Phase Transitions, Curve Evolution, and the Control of Semiconductor Manufacturing Processes

Jordan Berg
Department of Mechanical Engineering
Texas Tech University
Lubbock, TX 79409-1021
jberg@ttu.edu

Anthony Yezzi and Allen Tannenbaum
Department of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455
tannenba@ee.umn.edu

Abstract

Perhaps the most common approach to solving PDEs in engineering applications is through use of finite-element approximations. In free-boundary problems this usually requires adaptive meshing that tracks the surface motion. These meshing schemes may not be able to handle shock formation or topological changes without arcane logic or user intervention. Inconvenient in simulation, this can cause grave difficulties in real-time control. Thus alternatives are needed. We find these alternatives under development in the surprisingly related areas of phase transition physics and computer vision. We present (i) A mathematical representation of free surface motion that is particularly well-suited to real-time implementation, (ii) a technique for estimating an isotropic and homogeneous normal velocity based on a simple measurement and (iii) application to a semiconductor etching problem.

1 Introduction

Given a material that may exist in either of two phases, how will the boundaries between the phases evolve, and what equilibrium configurations are possible? This is a fundamental question of phase transition physics.

One way of mathematically representing phase transition, to investigate these issues, is through an *order parameter*, Φ. The value of Φ at any point indicates the state of the material, with (for example) zero corresponding to one of the phases, unity to the other. Analysis of the transition dynamics proceeds by allowing Φ to vary continuously, without concern for the physical meaning of intermediate values. Then the following parabolic *phase field* equation is an expression of energy minimization for many important physical processes:

$$\Phi_t = \epsilon \Delta \Phi + (1/\epsilon) W'(\Phi) \tag{1}$$

[*] This work was supported in part by grants from the National Science Foundation ECS-9122106, by the Air Force Office of Scientific Research AF/F49620-94-1-0058DEF, and by the Army Research Office DAAH04-94-G-0054, DAAH04-93-G-0332s, and by the Institute for Mathematics and its Applications.

where W is a double-well potential with a local minima at one and another at zero—the local minima correspond to the two phases. As ϵ goes to zero, the domain will typically split into distinct subdomains, with Φ tending to values of zero or one in each—to one of the two phases or minima—except in boundary regions, where one phase smoothly transforms to the other. The width of these boundary regions goes to zero with ϵ; in the limit the phases are separated by a sharp interface. Thus, in the order parameter approach a sharp interface is derived as an approximation, in the limit as the thickness of the region of phase change vanishes. The motion of the sharp interface may be studied through the limiting behavior of Eq. (1). Its normal velocity, that is, its velocity projected along the normal to the interface, turns out to be given by the sum of two terms. One term is related to the curvature of the interface, the other is a constant *inflationary* term, which vanishes if the values of W at its local minima are equal. In most physical situations, the curvature term tends to reduce the interface area, corresponding to a decrease in surface energy. This tendency may be balanced, however, by a strong inflationary term, driving the system from a high energy phase to a low energy phase. Limiting models of this type include the Allen-Cahn antiphase boundary model, the Stefan problem, the Hele-Shaw model, dendritic solidification, and thermal grooving. Kichenassamy, *et al.* [10], present a detailed discussion of phase field equations, together with a large set of references to which we refer the interested reader.

Although Eq. (1) gives insight into the behavior of the phase transition problem, it is not necessary to work with it directly. Rather, one can write down the equations of motion of the evolving interface directly. In two dimensions, these are the equations of *curve evolution.* Here, the moving interface is described by a family of parameterized curves, $C : [0,1] \times [0,t_f) \to \mathbf{R}^2$. The curve describing the interface evolves according to,

$$\frac{\partial C}{\partial t} = \alpha(s,t)\mathcal{T} + \hat{\beta}(s,t)\mathcal{N} \tag{2}$$

where s parametrizes the curve, $\mathcal{N}$ is the normal vector, $\mathcal{T}$ is the tangent vector, and α, $\hat{\beta}$ are velocity functions. Since we are interested in shape only we may take $\alpha = 0$. Changing α changes only the curve's parametrization, and not its shape. Much of the mathematical literature on curve evolution considers the special case where the motion is determined solely by the local geometry of the curve. This leads to the following equation:

$$\frac{\partial C}{\partial t} = \beta(\kappa)\mathcal{N}. \tag{3}$$

where $\kappa(s,t)$ is the curvature. Let $\beta_0 := \beta(0)$ denote the inflationary contribution, which will be essential in our study of etching processes below. For a rigorous formal treatment, as well as a look at some velocity functions of special interest, see [6, 7, 8, 11, 22].

Numerical solution of curve evolution problems can be difficult, for several reasons. One is that, if curvature terms are absent or small, the solution will typically develop corners, or "shocks," even if the initial data is smooth. Solving for the curve motion after a shock develops—in fact, just giving meaning to such a notion—requires that some form of weak solution be defined, along with associated entropy conditions or viscosity solutions to ensure uniqueness. Another difficulty arises in treating topological transitions of the evolving curve, such as arise when curves merge or split. A series of algorithms that successfully address both these problems have been pioneered by Osher and Sethian, and their coworkers, based in part on techniques for *hyperbolic conservation laws* [17, 18, 22]. These algorithms employ a *level set* representation of the interface, and are the foundation of our numerical approach.

Perhaps the most common approach to solving PDEs in engineering applications is through use of finite-element approximations. In free-boundary problems this often requires adaptive meshing that tracks the surface motion (As in, for example, [13]). These meshing schemes may not be able to handle shock formation or topological changes without tortuous logic or user intervention. Inconvenient in simulation, this is disastrous in real-time control. The techniques mentioned above provide an alternative.

The idea of applying the theory of curve evolution for modeling surface development in reactive etching processes has been considered by a number of authors, in particular Shaqfeh and Jurgensen [24], and Singh, *et al.* [25]. Adalsteinsson and Sethian build and greatly extend this methodology, culminating in a powerful level set approach to modeling etching, deposition, and lithography fabrication processes [1, 22]. (See also Katardjiev *et al.* for a discussion of curvature dependent flows and level sets in plasma etching [9].) The present work, which emphasizes the use of such models as the basis for real-time estimation, owes a great deal to the efforts of these researchers. Finally, work that attacks a similar real-time interface estimation and control problem (for crystal growth) through a fixed-grid method has been presented by Srinivasan, *et al.* [29]. Though related through the interpretation of the enthalpy as a level set function, nevertheless the authors rely on a finite element discretization of the field equations, and so the technique is quite different.

The authors would like to thank Pramod Khargonekar for his very helpful information on the control literature in the area of semiconductor manufacturing. Also we would like to thank Jack and Ted Higman for enlightening conversations on thin-film processing.

2 Curvature Flows and Interface Evolution

Let us discuss some properties of Eq. (3), in particular those that give rise to difficulties in analyzing the motion. In what follows, for $C(t) = (x(s,t), y(s,t))$, $\rho(s,t)$ will denote the metric, $[x_s^2 + y_s^2]^{1/2}$. $\tilde{s}$ will be the *Euclidean arc-length parameter,* $\tilde{s}(s,t) := \int_0^s \rho(\zeta,t)d\zeta$. Note that the total length of the curve is just $L(t) = \tilde{s}(1,t)$. θ is defined to be the angle between the tangent $\mathcal{T}$ and the x-axis. The tangent, $\mathcal{T}$, curvature, κ, and normal, $\mathcal{N}$ are defined in the standard way [5]. Further, we let

$$K(t) := \int_0^1 \kappa(s,t)\rho(s,t)ds \tag{4}$$

denote the *total curvature.* The following formula, a derivation of which may be found in several places [7, 8, 12, 18], will be required later:

$$\frac{\partial L}{\partial t} = \int_0^1 \beta\kappa\rho ds \tag{5}$$

2.1 Hyperbolic Conservation Laws

In principle, corners can form only when curvature terms are absent. In practice, the curvature can become extremely large, and cause numerical problems, even when these terms are nonzero. However, the causes and handling of shocks is best understood by studying the special case $\beta =$ constant. Here in the classical manner we will derive a hyperbolic conservation law, following the treatment of [18].

