways, especially due to the sharp change of material properties near the boundary of the exclusions. This makes local adaptivity an attractive tool for this problem.

We now describe an algorithm to compute the minimizers, in the case of a domain with exclusions, and we recall that the area is measured by putting a weight of 0 inside the exclusions and 1 outside the exclusions. For the purposes of numerical computations, it is not possible to handle the case when the coefficients of the PDE are 0 at some point in the domain, since the elliptic term in the bilinear form would disappear. To avoid this, we use a smooth weight $\epsilon \leq w \leq 1$, with $\epsilon \ll 1$, where w has value ϵ inside the exclusions and increases smoothly to 1 on the outside. Since w is smooth, the existence of minimizers is contained in [10], as well as a discussion on the regularity of minimizers. Since mean curvature is a local property, we can restrict ourselves to an open set $\Omega \subset \mathbb{R}^{n-1}$, where the minimal surface is the graph of the smooth function $f(x')$, $x' \in \mathbb{R}^{n-1}$. The function f minimizes

$$\int_{\Omega} w(x', f(x')) \sqrt{1 + |\nabla f(x')|^2} \quad (11)$$

over all smooth functions agreeing with f on $\partial\Omega$. Thus f satisfies the Euler–Lagrange equation

$$\operatorname{div} \left(w(x', f(x')) \frac{\nabla f(x')}{\sqrt{1 + |\nabla f(x')|^2}} \right) - w_{y_n}(x', f(x')) \sqrt{1 + |\nabla f(x')|^2} = 0. \quad (12)$$

The first variation (12) of the functional (11) gives a formula for the weighted mean curvature at the point $(x', f(x'))$. We will see later that, in order to apply the level set approach, it is necessary to find the analogous equation to (12) for the case of a surface defined implicitly. The main idea in the level set approach is to embed the initial surface as the zero level set of a continuous function, and find the PDE that evolves this surface according to a prescribed velocity. This adds one dimension to the problem. Each point in the surface will move in the normal direction with a velocity $-H_w$, where $H_w(x)$ is the weighted mean curvature at each point x of the surface.

Suppose we have $\Gamma_0 = \{x \in \mathbb{R}^n : g(x) = 0\}$, where g is a smooth function, and we want to evolve Γ_0 according to its weighted mean curvature. Assume that at each time $t \geq 0$, the evolved hypersurface is given implicitly by the equation $u(x, t) = 0$. Then $u(x, 0) = g(x)$. Let $x \in \mathbb{R}^n$ and let $\alpha(s), \alpha(t) = x$, be the trajectory of x under this evolution. We have $u(\alpha(s), s) = 0$ for all s . Differentiating with respect to s gives $u_t(\alpha(s), s) + \nabla u(\alpha(s), s) \cdot \alpha'(s) = 0$; that is, $u_t(x, t) + \nabla u(x, t) \cdot \alpha'(t) = 0$. However,

$$\alpha'(t) = -H_w(x) \frac{\nabla u}{|\nabla u|}. \quad (13)$$

Since $\nabla u/|\nabla u|$ is the exterior unit normal vector at x , equation (13) states that the level sets of u are evolving in the normal direction with velocity $-H_w(x)$. Since locally the level sets of u are the graph of a function, we can use (12) to prove (after a somewhat lengthy computation) that

$$H_w(x) = \operatorname{div} \left(w(x) \frac{\nabla u}{|\nabla u|} \right). \quad (14)$$

Therefore

$$u_t(x, t) = \nabla u \cdot H_w(x) \frac{\nabla u}{|\nabla u|} = |\nabla u| H_w(x) = |\nabla u| \operatorname{div} \left(w(x) \frac{\nabla u}{|\nabla u|} \right).$$

Then the PDE that evolves Γ_0 according to its weighted mean curvature is

$$\begin{aligned} u_t(x, t) &= |\nabla u| \operatorname{div} \left(w(x) \frac{\nabla u}{|\nabla u|} \right), \\ u(x, 0) &= g(x). \end{aligned} \quad (15)$$

We consider a linearization of equation (15) analogous to the linearization of equation (1) by (9). That is, we want to find a BMO version for equation (15). This is in fact possible and the equation

$$\begin{aligned} u_t(x, t) &= \operatorname{div}(w(x)\nabla u) + \frac{1}{2}\nabla w \cdot \nabla u, \\ u(x, 0) &= g(x), \end{aligned} \quad (16)$$

when reinitialized after small times, converges to equation (15). More specifically, if we let Ω_0 be an open set such that $\partial\Omega_0 = \Gamma_0$, then the following scheme:

- $u_{\Delta t}(x, 0) = \varphi_{\Omega_0} - \varphi_{\Omega_0^c}$.
- Solve:
 - $v_t = \operatorname{div}(w(x)\nabla v) + \frac{1}{2}\nabla w \cdot \nabla v$ in $\mathbb{R}^n \times (0, \Delta t]$,
 - $v(x, 0) = u_{\Delta t}(x, (n-1)\Delta t)$;

- $u_{\Delta t}(x, n\Delta t) = 1$ if $v(x, \Delta t) > 0$,
 $u_{\Delta t}(x, n\Delta t) = -1$ if $v(x, \Delta t) \leq 0$,

converges, as $\Delta t \rightarrow 0$, to motion by weighted mean curvature. Note that this algorithm contains not only the standard *diffusion* term $\text{div}(w(x)\nabla u)$, but also the *convective* term $\frac{1}{2}\nabla w \cdot \nabla u$.

A proof of the convergence of this linearization to the nonlinear equation (15) can be found in [35].

4.1 Numerical implementation

We now describe the implementation of the BMO algorithm corresponding to motion by weighted mean curvature. This amounts to solving equation (16) in a parallelepiped using Dirichlet boundary conditions $u(x, t) = 1$ and $u(x, t) = 0$ on the bottom and the top of the parallelepiped respectively, and periodic boundary conditions on the sides. Our initial condition is the characteristic function of a set bounded by a curve (see Figure 2). We solve equation (16) for a small time and then reinitialize, in the same way as for equation (9). The algorithm can be summarized as follows:

1. Let $u_0 = \varphi_C$.
2. Solve:

$$u_t(x, t) = \text{div}(w(x)\nabla u(x, t)) + \frac{1}{2}\nabla w \cdot \nabla u,$$

$$u(x, 0) = u_0.$$

3. Reinitialize at $t = T_{\text{chop}}$:
 if $u(x_i) \geq 0.5$ then set $u_0(x_i) = 1$ else set $u_0(x_i) = 0$.
4. Repeat step 2.
 The location of the front is given by the level set $\{u = 1/2\}$.

The algorithm has been implemented in two dimensions by modifying the Fortran 90 code 2Dhp90 (see [17], [16]). This code solves the equation $D_j(a_{ij}u_i) = 0$ in two dimensions, with built-in

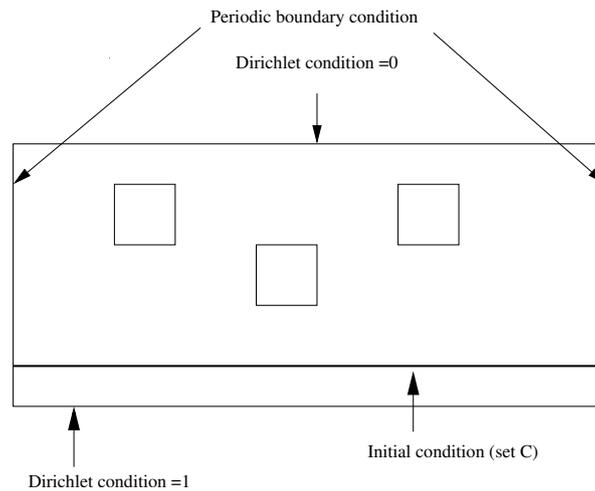


FIG. 2. Diagram showing the domain with exclusions where we are solving equation (16).

capability for *local* adaptivity. Local adaptivity is very important for resolving the sharp gradients along the interfaces of the exclusions. The implementation involved modifying the 2Dhp90 code with the convection term, the weighting function, and the periodic boundary conditions. Although 2Dhp90 has the capability for higher order polynomial basis functions, only bi-linear basis functions were used in the computations. The time discretization was accomplished using a backward Euler scheme, which is first-order accurate.