Without loss of generality, let $\beta_0 = 1$. Then,

$$x_t = \frac{y_s}{(x_s^2 + y_s^2)^{1/2}}, \tag{6}$$

$$y_t = -\frac{x_s}{(x_s^2 + y_s^2)^{1/2}}. \tag{7}$$

From equations (6) and (7) we can derive a hyperbolic conservation law. As long as C stays smooth (with respect to s) and non-self-intersecting, by virtue of the implicit function theorem, we can express the front in the form

$$y = U(t, x). \tag{8}$$

(Note that C_t is the graph of (8).) One can then verify that U satisfies the Hamilton-Jacobi equation

$$\frac{\partial U}{\partial t} - (1 + U_x^2)^{1/2} = 0. \tag{9}$$

Set $u := \partial U / \partial x$. Then differentiating (9) with respect to x we see that

$$u_t - \Big((1 + u^2)^{1/2}\Big)_x = 0. \tag{10}$$

which has the form of a hyperbolic conservation law. There is a huge classical and modern literature devoted to equations of this type; see [26] and the references therein.

Geometrically, it is very easy to see how discontinuities develop for the system (6,7). Indeed, assuming that C_t remains smooth, one may compute that the curvature satisfies the following evolution equation:

$$\kappa_t = -\kappa^2. \tag{11}$$

Now we can explicitly solve (11) to find that

$$\kappa(s, t) = \frac{\kappa(s, 0)}{1 + t\kappa(s, 0)}. \tag{12}$$

Notice then that if the initial curve is anywhere concave, i.e., has curvature negative at any point, $\kappa(s, t)$ will blow up in finite time, and the resulting curve will develop a singularity, i.e., a *shock*.

Once such shocks occur, one must be careful in defining precisely what is meant by a "solution" to (10), since the curve is now non-differentiable and so cannot satisfy a differential equation. Nonconvex optics problems can be handled by the Huygens principle, which defines the propagating front as the *envelope* of a continuum of circles, centered on the initial front. This construction selects a *unique* solution from the many that satisfy the weak form of the eikonal equation. A similar problem arises in solving the Euler equations of compressible flow. This is precisely where the notion of viscosity solution arises. Viscosity solutions may be defined for curvature flows as well, with curvature playing the role of viscosity. In these flows, the viscosity solution coincides with the entropy condition imposed in [22], interpreting the hyperbolic evolution law in the prairie-fire sense that *"once a particle is burnt, it stays burnt."* The power of the unifying mathematical principles underlying these seemingly disparate areas can be glimpsed in the fact that for prairie fires and semiconductors, as well as for geometric optics, the proper entropy condition corresponds exactly to Huygens' classical construction.

2.2 Level Set Representations

We now briefly discuss some of the numerical algorithms developed for curve evolution. Much of this work is based on writing the approximations in *conservation form* and applying the *Godunov method* [14]. For more details, see the fundamental work of Osher and Sethian in [17, 18, 22].

Let $C(s,t) : S^1 \times [0,\tau) \to \mathbf{R}^2$ be a family of curves satisfying the following evolution equation:

$$\frac{\partial C}{\partial t} = \beta(\kappa)\mathcal{N}. \tag{13}$$

In numerical implementations, the evolving curve is embedded in a two dimensional surface, and then the equations of motion are solved using a combination of straightforward discretization, and numerical techniques derived from hyperbolic conservation laws and Hamilton-Jacobi theory [27].

The embedding step is done in the following manner: The curve $C(s,t)$ is represented by the zero level set of a smooth and Lipschitz continuous function $\Phi : \mathbf{R}^2 \times [0,\tau) \to \mathbf{R}$. Assume that Φ is negative in the interior and positive in the exterior of the zero level set. We consider the zero level set, defined by

$$\left\{X(t) \in \mathbf{R}^2 \,:\, \Phi(X,t) = 0\right\}. \tag{14}$$

We have to find an evolution equation of Φ, such that the evolving curve $C(t)$ is given by the evolving zero level $X(t)$, i.e.,

$$C(t) \equiv X(t)\,. \tag{15}$$

By differentiating $\Phi(X(t),t) = 0$ we obtain:

$$\nabla\Phi(X,t) \cdot X_t + \Phi_t(X,t) = 0\,. \tag{16}$$

Note that for the zero level, the following relation holds:

$$\frac{\nabla\Phi}{\parallel \nabla\Phi \parallel} = \mathcal{N}\,. \tag{17}$$

In this equation, the left side uses terms of the surface Φ, while the right side is related to the curve $\mathcal{C}$. Combining equations (13) to (17) gives

$$\Phi_t + \beta(\kappa) \parallel \nabla\Phi \parallel = 0 \tag{18}$$

and the curve C, evolving according to (13), is obtained by the zero level set of the function Φ, which evolves according to (18). Sethian [22] called this scheme an *Eulerian formulation* for front propagation, because it is written in terms of a fixed coordinate system.

The second step of the algorithm consists of the discretization of the equation (18). If singularities cannot develop during the evolution, as in the geometric heat equation flow, a straightforward discretization can be performed [18]. If singularities can develop, as in the case of $\beta = 1$, a special discretization must be implemented. In this case, the implementation of the evolution of Φ is based on a *monotone* and *conservative* numerical algorithm, derived from the theory of hyperbolic conservation laws [14, 18, 27]. For a large class of functions β of this type, this numerical scheme automatically obeys the entropy condition derived from the Huygens principle [27]. For velocity functions such as $\beta = \beta_1\kappa + \beta_0$, where curvature in theory remains finite but in practice may become very large, a combination of both methods is used [18].

3 Piecewise Constant Normal Velocity

The equations given in Section 2 describe a *smooth* curve. However, the cases to be considered below will only be *piecewise* smooth. We will need an expression for $\partial L/\partial t$ for a continuous curve made up of smooth segments, with the segments joined at corners, or shocks, at which the curve fails to be differentiable. In the case to be considered the normal velocity will be *piecewise constant*, that is, constant on a given segment, but possibly varying from segment to segment. In particular, it may take one of two values; either zero, or a positive value which will be denoted β. The underlying notion is that the curve is propagating through an "active" medium in which it has a uniform, isotropic, normal velocity, β. This medium has imbedded in it inert inclusions, through which the curve cannot pass.

Corners in the curve may arise in three ways. First, shocks may form in the evolving curve, as described in Section 2.1. We will refer to these as active-active shocks or fans. Second, an active portion of the curve may encounter an inert inclusion, forming an active-inert shock or fan. Third, the inert inclusion may have a corner, which, as it becomes part of the evolving curve, forms an inert-inert shock. To derive the necessary expression for $\partial L/\partial t$, consider an arbitrary arrangement of such segments at time t. Then propagate each segment for small time Δt at the appropriate normal velocity, *ignoring the effect of the corners.* The result will be a series of segments that no longer meet at their endpoints. Note that the inert segments do not move. At this point the total $\Delta L/\Delta t$ will approximate that found by piecewise application of (5). The *shock correction terms* are found by reattaching the endpoints according to the following case-by-case rules. In all cases there is a direction associated with each segment, with the interior of the curve defined as lying to the left, and the velocity taken to be outward. Since the segments themselves are smooth, it suffices to consider them as straight lines in a sufficiently small neighborhood of the corner. The corner is characterized by an angle $\Delta\theta$, which is the jump in the orientation angle, moving along the curve in the positive sense. The shock correction terms all have the form $\beta\ \chi(\Delta\theta)$. The functions χ are listed below.

1. Inert-Active Shocks. In all cases of inert-active corners, it is assumed that the underlying boundary between inert and active material lies tangent to the inert segment in some neighborhood of the corner. Then, as can be seen in Fig. 1, for $0 < \Delta\theta \leq \pi/2$ or for $-\pi < \Delta\theta \leq -\pi/2$ a portion of the active segment is annihilated. The length of this portion can be seen to be,

$$\Delta L = \begin{cases} \beta\Delta t/\tan\Delta\theta; & 0 < \Delta\theta \leq \pi/2 \\ -\beta\Delta t/\tan\Delta\theta; & -\pi < \Delta\theta \leq -\pi/2 \end{cases}$$

   Therefore, the shock correction term is

$$\chi(\Delta\theta) = \begin{cases} -1/\tan\Delta\theta; & 0 < \Delta\theta \leq \pi/2 \\ 1/\tan\Delta\theta; & -\pi < \Delta\theta \leq -\pi/2 \end{cases}$$

   Although the figure shows an inert-active intersection, it is easily seen that an active-inert intersection yields the same shock correction terms. The shock correction terms derived above give the correction in the length of the *active* segment. Although similar expressions can be obtained for the inert segment, this is not necessary for our purposes.

2. Inert-Active Fans. When $\pi/2 < \Delta\theta < \pi$, or $-\pi/2 < \Delta\theta < 0$, then a gap opens between the two segments, which is filled with a *fan*. This is done using Huygens' construction, as shown in Fig. 2, yielding the following shock correction terms:

$$\chi(\Delta\theta) = \begin{cases} (\Delta\theta - \pi/2); & \pi/2 < \Delta\theta < \pi \\ (\Delta\theta + \pi/2); & -\pi/2 < \Delta\theta < 0 \end{cases}$$

   Again, only the corrections to the active segment will be needed.

3. Active-Active Shocks. When the angle between two active segments satisfies $-\pi < \Delta\theta < 0$, after some small time Δt the segments will overlap. Applying Huygens' construction, the overlapping segments are removed. The resulting total loss of length (that is, due to both segments) is seen from Fig. 3 to be $\Delta L = 2\beta\Delta t\tan(-\Delta\theta/2)$. The shock correction term is then,

$$\chi(\Delta\theta) = 2\tan(\Delta\theta/2).$$

4. Active-Active Fans. When the angle between two active segments satisfies $0 < \Delta\theta < \pi$, after some small time Δt a gap will open between the segments. Applying Huygens' construction to fill that gap, as in Fig. 4, gives a total increase in length of $\Delta L = \beta\Delta t\Delta\theta$. The shock correction term is

$$\chi(\Delta\theta) = \Delta\theta.$$

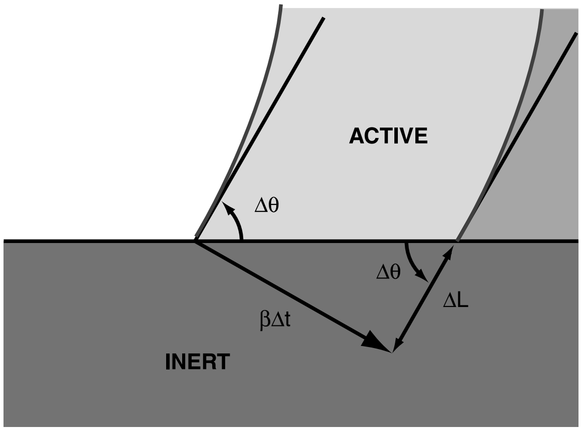

Figure 1: Inert-Active shock geometry

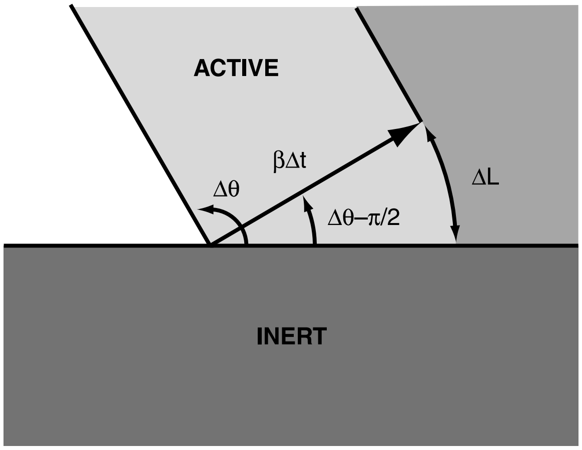

Figure 2: Inert-Active fan geometry

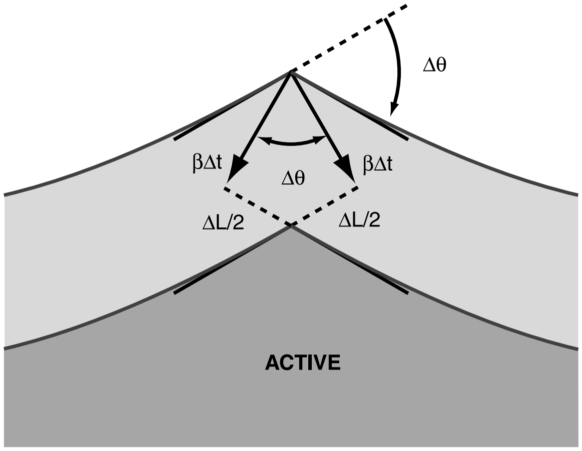

Figure 3: Active-Active shock geometry

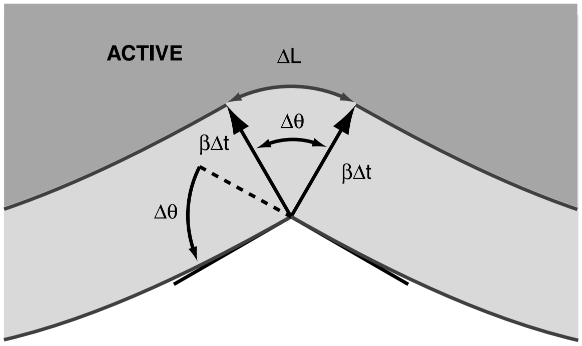

Figure 4: Active-Active fan geometry

5. Inert-Inert Shocks. These structures do not evolve, and hence require no shock correction term.

These formula are easy to calculate mathematically, but there is a major obstacle to their practical application. That is, it is quite tricky to compute the change in tangent direction, $\Delta\theta$, accurately. Although the level set function may be used to directly generate this information (through computation of the unit normal, which is simply $\nabla\Phi/\|\nabla\Phi\|$), these estimates are most precise when Φ is *smooth.* Of course, it is exactly when Φ is *non-smooth* that we require the values. In pursuing this research, we first applied a simple pixel-based algorithm for recovering the level set itself [2]. This technique is quite crude, and provides a fast, but rough sketch of the evolving feature. The next version of the algorithm used bilinear approximation in each grid cell to estimate the curve length, and, as suggested in [22], averaged the unit normal vector estimates at neighboring grid points to obtain the change in orientation. The resulting length estimate was reasonable, but the contour orientation was not satisfactory for our purposes near the shock points. This problem has been addressed in [21] by Siddiqi *et al.*, whose algorithms are applied in the following sections.

Siddiqi *et al.* compare techniques for recovering contours from a level set function, with subpixel resolution through interpolation. They develop a method that is similar to standard contour tracing algorithms found in computer vision applications (see, for example, [19]) with the addition of *shock placement* logic. The algorithm first detects zeros along the grid lines of the mesh. It then uses geometric interpolation based on lines, arcs, and Euler spirals (curves of linearly varying curvature) to approximate the entire contour. Threshholding logic determines whether the contour contains a shock. The scheme presented in [21] detects shocks by abrupt changes in the *orientation* or *curvature* of the contour. If two shocks occur in a row, they are "relieved" by placing a single new shock with subpixel resolution. The curves on either side of that single point will have smooth orientation and curvature. We refer to [21] and the references therein for more details. The version implemented here is similar, but only first order. That is, shocks are detected by a large change in orientation only, and when consecutive shocks are relieved by placing a single subpixel shock, the orientation is smoothed, but the curvature may not be. The algorithm presented in [21] is quite efficient once any point on a contour has been found. It can be computationally intensive, however, to find such starting points. As will be seen, that can be mitigated by exploiting certain side-benefits of our integration scheme.