5. Results

5.1 Results from the first algorithm

To demonstrate the first algorithm, which enforces an orthogonality condition at the surface/void interface, we show, in Figures 3, 4, the evolution of a sample curve from an initial sine curve to equilibrium. Note how the parts of the curve which have both ends intersect the same void form

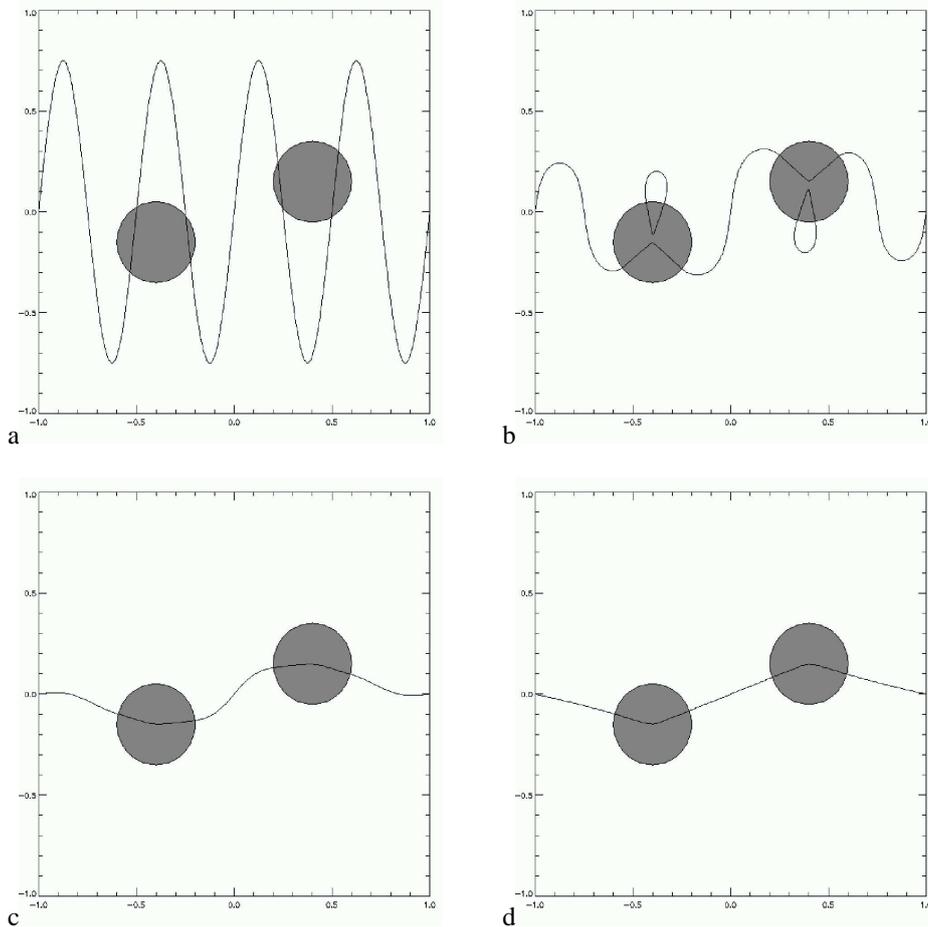


FIG. 3. Sample evolution with two disk-shaped voids using the first algorithm.

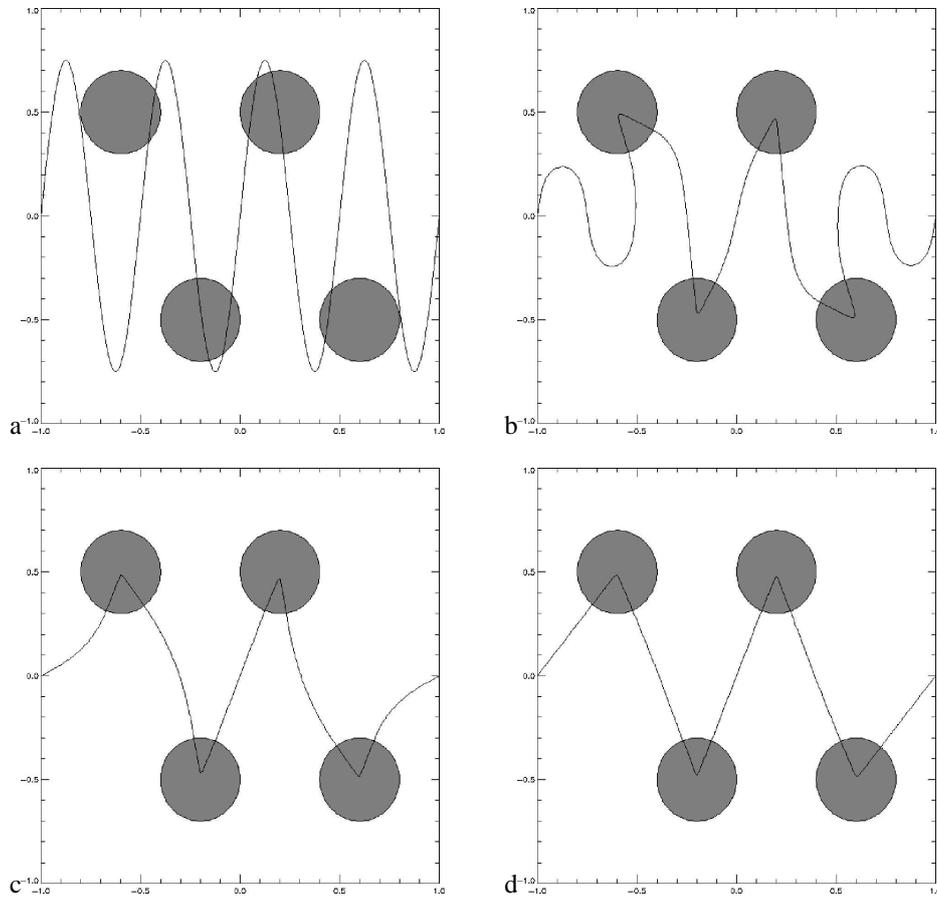


FIG. 4. Sample evolution with four disk-shaped voids using the first algorithm.

loops which eventually vanish, leaving only a straight line segment solution. In both cases, the computation was done on a uniform 200×200 mesh, periodic in the x -direction. In these particular examples, it can be seen by inspection that the final curves are local minima. However, as noted in the introduction, in general we cannot prove that our algorithms will always converge to a global or local minima.

By specifying the voids as a signed distance function, it is straightforward to make voids of different shapes, such as squares as shown in Figure 5, or a combination of a square and a parallelogram as in Figure 6. We note that the final curve obtained in Figure 5 is actually a global minimum. This is not guaranteed to always happen, but sometimes the algorithms will actually converge to the global minimum, as in this case.

We also note that it is possible to construct examples which require significant time to converge to equilibrium, such as in Figure 6. Here, the normal boundary condition creates a slight curvature between the square and the parallelogram, but the curvature can be very small. This leads to very slow convergence to the minimum, where the curve passes through the corners of the voids that