For a certain class of curve evolutions, including the case of piecewise constant β, a special choice of level set function is possible. We motivate this choice by noting that the function Φ contains far more information than we actually require. Φ is propagated over the entire xy-plane, despite the fact that only a particular level set is desired. While "narrow band" numerical implementations have been designed that reduce the computational overhead, a more elegant approach is possible.

Consider the class of curves evolving so that the normal velocity never changes sign. Then once the curve has passed through a point, it never returns. Thus at each point we may assign a unique value, equal to the time at which the curve passed through it. If the curve never reaches that point, a value of ∞ may be used. The resulting *crossing-time* function $T(x, y)$

is in fact a time-invariant level set function. Note that the crossing-time function stores the complete evolution. No information is wasted, because only the level set values corresponding to the actual curve at some time are computed. The corresponding time-invariant governing PDE is

$$\beta\|\nabla T\| = 1 \tag{19}$$

This expression is obtained from the original formulation by defining $\Phi(x,t) := T(x) - t$. Sethian has presented a *fast marching* (FM) algorithm for evolving Eq. (19); see [23]. In addition to reduced storage requirements and greater speed, the FM algorithm has numerous advantages for implementation in a real-time estimator. In particular, FM works extremely well with the contour tracing approach to recovering orientation and curvature. As a by-product of the FM algorithm, we obtain a list of the grid points, sorted by crossing time. Then, when we must search through the grid to find new contours, corresponding to a particular level surface—that is, to the evolving curve at a particular time—it suffices to search only a small interval of the sorted points. This interval is centered on the desired time, and is on the order of h/β in either direction, where h is the mesh size, and β is the etch rate. Restricting the contour search to these points represents a significant savings in computation.

Each iteration of the method determines the crossing time of a single grid point. Thus the entire grid is solved in n iterations, where n is the total number of grid points. Each iteration involves a sort, and requires at most $O(\log n)$ operations, for a total operation count of $O(n \log n)$. The FM method is notable in that it does not have a time step, as such. However there is a lower limit on the time resolution of the method. In the case of the isotropic etch, it is $t_{\min} = h/\beta$. This must be be kept in mind when choosing a time interval at which to generate estimates. Given some fixed estimation interval, say Δt_{est}, the *minimum* etch rate that can be determined is $\beta_{\min} = h/\Delta t_{\text{est}}$. That is, the surface must advance at least one grid point between updates of the etch rate and surface estimate.

4 Estimation and Control of Etching

Semiconductor manufacturing processes are well-suited for the level set techniques described above. As in prairie fires, the entropy condition has a clear physical interpretation. For etching this is *material that is removed is never restored.* Some other features of level sets are the ease with which they can be extended from two space dimensions to three, and the potential for capturing effects like surface diffusion in a curvature term. These issues, and others, are discussed extensively in the work [9, 22, 24, 25].

Equations (18) or (19) are completely general descriptions of curve or surface evolution. The physics of the underlying process are entirely contained in the function β. So far, we have discussed cases where β is a function of curvature. A realistic description of manufacturing processes for thin-film devices may require that β have a more complicated functional dependence. One possibility is that it will depend on other quantities intrinsic to the interface, such as the direction of the unit normal, or derivatives of the curvature. These flows can be treated by similar methods. It also may occur that the interface normal velocity is determined in part by inherently non-local effects. Some non-local contributions that play a role in low pressure deposition and etching processes are treated by Adalsteinsson and Sethian [1, 22]. We have chosen to start with the simplest possible case—isotropic etching.

A number of studies have investigated incorporating feedback control into the plasma etching process [4, 15, 16, 20]. The focus of these studies has been on the plasma itself. Here we are interested in using *in situ* measurements in real-time to directly control the evolving features. This is a challenging task, but the potential benefits are large. Ultimately it is the surface morphology, rather than the plasma properties, that directly influences the performance of the finished device. The problem we address in this paper is more limited in scope—namely *estimating* the evolving shape of the surface features. Clearly this is a necessary predecessor to control. This work is completely complementary to other process control tasks, such as the control of plasma variables. One can imagine, for example, adjusting plasma inputs in response to changing etch rate estimates. Or, one might use the area of the active surface generated by the estimator as an input to the plasma model.

4.1 Isotropic Etching of a Long Trench

This section describes a highly simplified model of a plasma etching process. A uniform layer of silicon sits on an inert substrate. The silicon is masked with a thin layer of resist, except for one or more narrow gaps. At $t = 0$, a reactive substance, for example chlorine gas, is introduced at the surface. This substance etchs silicon, but not the resist or the substrate. The simplifying assumptions are as follows:

1. The feature to be etched is composed of lines, very long compared to their width. This allows a 2-D planar approximation.

2. A basic feature made up of only a few lines is repeated indefinitely along the wafer surface. Then only one such may be considered, with periodic boundary conditions.

3. The etch is isotropic. Although isotropic plasma etching is rare, we begin with this case because an analytical solution is available as a truth model.

4. The etch rate is constant and homogeneous within the "active" material.

5. The mask and substrate are perfectly inert, with an etch rate of zero.

It remains to define an appropriate measurement. Preliminary experimental work using a chlorine/argon plasma and a crystalline silicon wafer suggests that the concentration of chemical by-products to the etch can be measured via optical emission spectroscopy. This is the signal that we propose to use for reconstructing the etch rate and feature evolution. Ultimately some modeling of the plasma/chamber system will be required. For the purposes of this preliminary study we use the following simplification: Assume that the rate of change of the total etching by-products into the plasma is known, and that this rate is due solely to removal of material from the wafer. Then (see Fig. 5) the measurement, y, is related to the feature geometry and etch rate by $y(t) := \int_{X(t)} \chi_e \beta dl$, that is, the etch rate integrated over the active surface. The characteristic function χ_e is needed to eliminate from the integral

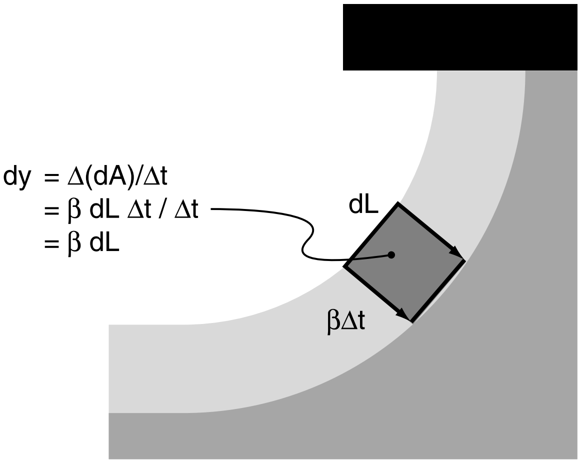

Figure 5: Measurement geometry.

those parts of the surface that are masked or inert. Note that it is impossible to relate y to the etch rate without knowledge of the feature shape. A complete model of the system is then,

$$\beta\|\nabla T\| = 1 \tag{20}$$

$$\{X(t) \in \mathbf{R}^2 \, : \, T(X) = t\} \tag{21}$$

$$y(t) := \beta \int_{X(t)} \chi_e dl \tag{22}$$

Under the above assumptions, the predicted measurement is just $\beta L(t;\beta)$ where $L(t;\beta)$ is the length of the estimated surface, and the dependence of the time history of L on β is explicitly indicated. It is also understood that $L(t;\beta)$ refers only to the portion of the surface that is exposed silicon, not resist or substrate.

An exact solution to the feature evolution question is easily obtained for a single-trench etching using the Huygens principle. Figure 6 depicts the feature geometry. Then the corresponding measurement is given as follows:

$$y(t) = \begin{cases} \beta L_0 + \pi\beta^2 t; & \beta t \in [0, H) \\ 2\beta^2 t \sin^{-1}(H/\beta t); & \beta t \in [H, W) \\ 2\beta^2 t[\sin^{-1}(H/\beta t) - \cos^{-1}(W/\beta t)]; & \beta t \in [W, (H^2+W^2)^{1/2}) \\ 0; & \beta t \geq (H^2+W^2)^{1/2} \end{cases} \tag{23}$$

where H and W are as shown in Fig. 6. For $\beta = 1$, Fig. 7 compares this exact measurement to the values predicted by a level set simulation with a mesh size of 0.25. Figure 8 compares the resulting surface evolution. Figures 9 and 10 make the same comparisons, but with a mesh size of 0.1. The decrease in mesh spacing gives a notable increase in accuracy, both in the predicted measurement and the evolving feature.

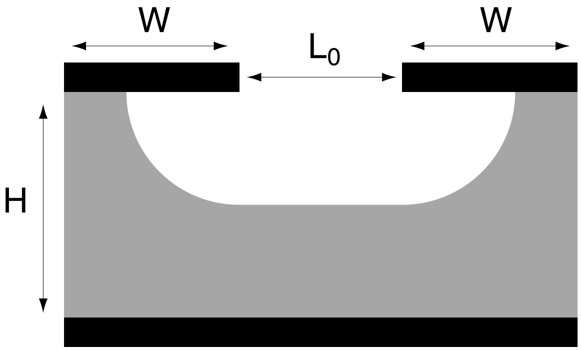

Figure 6: One-trench etching geometry.

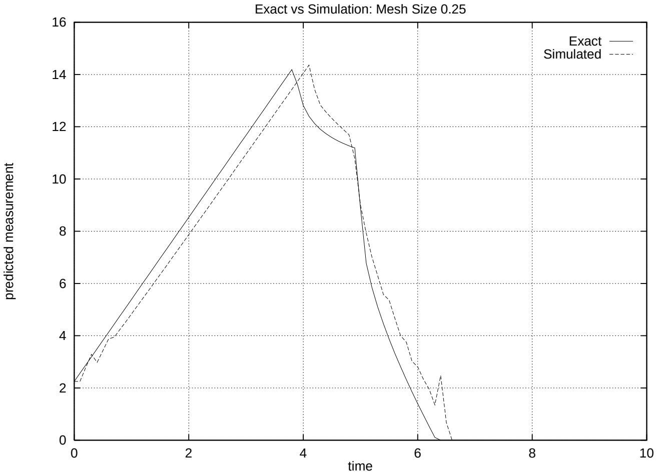

Figure 7: Exact and simulated output; $h = 0.25$

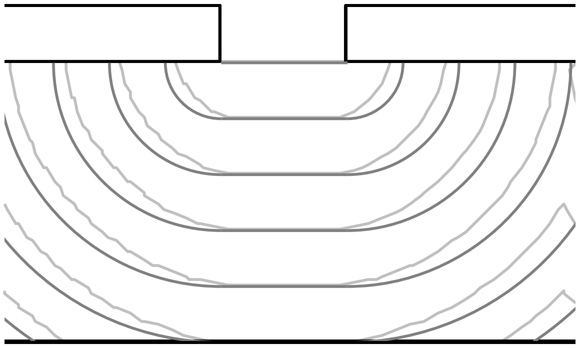

Figure 8: Exact and simulated feature evolution; $h = 0.25$. Light: Simulated; Dark: Exact.

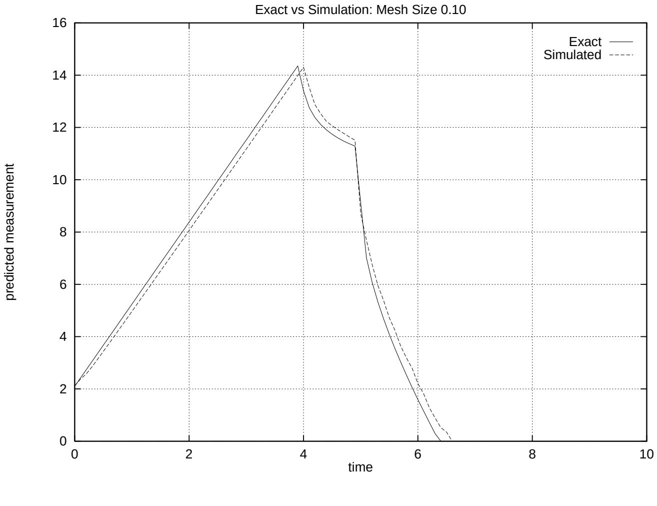

Figure 9: Exact and simulated output; $h = 0.10$

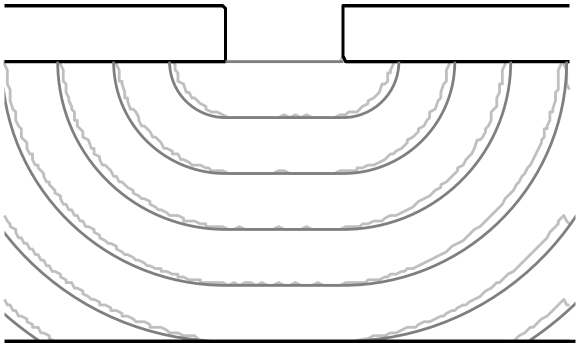

Figure 10: Exact and simulated feature evolution; $h = 0.10$. Light: Simulated; Dark: Exact.

The remainder of this paper concerns the construction of an estimator that will track the evolving feature morphology, based on the total etch rate measurement. The overall strategy is shown in Fig. 11. The level set simulation plays the role of the plant model, and the etch rate β is used as an adjustable parameter. We assume that the initial geometry is known exactly, but that the etch rate, though constant or slowly varying, is not known. The box labeled "ID" in Fig. 11 represents some algorithm that solves the inverse problem of "best" matching an etch rate estimate to the measured data. For our present purpose, "best" will be in a least squares sense. The estimated etch rate is then used to propagate the estimated feature.

We express the objective as a minimization problem,

$$\begin{aligned} \min_{\beta} J(\beta) &= \min_{\beta} \frac{1}{2} \sum_i (\beta L(t_i;\beta) - y(t_i))^2 && (24) \\ &= \min_{\beta} \frac{1}{2} R(\beta)^t R(\beta) && (25) \end{aligned}$$

where $[R(\beta)]_i = \beta L(t_i;\beta) - y(t_i)$. The goal of this study is real-time estimation of the evolving feature. This requires that the minimization problem be solved in the most efficient possible manner. Generally, efficient solution of minimization problems requires analytical expressions for the first derivative of the cost function with respect to the unknown parameters. Here that is not straightforward, because it is not obvious how to take derivatives through the set operation in (21). We now show how the necessary expressions can be obtained.

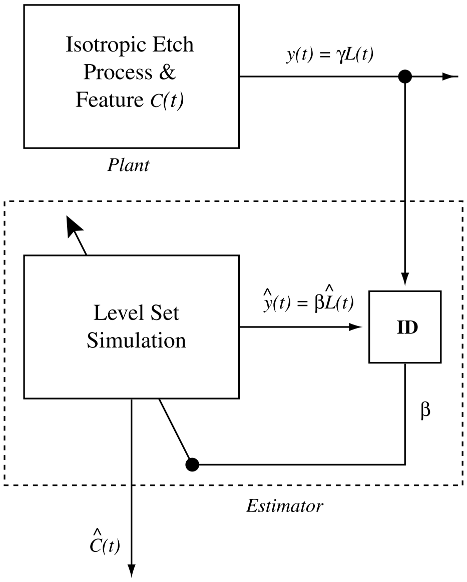

Figure 11: Estimator Structure

The derivative $J_\beta(\beta)$ is

$$J_\beta(\beta) = R(\beta)^T \nabla R(\beta) \tag{26}$$

The necessary derivative is given by,

$$[\nabla R(\beta)]_i = L(t_i;\beta) + \beta L_\beta(t_i;\beta). \tag{27}$$

At any time t_i, given some value of β, $L(t_i;\beta)$ can be found using a forward solve. It then remains only to find $L_\beta(t_i;\beta)$.