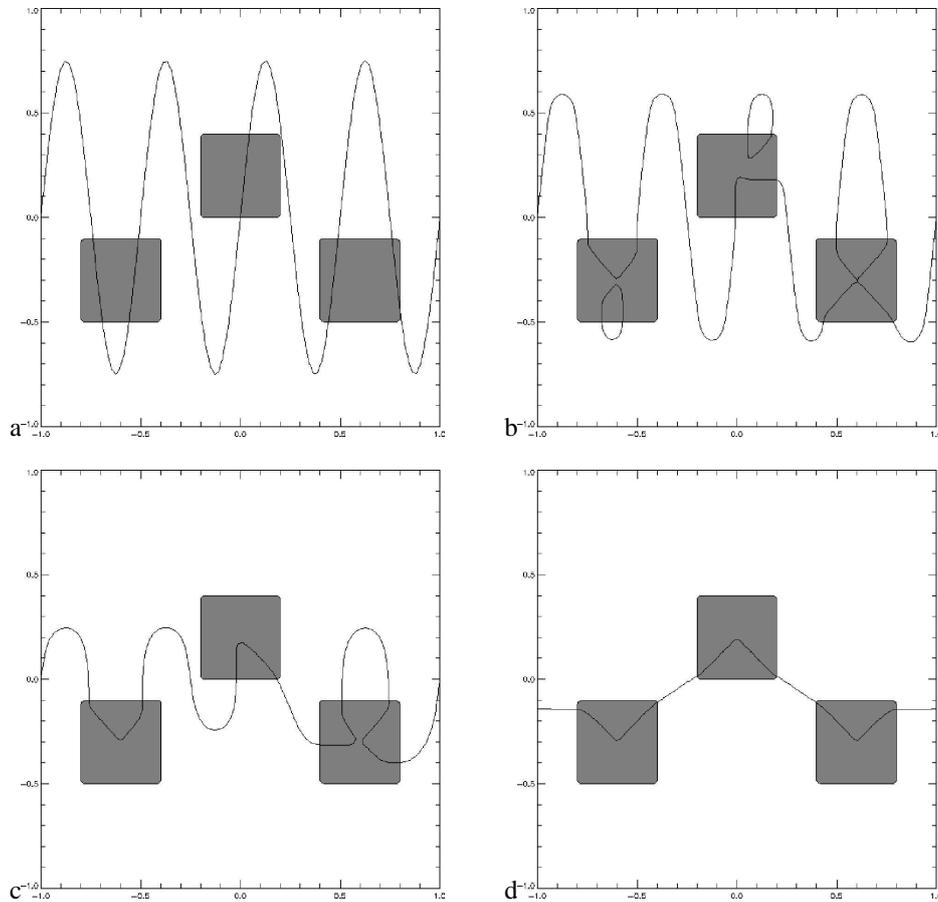


FIG. 5. Sample evolution with three square voids using the first algorithm.

are nearest to each other. Nonetheless, given sufficient time, the algorithm does gradually attain the proper minimal curve.

To show that the method is convergent, we take an example of four disks with an initial sinusoidal curve passing through them. In Figure 7b, we plot the location of the evolving curve at a common intermediate time, and at the final time for mesh sizes of 50×50 , 100×100 , 200×200 , and 400×400 . The times to compute the final equilibrium solution for the different mesh sizes are listed in Table 1. Of course, as noted above, the time required to reach equilibrium is highly dependent on the configuration of the exclusions.

The algorithm extends to two-dimensional surfaces in \mathbb{R}^3 easily. For each of the three-dimensional cases, a uniform mesh of size $80 \times 80 \times 80$ was used, periodic in the x - and y -directions. In Figure 8, we show a sample evolution of a surface which passes through three spherical voids. In this case it is not obvious whether the final surface is a global minimum or a local minimum.

In general, a mathematical analysis in three dimensions to determine if the final surface is a global minimum or a local minimum is difficult. In two dimensions, however, a mathematical

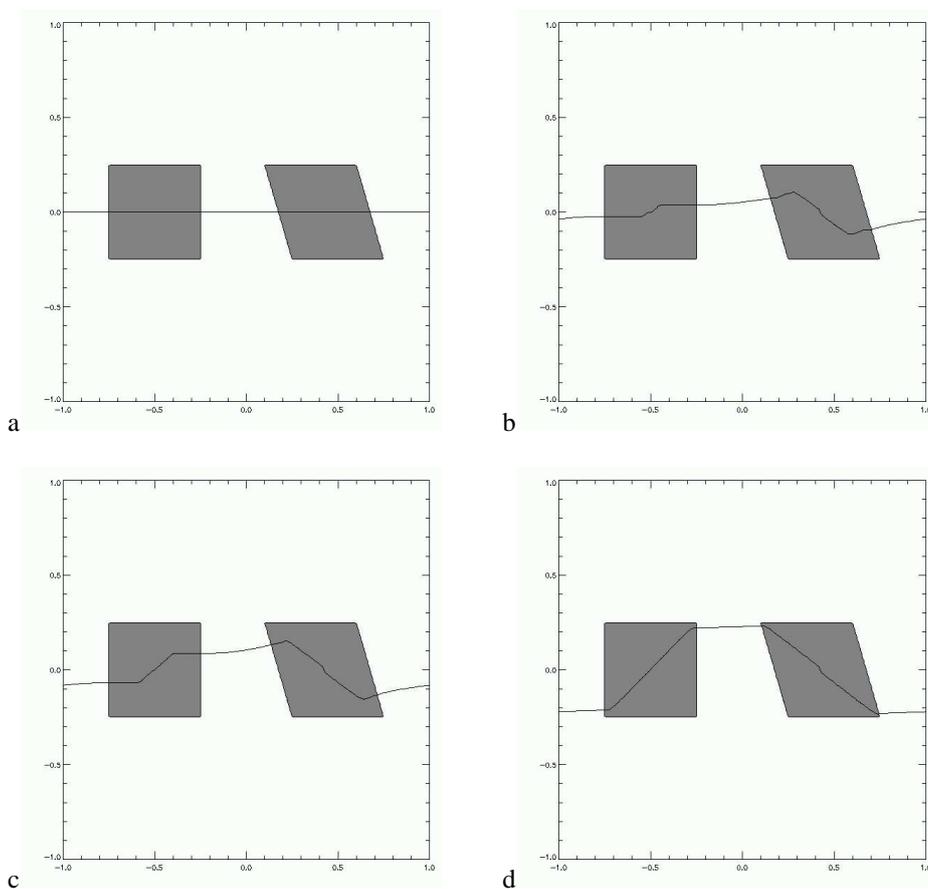


FIG. 6. Sample evolution with a square and a parallelogram void using the first algorithm.

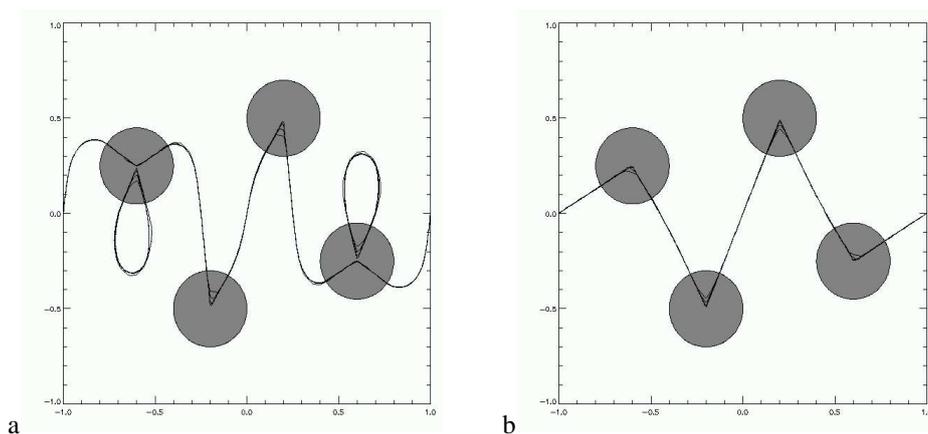


FIG. 7. Convergence test for the first algorithm at (a) an intermediate time, and (b) terminal time.

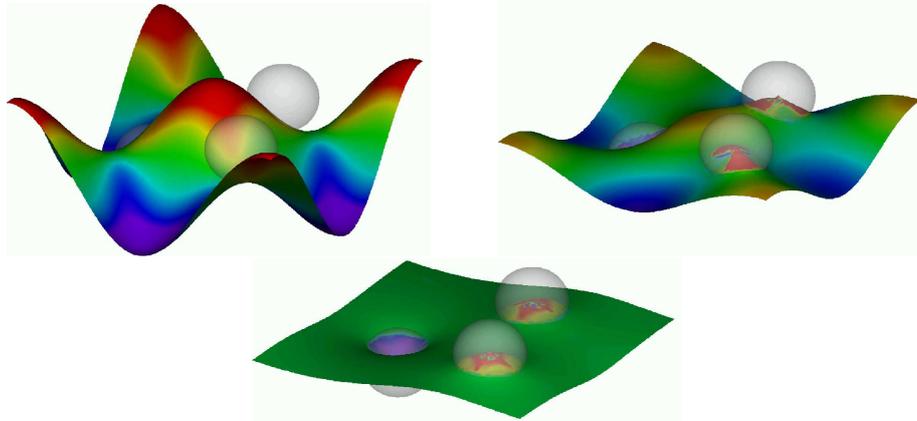


FIG. 8. Sample evolution with three spherical voids using the first algorithm. Color (see the pdf file of the article) indicates positive (red) and negative (blue) curvature regions.