Recall that in Section 2 we presented an analytical expression for $\partial L/\partial t$,

$$\begin{aligned} L_t(t;\beta) &= \beta \int_0^1 \kappa \rho ds + \beta \sum_{\text{shocks}} \chi & (28)\\ &= \beta K(t;\beta) + \beta X(t;\beta) & (29)\end{aligned}$$

where $X := \sum_{\text{shocks}} \chi$. We make the following observation: The shape of the evolving feature may be expressed as a function of the variable $R := \beta t$ only. This may be seen from Huygens' principle, which states that the front is the envelope of the set of circles with radii R centered on the initial active surface. Thus we can write $L = L(R)$, $K = K(R)$, and $X = X(R)$, and $L_t = L'(R)R_t = L'(R)\beta$. Comparison with Eq. (28) gives $L'(R) = K(R) + X(R)$. Finally, $L_\beta = L'(R)R_\beta = L'(R)t$, or

$$L_\beta(t;\beta) = (K(R) + X(R))t = (K(\beta t) + X(\beta t))t. \tag{30}$$

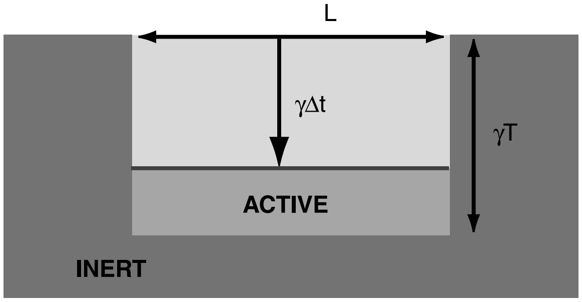

Figure 12: Planar etch.

For any time t and etch rate β the value of $L_\beta(t;\beta)$, and therefore $J_\beta(\beta)$, can be found from Eq. (30) via a forward solve.

The choice of numerical method for minimizing Eq. (24) is critical. With an expression for the first derivative available almost everywhere, it is tempting to apply a standard approach such as Gauss-Newton. However, these cost functions are not sufficiently smooth for use of Hessian-based algorithms. To see this, consider an extremely simple planar etch geometry, as shown in Fig. 12. Here, due to the shape of the inert inclusions, the evolving feature is a line segment of constant length L. The corresponding measurement at any time is simply $y = \gamma L$, where γ is the true etch rate. Let the etch go to completion, that is, until the measured signal is zero. Let the time required for this to occur be denoted T. For an etch rate estimate β, the cost function at completion is

$$2J(\beta) = \begin{cases} \int_0^{M/\gamma} L(\beta-\gamma)^2 dt + \int_{M/\gamma}^{M/\beta} L\beta^2 dt; & \beta < \gamma \\ 0; & \beta = \gamma \\ \int_0^{M/\beta} L(\beta-\gamma)^2 dt + \int_{M/\beta}^{M/\gamma} L\gamma^2 dt; & \beta > \gamma \end{cases} \tag{31}$$

$$= \begin{cases} LM(\gamma-\beta); & \beta < \gamma \\ 0; & \beta = \gamma \\ LM(\beta-\gamma); & \beta > \gamma \end{cases} \tag{32}$$

$$= LM|\beta-\gamma| \tag{33}$$

where $M := L\gamma T$ is the total area of material to be etched. The cost function is not smooth at the solution, and algorithms approximating the second derivative will give no insight into the minimal point, even very near the solution. The isotropic etch exhibits similar behavior. Fig. 13 is a plot of the square root of J vs. β. J is calculated by numerically integrating Eq. (24), where the etch rate estimate is generated by Eq. (23) with an etch rate estimate of β, and the measurement is also given by Eq. (23), but with true the etch rate $\gamma = 1$. Again the cost function shows a corner at the solution. However, it is still possible to use the information contained in our calculation of the first derivative.

Figure 13 suggests that J is smooth away from the solution, and that there are no other local minimizers in a broad range of etch rates. Therefore J decreases in the direction of

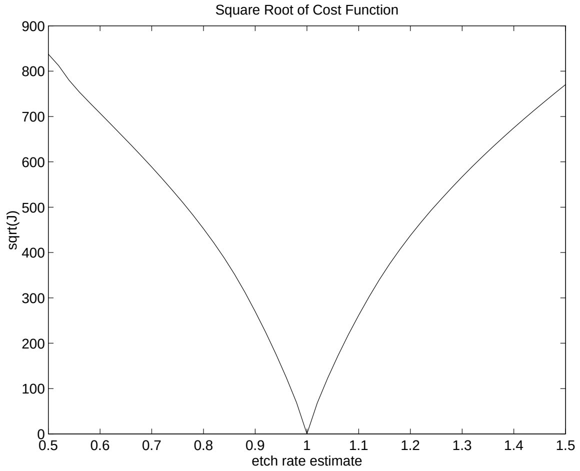

Figure 13: Square root of J vs. β, with $\gamma = 1$

the solution, and the sign of J_β changes only across the true etch rate. Then for the present case, where the parameter space is one-dimensional, and the parameter must be positive, the following algorithm works well:

1. Make an initial guess at the etch rate $\beta^{(i)} = \beta_0$.
2. Evaluate $J_\beta^{(i)} := J_\beta(\beta^{(i)})$ for the current etch rate estimate.
3. If $J_\beta^{(i)}$ is negative, set $\beta^{(i+1)} = k\beta^{(i)}$, if $J_\beta^{(i)}$ is positive, set $\beta^{(i+1)} = \beta^{(i)}/k$, where $k > 1$.
4. Loop to (2) until $J_\beta^{(i)}$ changes sign.
5. When $J_\beta^{(i)}$ changes sign, the solution has been bracketed. Proceed by bisection on J_β, as though searching for a root of J_β.
6. When the bisection interval is sufficiently small, return the midpoint as the etch rate estimate.

We remark that the behavior inferred from Fig. 13 is based on frequent sampling of the measurement y. (Essentially turning the summation of Eq. (24) into an integral. Infrequent sampling may introduce spurious local minima.

It must be noted that Eq. (30) is correct at *almost every* t, but that there may be isolated times at which it breaks down. For example, if a section of active curve contacts an inert inclusion *parallel* to it, a segment of finite length is removed in an infinitesimally small time interval. To see the effect of such an event, write the expression for the surface length, with

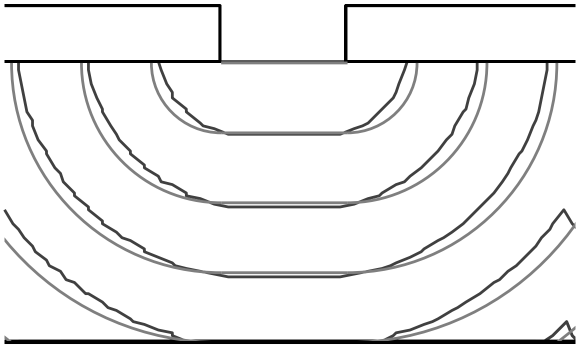

Figure 14: Feature evolution: estimated vs true, $h = 0.25$. Dark: Estimated; Light: Exact.

these step changes included.

$$L(R) = L_0 + \int_0^R [K(u) + X(u)]du + \sum_i l_i u(R - R_i) \tag{34}$$

where u is the unit step function, the i^{th} step change has magnitude l_i and occurs at a value of R equal to R_i. Then the partial derivative is,

$$L_\beta(t; \beta) = L'(R)t = [K(R) + X(R)]t + \sum_i l_i \delta(R - R_i)t \tag{35}$$

So at such points, the sensitivity fails to exist. The problem can be avoided by removing these data points. This is not a particularly satisfactory way of addressing this problem. First one must define some automatic logic to detect this situation, next one must discard information that could be profitably exploited. A better approach will be the topic of future inquiries.