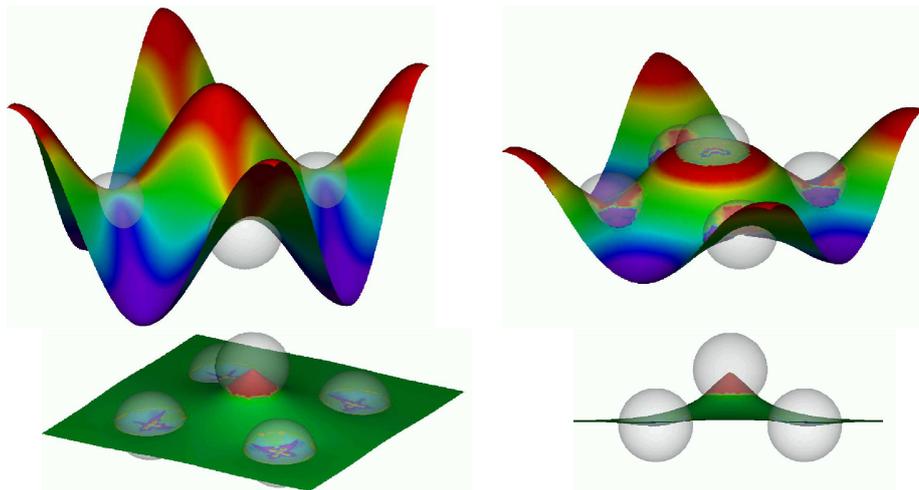


FIG. 9. Sample evolution with five spherical voids, four coplanar, and one center void higher. Two different views are shown for the final minimal surface.

analysis of the optimal path is much more accessible. We refer to [59] for an analysis of the optimal path in two dimensions (assuming that the exclusions are periodic closed balls) as well as its relation with the theory of homogenization of Hamilton–Jacobi equations.

In Figures 9, 10, the same initial surface is used to find the minimal surface through five spherical voids. Four voids are coplanar, while the fifth is in the center and above the others. Figure 9 shows that if the middle void is low enough, a minimal surface can be formed which passes through the middle void. On the other hand, if the middle void is raised too high, as in Figure 10, the same initial surface pulls away from the center void to form a single flat plane solution. This is consistent with the fact that if the middle void is too high, it is not optimal for the surface to travel the extra distance to pass through the middle void.

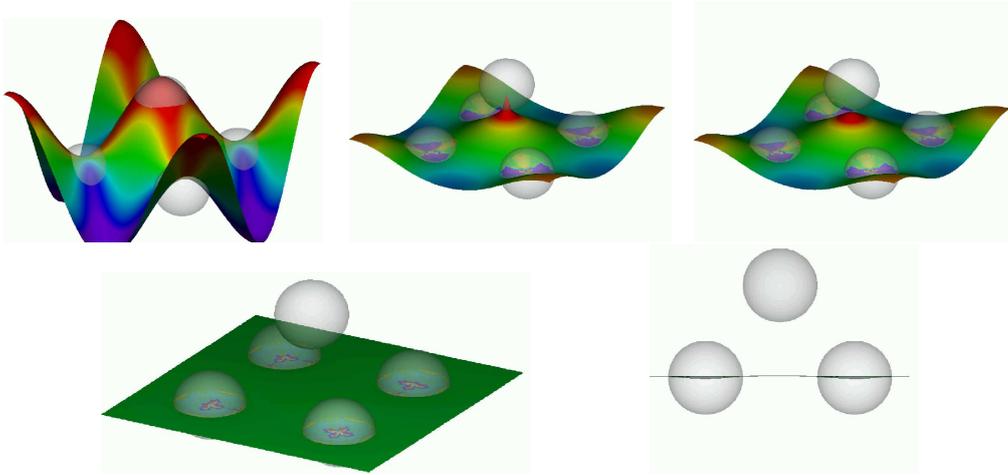


FIG. 10. Sample evolution with five spherical voids, four coplanar, and one center void higher. Two different views are shown for the final minimal surface. Compared with Figure 9, the center void is higher.

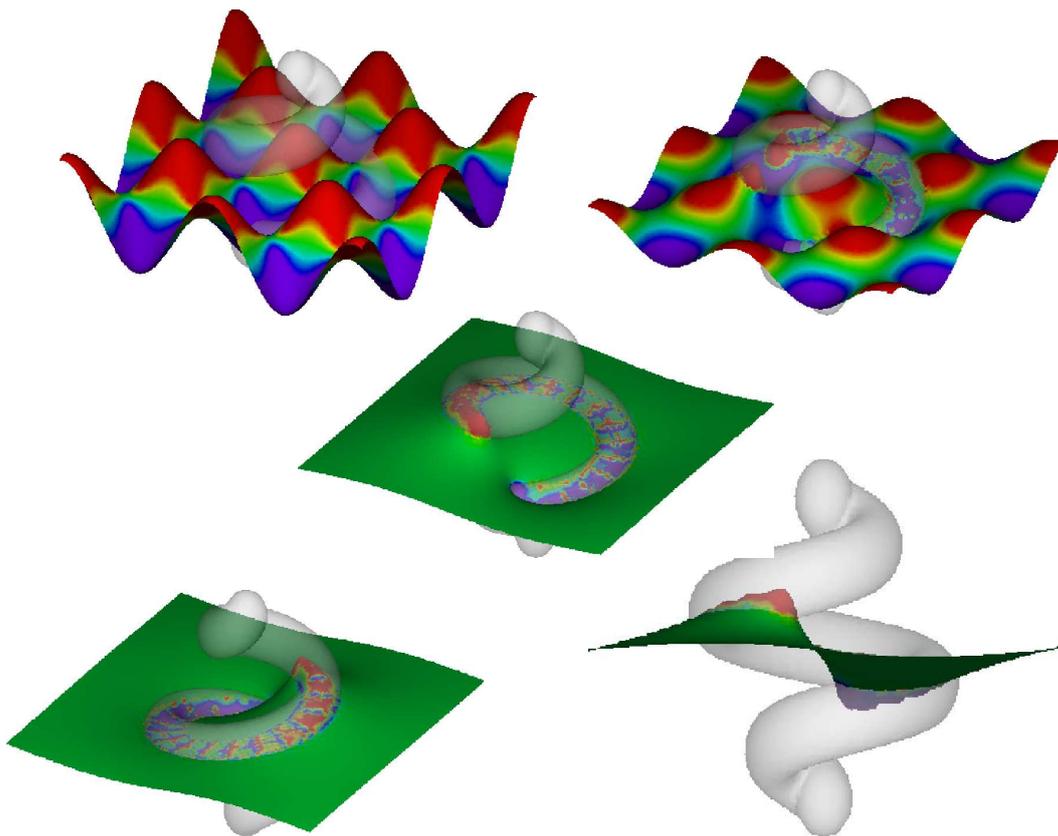


FIG. 11. First two images show sample evolution with a coil-shaped void. Last three views are of the final minimal surface.

TABLE 1

Time required for reaching equilibrium in the example in Figure 7 on a Dell workstation. All measurements are in seconds.

Mesh	Time (sec.)
50 × 50	18
100 × 100	229
200 × 200	1435
400 × 400	20774

Finally, to demonstrate the ease of setting up arbitrary void regions, we show, in Figure 11, an evolution where the void is formed by the distance function from a winding coil-shaped path. The first two images show the evolution from an initial sine curve and the last three views are the final minimal surface.