4.2 Simulation Results

Several cases were simulated. For a single trench, the exact solution is known, and is given by Eq. (23). Recall that the success with which the FM method simulated the surface motion depended on the fineness of the mesh. The exact solution was used to generate measurement data, and the algorithm of the preceding section applied. In these simulations the height of the active layer is 5 units, and the width of the periodic cell is 10 units. The true etch rate was set to 1.25, and the initial guess given to the estimator was one. The etch rate was estimated

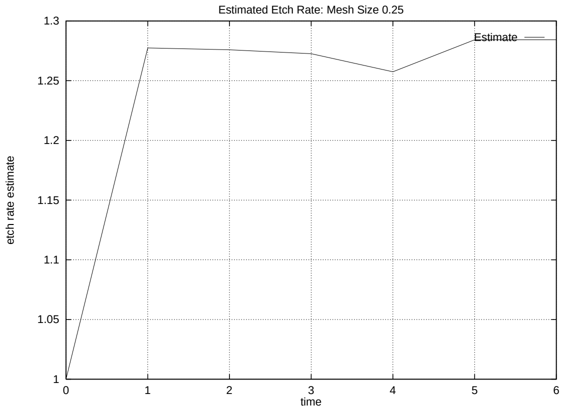

Figure 15: Etch rate estimate, $h = 0.25$. True value 1.25.

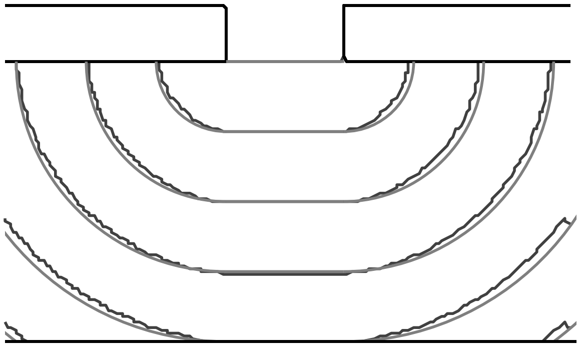

Figure 16: Feature evolution: estimated vs true, $h = 0.10$. Dark: Estimated; Light: Exact.

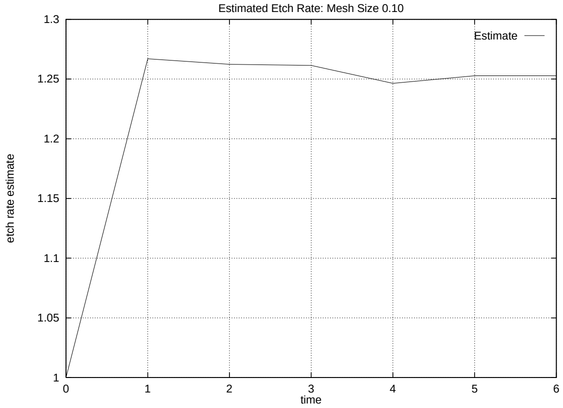

Figure 17: Etch rate estimate, $h = 0.10$. True value 1.25.

at intervals of 1 time unit, and the corresponding surface drawn. The process was carried out for an estimator mesh size of 0.25, and of 0.1. Figures 14-17 show the results compared to the exact solution.

With more complex geometries, Eq. (23) no longer gives the exact solution (although such an expression could be derived). Figure 18 shows a mask pattern which repeats after three trenches. Again, the height of the active layer is 5 units, but the periodic cell now has a width of 20 units. In this case the truth model used is an FM simulation with a mesh size of 0.1. Again, a true etch rate of 1.25 is used, and an initial guess of 1.0 is supplied to the estimator, which updates an etch rate estimate at 0.5 time unit intervals. Figures 19-22 show the results, first for an estimator mesh size of 0.25, then for a mesh size of 0.1. Here the "true" solution is simulated on a grid with a mesh size of 0.1. This is also the source of the measurements. The results seem very promising. When the simulator is incapable of recovering the true feature shape, then of course the estimator cannot either. But even in those cases the etch rate estimate is quite reasonable. When the mesh is sufficiently fine, then both the etch rate and the feature shape are obtained. The effects of noise have yet to be considered.

4.3 Computation Time

This work has been motivated by the desire to implement surface evolution models in real time. The results to this point indicate that this is feasible. For example, in the three-trench problem described in the previous section, the estimator using a mesh size of 0.1—corresponding to a computational grid of dimension 200 by 100—required approximately 15 forward solves to converge to a solution. This required under 20 seconds on a 333 MHz DEC AlphaStation 500. This compares favorably with plasma etch times, which may be on the order of several minutes, or more. Thus the algorithm presented above, with the simple geometries considered here, is fast enough for process monitoring, with the possibility of control. The critical question is whether a relatively small 2-D grid is sufficient to represent feature geometries

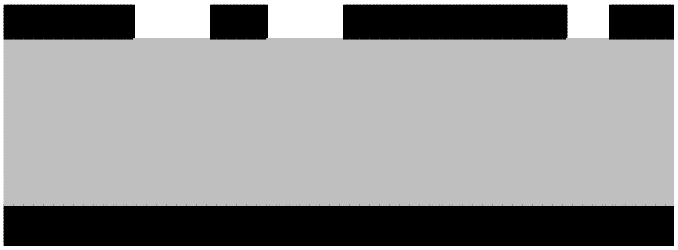

Figure 18: Three-trench feature evolution: feature geometry

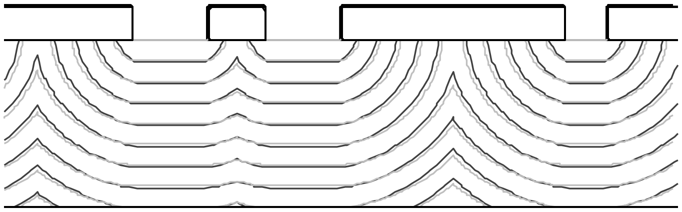

Figure 19: Three-trench feature evolution: estimated vs true, $h = 0.25$. Light: Estimate; Dark: Truth model.

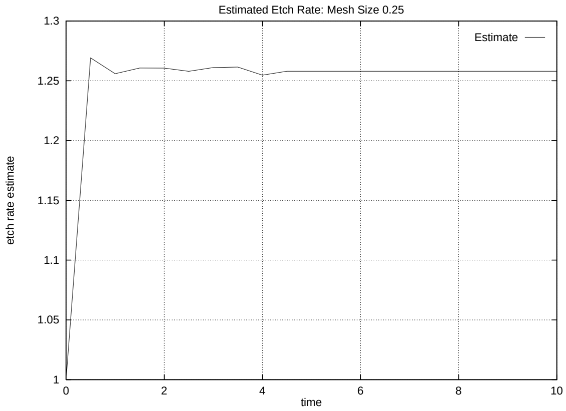

Figure 20: Three-trench etch rate estimate, $h = 0.25$. true value 1.25.

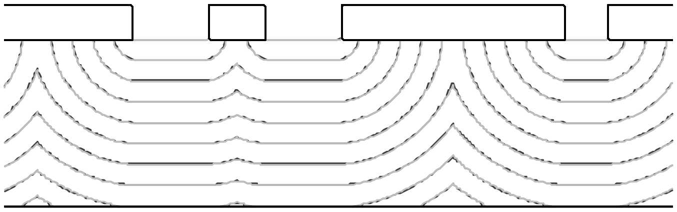

Figure 21: Three-trench feature evolution: estimated vs true, $h = 0.10$. Light: Estimate; Dark: Truth model.

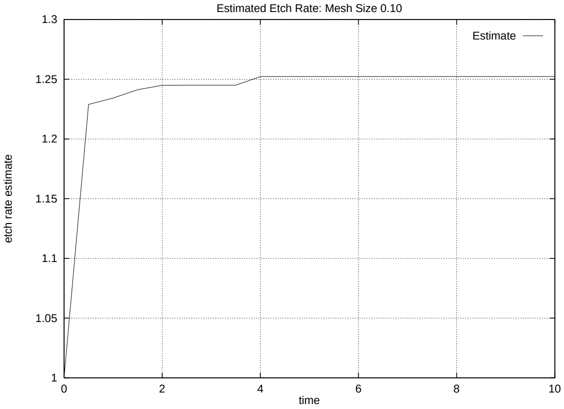

Figure 22: Three-trench etch rate estimate, $h = 0.10$. True value 1.25.

of interest. Implementing more elaborate (and realistic) features will require larger grids, which will increase computation times. Also, the approach presented above can be extended to 3-D grids, but the number of points, and therefore the computation time, will increase dramatically.