5.2 Results from the second algorithm

In Figure 12, we present a sample evolution with square voids using the second algorithm. The initial condition (Figure 12a) is a simple sine wave but interpolated on the finite element mesh. Two intermediate states are shown, and then the final minimal surface is shown in Figure 12d. When the minimal surface (Figure 12d) enters through a side of the square, it appears to enter orthogonally. We note that at the corners of the squares, the path appears to enter at 45° . This is consistent with the results of the first algorithm, in Figure 4.

Figure 13 shows the final contour corresponding to an evolution with staggered voids using the second algorithm. The exclusions are shown, surrounded by a layer of small elements that were introduced to resolve the sharp change in material properties along the boundaries of the exclusions. The contour passes through all of the exclusions. Table 2 shows the compute times for the previous two examples.

TABLE 2

Time required for reaching equilibrium in the examples in Figures 12 and 13 on a linux workstation. All measurements are in seconds.

Mesh	Time (sec.)
Figure 12	135
Figure 13	115

We note that the same solution as in Figure 13 is obtained by taking simpler initial conditions, such as a straight line. One aspect of using a straight line for an initial condition should be mentioned. Since a straight line has zero mean curvature, the algorithm can get “stuck” in this local minimum if the reinitialization is done too often. Thus, in practical computations, care must be taken to allow the algorithm to proceed for a sufficient number of time steps before reinitializing. We recall that reinitialization is governed by the parameter T_{chop} .

This is more fully illustrated by examining the behavior of the previous example for different values of the parameter T_{chop} , using a straight line as an initial condition. Figure 14 shows the straight-line initial condition. Figure 15 shows a sequence of final surfaces, obtained using $T_{\text{chop}} =$

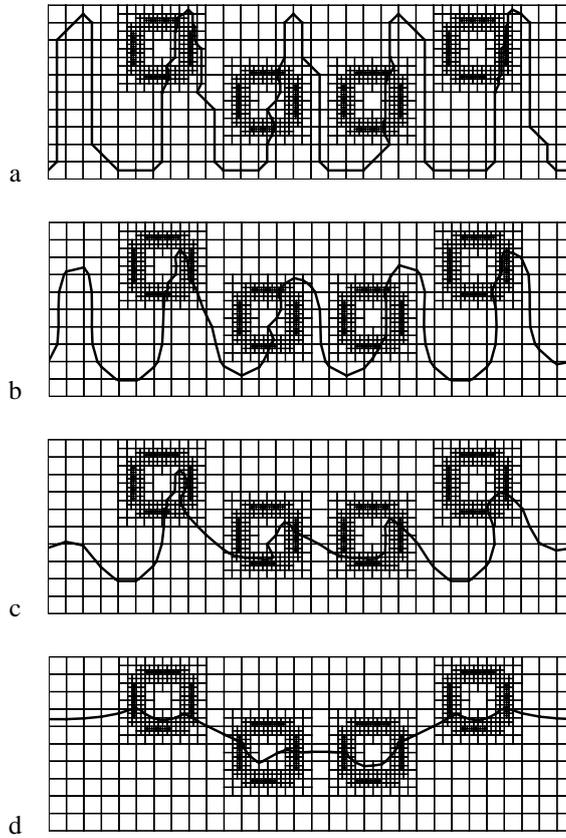


FIG. 12. Sample evolution with four square voids using the second algorithm. The parameters are $\text{time_step} = 0.0035(\text{s})$, $\epsilon = 0.01$, and $T_{\text{chop}} = 0.07(\text{s})$.

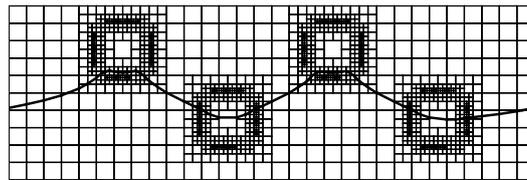


FIG. 13. Final minimal surface with square voids using the second algorithm. The same sine wave initial condition given in Figure 12a is used. The parameters are $\text{time_step} = 0.0035(\text{s})$, $\epsilon = 0.01$, and $T_{\text{chop}} = .07(\text{s})$.

0.0035, 0.0175, 0.07, and 0.14(s). The first two values of T_{chop} are too small, and the surface does not evolve sufficiently. The final two values of T_{chop} appear to be sufficient. We remark that if T_{chop} is chosen to be too large, then the linearization (16) will not be an accurate approximation to the solution of the full nonlinear equation (15).

The parameter ϵ must also be chosen carefully. If ϵ is chosen to be too large, then it will not correctly represent the effect of the exclusions, which effectively require a value of $\epsilon = 0$. On

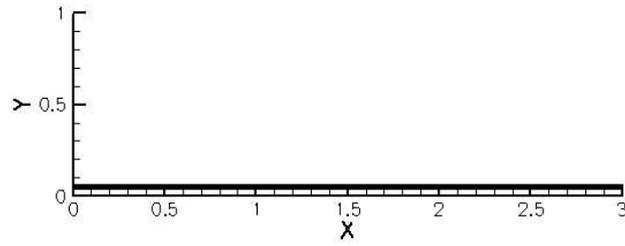


FIG. 14. Initial surface used for Figure 15.

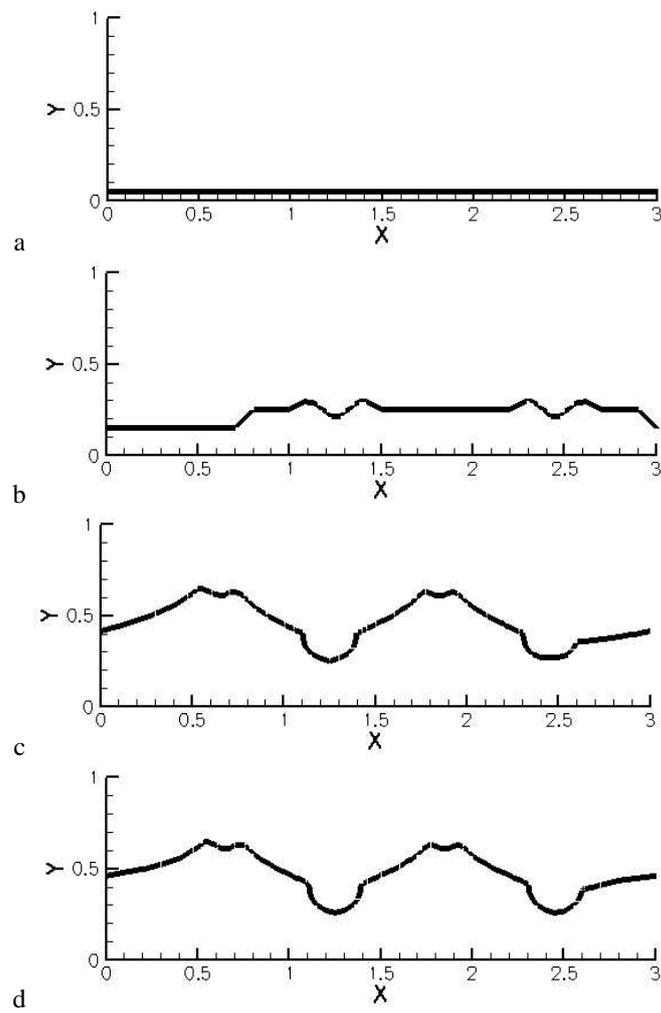


FIG. 15. The variation of the solution with increasing T_{chop} is studied. The curves a–d correspond to T_{chop} of 0.0035(s), .0175(s), .07(s), and 0.14(s). The time step in this case is 0.0035(s), and $\epsilon = 0.01$. Little change is observed between the two larger values of T_{chop} .

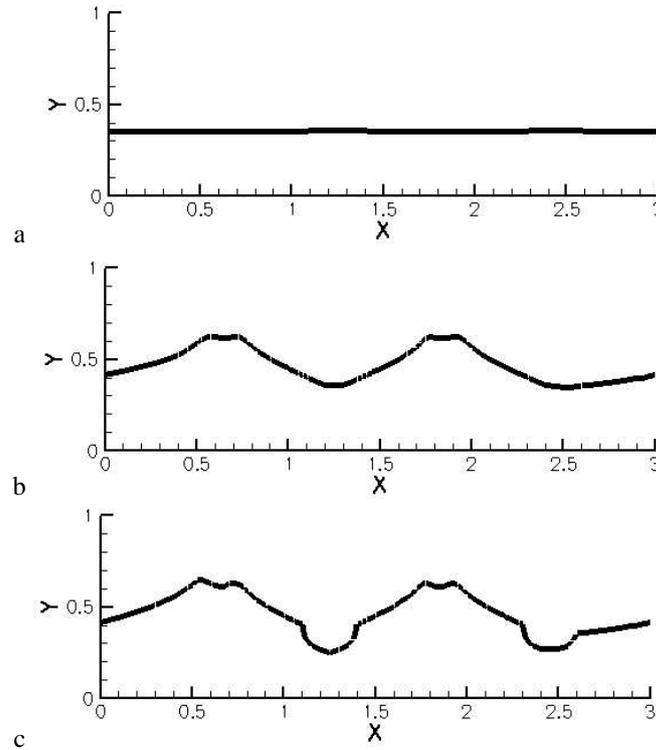


FIG. 16. The variation of the solution with decreasing ϵ is studied. The curves a-c correspond to ϵ of 0.5, .05, and .01. The time step in this case is 0.0035(s), with $T_{\text{chop}} = 0.07$ (s). Little change is observed between the two smaller values of ϵ .

the other hand, if ϵ is chosen too small, then the weak formulation of equation (16) becomes ill-conditioned, thus precluding an accurate finite element solution. Figure 16a shows the same example presented in Figure 13, except ϵ is chosen as $\epsilon = 0.5$, which is obviously too large. In this case, the resulting solution fails to pass through the upper two exclusions, since the weight is not small enough for the exclusions to attract the surface. Figures 16b and 16c show the results for $\epsilon = 0.05$ and $\epsilon = 0.01$, and these converge to the same solution as was seen previously.

Next, we discuss numerical convergence of the algorithm. There are two issues at hand—first, the convergence of the weighted BMO linearization of the nonlinear equation (15). A proof of this can be found in [35]. The second issue is the convergence of the finite element algorithm to the solution of equation (16). We present temporal and spatial convergence examples.

Figure 17 shows the surfaces obtained from using time steps of 0.028(s), 0.0035(s), 0.0017(s), and 0.000875(s), for the same example as in Figure 13. The latter two time steps are sufficiently fine that there is little difference between them, indicating convergence of the time discretization.

Figure 18 shows the surfaces obtained from using 0, 1, 2, and 3 levels of local refinements around the boundaries of the exclusions, again for the example of Figure 13. The only noticeable change with refinement level is around the boundaries of the exclusions whose centers are at 0.65 and 1.85

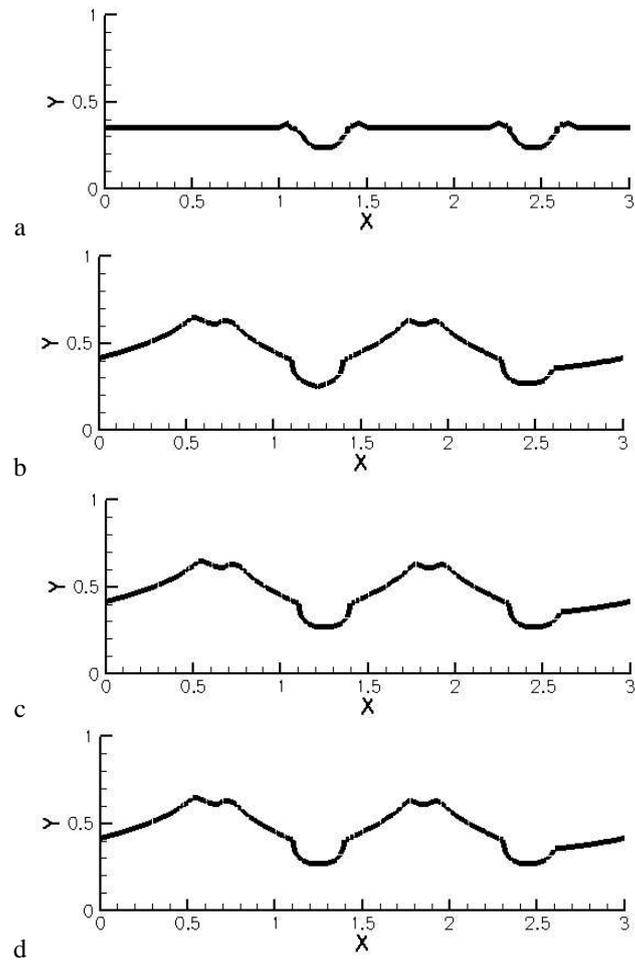


FIG. 17. Time discretization convergence study. In this case the variation of the solution with successively smaller time steps is studied. The curves a–d correspond to time steps of 0.028(s), 0.0035, 0.0017(s), and 0.000875(s). Little change is observed between the two smaller time steps, indicating convergence in time. The parameters are $T_{\text{chop}} = 0.07(\text{s})$, and $\epsilon = 0.01$.

along the X axis. It appears that the local behavior of the surfaces is changing with refinement level, but from the scale it is difficult to see.

Therefore, in Figure 19 we investigate more closely the effects of local adaptivity in the example from Figure 18. These figures show a sequence of blow ups around one of the boundaries of an exclusion. Each subfigure shows an increased level of adaptivity, resulting in more and more degrees of freedom near the exclusion boundary. With more levels of adaptivity, the contour is shifted vertically and, more importantly, the angle at which the contour impinges on the exclusion gradually approaches 90° . For the case with no adaptivity, the angle of incidence is approximately 75° , far from the ideal of normal incidence. This illustrates the need for h -adaptivity in resolving the fine properties of the contour near the boundary of the exclusion.

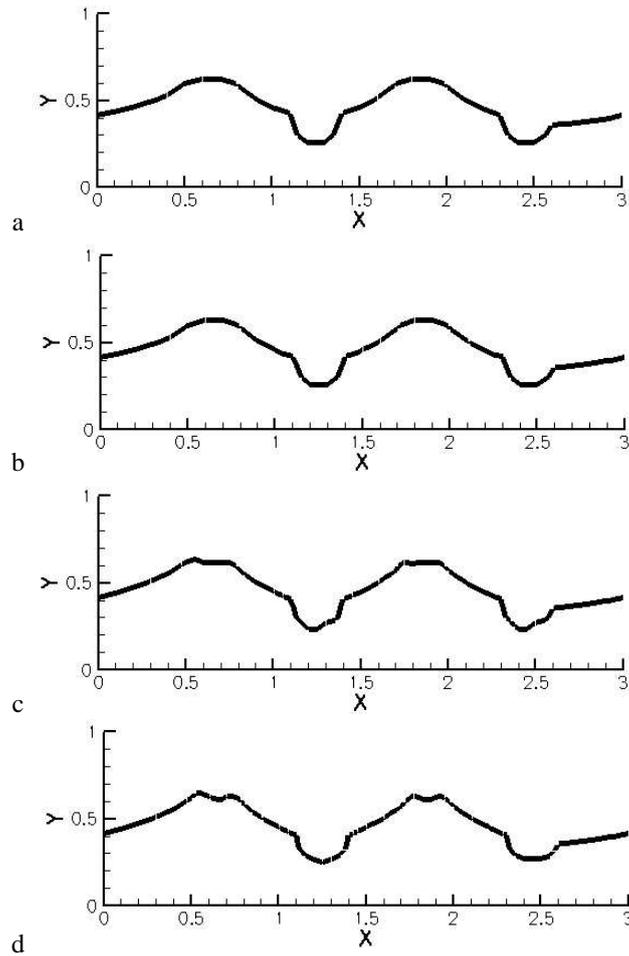


FIG. 18. Spatial discretization convergence study. In this case the variation of the solution with increasing numbers of mesh refinements near the exclusion boundaries is studied. The curves a–d correspond to 0, 1, 2, and 3 refinements. After one refinement, the solution converges quickly. The parameters are $\text{time_step} = .0035(\text{s})$, $T_{\text{chop}} = 0.07(\text{s})$, and $\epsilon = 0.01$.

6. Conclusions

The two algorithms presented in this paper are quite different, and yet they lead to similar results. They also provide insight into the issues raised in the introduction.