The code used to generate these examples could be significantly streamlined. Indeed, we do not claim that we have optimized our code in any way, and in fact we believe that with further refinements the computation times can be somewhat decreased. For example, it appeared that only 8 iterations per estimate would have sufficed in the three-trench example. On the other hand, even with these early versions, the results are very encouraging.

5 Extensions and Conclusions

As a final note, we comment on two interesting extensions of the inverse problem presented above. First, we show that it is possible to calibrate the measurement *simultaneously* with the etch rate estimate.

The measurement $y(t)$ was assumed to be equal to the rate of material removal of the etch. In fact, the measurement is only *proportional* to the total etch rate; and the constant of proportionality must be determined in a separate calibration step. It is desirable in an industrial setting to minimize such steps, which add downtime and increase operating costs. Denote the constant of proportionality by α. Then the estimated measurement is given by $\alpha\beta L(t;\beta)$. To see that this constant can be estimated *simultaneously* with β, consider the exact solution for the first stage of the etch, $y(t) = \alpha\beta L(t;\beta) = \alpha\beta(L_0 + \beta\pi t)$, with α and β unknown. The measurement will be a straight line, $y(t) = A + Bt$. Then A and B can be found by a simple linear regression, and α and β will be given by, $\beta = BL_0/A\pi$ and $\alpha = A/\alpha L_0$.

Next, we note that there is an alternative to Eq. (30) for calculating the sensitivity. The length of the etching surface may be written as an integral over the entire domain Ω [28, 22],

$$L(t) = \int_\Omega \chi_e \delta(T(x,y) - t)\|\nabla T\| dA \tag{36}$$

where δ is the delta function. Here the β dependence is entirely contained in the expression for T. Unlike Eq. (22), differentiating this expression for L does not require direct use of the set operation Eq. (21). However, the derivative with respect to β will involve T_β and its spatial derivatives. These may be found via a *sensitivity equation* approach (see [3], for example). In such an approach the sensitivity function $\Psi := T_\beta$ is found by solving an additional coupled PDE,

$$\|\nabla T\|^2 + \beta\langle\nabla T, \nabla\Psi\rangle = 0 \tag{37}$$

The success of level curve techniques in simulating etching and deposition processes suggests their potential for the estimation and control of these processes as well. This paper has presented a simple, but still practical, application to surface shape estimation during isotropic etching of a trench. The methods presented draw on the geometric theory of curve evolution.

The wealth of elegant mathematics available in that area has barely been touched here, and will be the subject of our future research.

References

[1] D. Adalsteinsson and J. A. Sethian, "A level set approach to a unified model for etching, deposition, and lithography I: algorithms and two-dimensional simulations," *Journal of Computational Physics* **120**:128–144, 1995.

[2] J. M. Berg, A. Yezzi, and A. R. Tannenbaum, "Phase transitions, curve evolution, and the control of semiconductor manufacturing processes" *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 3376–3381, Kobe, Japan, 1996.

[3] J. Borggaard and J. Burns, "A sensitivity equation approach to shape optimization in fluid," in *Flow Control*, M.D. Gunzburger, ed., Springer-Verlag, New York, 1995.

[4] S. W. Butler, K. J. McLaughlin, T. F. Edgar, and I. Trachtenberg, "Development of techniques for real-time monitoring and control in plasma etching, II: multivariable control system analysis of manipulated, measured, and performance variables," *J. Electrochem. Soc.* **138**:9, pp. 2727-2735, 1991.

[5] M. P. do Carmo, *Differential Geometry of Curves and Surfaces,* Prentice-Hall, New Jersey, 1976.

[6] L. C. Evans and J. Spruck, "Motion of Level Sets by Mean Curvature. 1," *Journal of Differential Geometry*, 33, pp. 635–681, 1991.

[7] M. Gage and R. S. Hamilton, "The heat equation shrinking convex plane curves," *J. Differential Geometry* **23**, pp. 69-96, 1986.

[8] M. Grayson, "The heat equation shrinks embedded plane curves to round points," *J. Differential Geometry* **26**, pp. 285-314, 1987.

[9] I. Katardjiev, G. Carter, and M. Nobes, "The application of the Huygens principle to surface evolution in inhomgeneous, anisotropic and time-dependent systems," *J. Phys. D: Appl. Phys.* **22**, pp. 1813–1824, 1989.

[10] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzi, "Conformal curvature flows: from phase transitions to active contours," *Archive for Rational Mechanics and Analysis* **134**, pp. 275–301, 1996. pp. 275–301.

[11] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deformations, I," *Int. J. Computer Vision*, June 1995.

[12] B. B. Kimia, A. Tannenbaum, and S. W. Zucker, "On the evolution of curves via a function of curvature, I: the classical case," *J. of Math. Analysis and Applications* **163**, pp. 438-458, 1992.

[13] M. E. Law, "Grid Adaption Near Moving Boundaries in Two Dimensions for IC Process Simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **14**:10, pp. 1223–1230, 1995.

[14] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser, Boston, 1992.

[15] K. J. McLaughlin, S. W. Butler, T. F. Edgar, and I. Trachtenberg, "Development of Techniques for Real-Time Monitoring and Control in Plasma Etching, I: Response Surface Modeling of CF_4/O_2 and CF_4/H_2 Etching of Silicon and Silicon Dioxide," *J. Electrochem. Soc.* **138**:3, pp. 789-798, 1991.

[16] K. J. McLaughlin, T. F. Edgar, and I. Trachtenberg, "Real-Time Monitoring and Control in Plasma Etching," *IEEE Control Systems*, pp. 3-10, April, 1991.

[17] S. Osher, "Riemann solvers, the entropy condition, and difference approximations," *SIAM J. Numer. Anal.* **21**, pp. 217-235, 1984.

[18] S. J. Osher and J. A. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics* **79**, pp. 12-49, 1988.

[19] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, 1982.

[20] B. A. Rashap, M. E. Elta, H. Etemad, J. P. Fournier, J. S. Freudenberg, M. D. Giles, J. W. Grizzle, P. T. Kabamba, P. P. Khargonekar, S. Lafortune, J. R. Moyne, D. Teneketzis, and F. L. Terry, "Control of semiconductor manufacturing equipment: real-time feedback control of a reactive ion etcher," *IEEE Trans. Semiconduct. Manufacturing* **8**:3, pp. 286-297, 1995.

[21] K. Siddiqi, B. Kimia, C-W. Wang. *Geometric Shock-Capturing ENO Schemes for Subpixel interpolation, Computation, and Curve Evolution*, Technical Report LEMS-142, Brown university, February, 1995.

[22] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1996.

[23] J. A. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts,"*Proc. Nat. Acad. Sci.***93**:4, pp. 1591–1595, 1996.

[24] E. Shaqfeh and C. Jurgensen, "Simulation of reactive ion etching pattern transfer," *J. Appl. Physics***66**, pp. 4664–4675, 1989.

[25] V. Singh, E. Shaqfeh, and J. McVittie, "Simulation of profile evolution in silicon reactive ion etching with re-emission and surface diffusion," *J. Vac. Technol. B* **10**, pp. 1091–1104, 1989.

[26] J. Smoller, *Shock Waves and Reaction-diffusion Equations*, Springer-Verlag, New York, 1983.

[27] G. A. Sod, *Numerical Methods in Fluid Dynamics*, Cambridge University Press, Cambridge, 1985

[28] M. Sussman, P. Smereka, and S. Osher, "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow," *Journal of Computational Physics*, **114**, pp. 146–159, 1994.

[29] A. Srinivasan, C. Batur, B. N. Rosenthal, and W. M. B. Duval, "Solid-Liquid Interface Shape Control During Crystal Growth," *Proc. of the American Control Conference*, Seattle, WA, pp. 1270–1274, 1995.