1. We have extended standard level set methods to handle the case with exclusions. For the first algorithm, this involved developing a numerical approach for enforcing numerically the orthogonality condition. For the second algorithm, this involved extending the standard BMO algorithm to include a convection term in addition to the standard diffusion term.
2. For both algorithms, the exclusions act as attractors for the surfaces; which is expected since the metric degenerates inside the exclusions, and hence the surfaces tend to enter the exclusions.

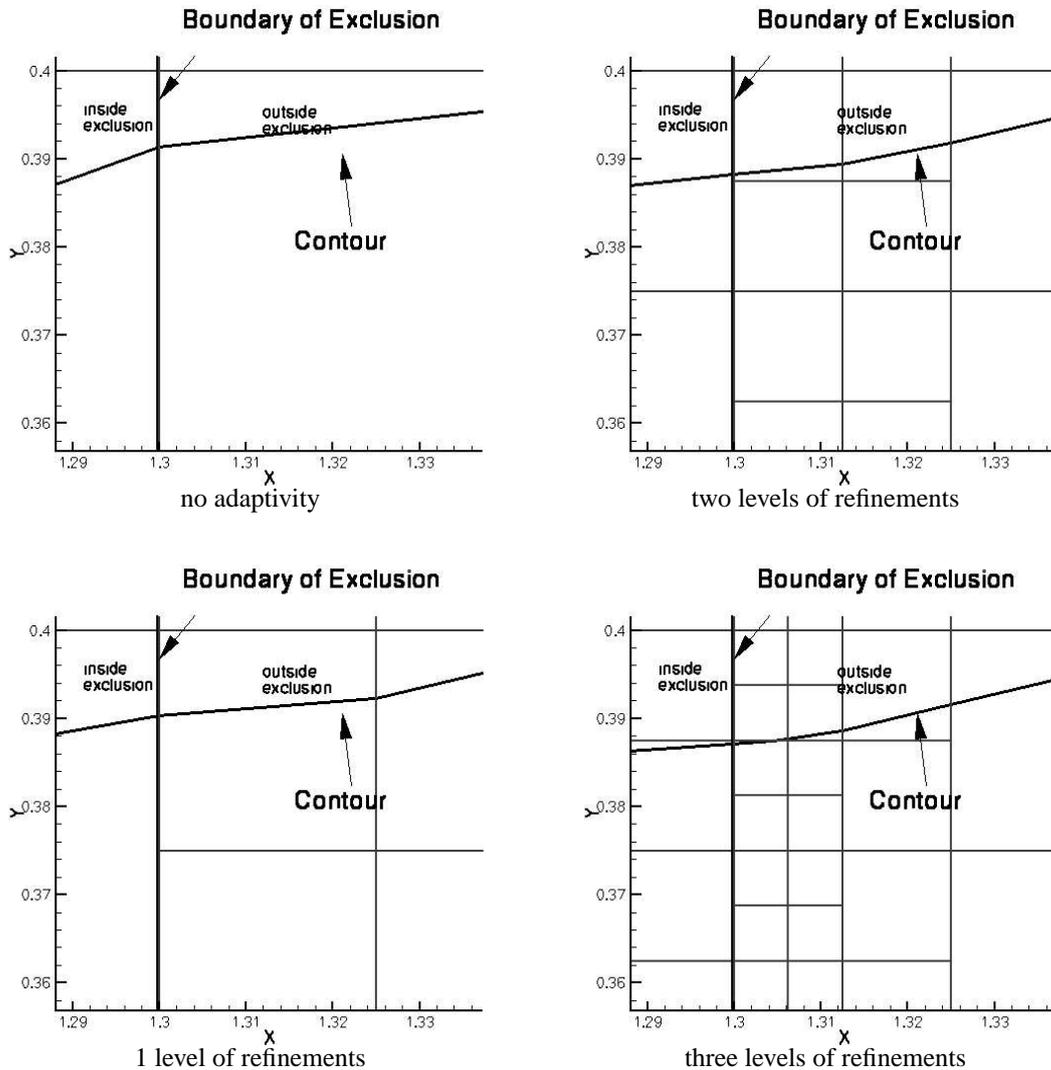


FIG. 19. The dependence of the angle of incidence on the level of adaptivity.

3. The first algorithm implements the orthogonality condition while the second one uses an additional weight function which produces the same orthogonality condition at equilibrium.

Acknowledgments

The authors are grateful to Dr. Leszek Demkowicz at the Texas Institute for Computational and Applied Mathematics for the 2Dhp90 code.

REFERENCES

1. ADALSTEINSSON, D. & SETHIAN, J. A. A fast level set method for propagating interfaces. *J. Comput. Phys.* **118** (1995), 269–277. Zbl 0823.65137 MR 1329634
2. ADALSTEINSSON, D. & SETHIAN, J. A. A level set approach to a unified model for etching, deposition, and lithography. III. Redeposition, reemission, surface diffusion, and complex simulations. *J. Comput. Phys.* **138** (1997), 193–223. Zbl 0952.65113 MR 1484289
3. ADALSTEINSSON, D. & SETHIAN, J. A. The fast construction of extension velocities in level set methods. *J. Comput. Phys.* **148** (1999), 2–22. Zbl 0919.65074 MR 1665209
4. BARLES, G. & GEORGELIN, C. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM J. Numer. Anal.* **32** (1995), 484–500. Zbl 0831.65138 MR 1324298
5. BARLES, G. & SOUGANIDIS, P. E. A new approach to front propagation problems: theory and applications. *Arch. Rat. Mech. Anal.* **141** (1998), 237–296. Zbl 0904.35034 MR 1617291
6. BRAKKE, K. A. *The Motion of a Surface by its Mean Curvature*. Princeton Univ. Press, Princeton, NJ (1978). Zbl 0386.53047 MR 0485012
7. BRAKKE, K. A. The surface evolver. *Experiment. Math.* **1** (1992), 141–165. Zbl 0769.49033 MR 1203871
8. BRAKKE, K. A. The surface evolver and the stability of liquid surfaces. *Philos. Trans. Roy. Soc. London Ser. A* **354** (1996), 2143–2157. Zbl 0870.76065 MR 1421042
9. BRAKKE, K. A. & SULLIVAN, J. M. Using symmetry features of the surface evolver to study foams. *Visualization and Mathematics* (Berlin-Dahlem, 1995), Springer, Berlin (1997), 95–117 MR 1607360
10. CAFFARELLI, L. A. & DE LA LLAVE, R. Planelike minimizers in periodic media. *Comm. Pure Appl. Math.* **54** (2001), 1403–1441. Zbl 1036.49040 MR 1852978
11. CHOPP, D. L. Computing minimal surfaces via level set curvature flow. *J. Comput. Phys.* **106** (1993), 77–91. Zbl 0786.65015 MR 1214016
12. CHOPP, D. L. Computation of self-similar solutions for mean curvature flow. *Experiment. Math.* **3** (1994), 1–15. Zbl 0811.53011 MR 1302814
13. CHOPP, D. L. Some improvements of the fast marching method. *SIAM J. Sci. Comput.* **23** (2001), 230–244. Zbl 0991.65105 MR 1860913
14. CHOPP, D. L. & SETHIAN, J. A. Flow under curvature: Singularity formation, minimal surfaces, and geodesics. *Experiment. Math.* **2** (1993), 235–255. Zbl 0806.53004 MR 1281473
15. CONCUS, P., FINN, R., & HOFFMAN, D. E. (eds.) *Geometric Analysis and Computer Graphics* (Berkeley, 1988). Math. Sci. Res. Inst. Publ. 17, Springer, New York (1991). Zbl 0776.68009 MR 1081324
16. DEMKOWICZ, L. 2d hp-adaptive finite element package (2dhp90) version 2.0. TICAM Report 02-06, The University of Texas at Austin, 2002.
17. DEMKOWICZ, L., GERDES, K., C., S., BAJER, A., & WALSH, T. Hp90: A general and flexible fortran 90 hp-fe code. *Computing and Visualization in Science* **1** (1998), 145–163. Zbl 0912.68014
18. DUBOIS-VIOLETTE, E. & PANSU, B. E. (eds.) *International Workshop on Geometry and Interfaces* (Aussois, 1990). Les Éditions de Physique, Les Ulis (1990). MR 1090143
19. EVANS, L. C. Convergence of an algorithm for mean curvature motion. *Indiana Univ. Math. J.* **42** (1993), 533–557. Zbl 0802.65098 MR 1237058
20. EVANS, L. C. & GARIEPY, R. F. *Measure Theory and Fine Properties of Functions*. CRC Press, Boca Raton, FL (1992). Zbl 0804.28001 MR 1158660
21. EVANS, L. & SPRUCK, J. Motion of level sets by mean curvature. I. *J. Differential Geom.* **33** (1991), 635–681. Zbl 0726.53029 MR 1100206
22. EVANS, L. & SPRUCK, J. Motion of level sets by mean curvature. II. *Trans. Amer. Math. Soc.* **330** (1992), 321–332. Zbl 0776.53005 MR 1068927

23. EVANS, L. & SPRUCK, J. Motion of level sets by mean curvature. III. *J. Geom. Anal.* **2** (1992), 121–150. Zbl 0768.53003 MR 1151756
24. EVANS, L. & SPRUCK, J. Motion of level sets by mean curvature. IV. *J. Geom. Anal.* **5** (1995), 77–114. Zbl 0829.53040 MR 1315658
25. FEDERER, H. *Geometric Measure Theory*. Grundlehren Math. Wiss. 153, Springer, New York (1969). Zbl 0176.00801 MR 0257325
26. FINN, R. *Equilibrium Capillary Surfaces*. Springer (1986). Zbl pre00583.35002 MR 0816345
27. GIUSTI, E. *Minimal Surfaces and Functions of Bounded Variation*. Birkhäuser, Basel (1984). Zbl pre00545.49018 MR 0775682
28. GOMES, D. A. Stability of rotating liquid films. *Quart. J. Mech. Appl. Math.* **55** (2002), 327–343. Zbl 1036.76018 MR 1919971
29. GRAYSON, M. A. The heat equation shrinks embedded plane curves to round points. *J. Differential Geom.* **26** (1987), 285–314. Zbl 0667.53001 MR 0906392
30. GRAYSON, M. A. A short note on the evolution of a surface by its mean curvature. *Duke Math. J.* **58** (1989), 555–558. Zbl 0677.53059 MR 1016434
31. GRÜTER, M. Optimal regularity for codimension one minimal surfaces with a free boundary. *Manuscripta Math.* **58** (1987), 295–343. Zbl 0609.35012 MR 0893158
32. HSU, L., KUSNER, R. & SULLIVAN, J. Minimizing the squared mean curvature integral for surfaces in space forms. *Experiment. Math.* **1** (1991), 191–207. Zbl 0778.53001 MR 1203874
33. HUISKEN, G. Flow by mean curvature of convex surfaces into spheres. *J. Differential Geom.* **20** (1984), 237–266. Zbl pre00556.53001 MR 0772132
34. KRAYNIK, A. & REINELT, D. Simple shearing flow of a dry kelvin soap foam. *J. Fluid Mech.* **311** (1996), 327–342. Zbl 0864.76008.
35. LEONI, F. Convergence of an approximation scheme for curvature-dependent motions of sets. *SIAM J. Numer. Anal.* **39** (2001), 1115–1131. Zbl 1008.65067 MR 1870835
36. LI, Z. L., LIN, X. B., TORRES, M., & ZHAO, H. K. Generalized Snell's law for weighted minimal surfaces in heterogeneous media. *Methods Appl. Anal.* **10** (2003), 199–214. Zbl pre02110033 MR 2074748
37. MALLADI, R. & SETHIAN, J. A. Image processing via level set curvature flow. *Proc. Nat. Acad. Sci. U.S.A.* **92** (1995), 7046–7050. Zbl 1023.68689 MR 1343443
38. MALLADI, R. & SETHIAN, J. A. Level set methods for curvature flow, image enhancement, and shape recovery in medical images. *Visualization and Mathematics* (Berlin-Dahlem, 1995), Springer, Berlin (1997), 329–345. MR 1607328
39. MERRIMAN, B., BENCE, J. K., & OSHER, S. J. Motion of multiple functions: a level set approach. *J. Comput. Phys.* **112** (1994), 334–363. Zbl 0805.65090 MR 127728
40. MICHALET, X., FOURCADE, B., & BENSIMON, D. Fluctuating vesicles of nonspherical topology. *Phys. Rev. Lett.* **72** (1994), 168–171.
41. MITTELMANN, H. Symmetric capillary surfaces in a cube. *Math. Comput. Simulation* **35** (1993), 139–152 MR 1290499
42. NITSCHKE, J. C. C. *Lectures on Minimal Surfaces. Vol. 1*. Cambridge Univ. Press, Cambridge (1989). Zbl 0688.53001 MR 1015936
43. OSHER, S. & SETHIAN, J. A. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* **79** (1988), 12–49. Zbl 0659.65132 MR 0965860
44. PHELAN, R., WEAIRE, D., & BRAKKE, K. Computation of equilibrium foam structures using the surface evolver. *Experiment. Math.* **4** (1995), 181–192. Zbl 0861.76014 MR 1387476
45. RACZ, L., SZEKELY, J., & BRAKKE, K. A general statement of the problem and description of a proposed method of calculation for some meniscus problems in materials processing. *ISIJ Int.* **33** (1993), 328–335.

46. SETHIAN, J. Fast marching methods. *SIAM Rev.* **41** (1999), 199–235. Zbl 0926.65106 MR 1684542
47. SETHIAN, J. A. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.* **93** (1996), 1591–1595. Zbl 0852.65055 MR 1374010
48. SETHIAN, J. A. *Level Set Methods. Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge Monogr. Appl. Comput. Math. 3, Cambridge Univ. Press, Cambridge (1996). Zbl 0859.76004 MR 1409367
49. SETHIAN, J. A. *Level Set Methods and Fast Marching Methods. Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* 2nd ed., Cambridge Monogr. Appl. Comput. Math. 3, Cambridge Univ. Press, Cambridge (1999). Zbl 0973.76003 MR 1700751
50. SETHIAN, J. A. Advancing interfaces: level set and fast marching methods. *ICIAM 99* (Edinburgh), Oxford Univ. Press, Oxford (2000), 222–233. Zbl 0994.65090 MR 1824446
51. SETHIAN, J. A. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *J. Comput. Phys.* **169** (2001), 503–555. Zbl 0988.65095 MR 1836524
52. SETHIAN, J. A. A review of level set and fast marching methods for image processing. *Modern Methods in Scientific Computing and Applications* (Montréal, QC, 2001), NATO Sci. Ser. II Math. Phys. Chem. 75, Kluwer, Dordrecht (2002), 365–396. Zbl pre02110608 MR 2004360
53. SETHIAN, J. A. & CHOPP, D. L. Physics and geometry of flow under curvature: singularity formation, minimal surfaces, geodesics, and surface tension. *Motion by Mean Curvature and Related Topics* (Trento, 1992), de Gruyter, Berlin (1994), 168–185. Zbl 0815.49027 MR 1277398
54. SETHIAN, J. A. & SMEREKA, P. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics, Vol. 35*, Annual Reviews, Palo Alto, CA (2003), 341–372. Zbl 1041.76057 MR 1967017
55. SHAPIRO, G. *The Mathematics of Soap Films: Explorations with Maple.* Amer. Math. Soc. (2000).
56. SHAPIRO, G. *Geometric Partial Differential Equations and Image Analysis.* Cambridge Univ. Press (2001).
57. SIMON, L. *Lectures on Geometric Measure Theory.* Austral. Nat. Univ. Centre for Math. Analysis, Canberra (1983). Zbl 0546.49019 MR 0795227
58. TEGART, J. Three-dimensional fluid interfaces in cylindrical containers. AIAA paper AIAA-912174, 27th Joint Propulsion Conference, Sacramento, CA (1991).
59. TORRES, M. Plane-like minimal surfaces in periodic media with exclusions. *SIAM J. Math. Anal.* **36** (2004), 523–551. Zbl pre02139025 MR 2111789