

MODELING, SIMULATION, AND DESIGN FOR A CUSTOMIZABLE ELECTRODEPOSITION PROCESS*

PRADEEP THIYANARATNAM[†], RUSSEL CAFLISCH[‡], PAULO S. MOTTA[§], AND
JACK W. JUDY[¶]

Abstract. Judy and Motta developed a customizable electrodeposition process for fabrication of very small metal structures on a substrate. In this process, layers of metal of various shapes are placed on the substrate, then the substrate is inserted in an electroplating solution. Some of the metal layers have power applied to them, while the rest of the metal layers are not connected to the power initially. Metal ions in the plating solution start depositing on the powered layers and a surface grows from the powered layers. As the surface grows, it will touch metal layers that were initially unpowered, causing them to become powered and to start growing with the rest of the surface. The metal layers on the substrate are known as seed layer patterns, and different seed layer patterns can produce different shapes. This paper presents a mathematical model, a forward simulation method, and an inverse problem solution for the growth of a surface from a seed layer pattern. The model describes the surface evolution as uniform growth in the direction normal to the surface. This growth is simulated in two and three dimensions using the level set method. The inverse problem is to design a seed layer pattern that produces a desired surface shape. Some surface shapes are not attainable by any seed layer pattern. For smooth attainable shapes, we present a computational method that solves this inverse problem.

Key words. electrodeposition, design, simulation, level set method

AMS subject classifications. 34A65, 35R35

DOI. 10.1137/070706501

1. Electrodeposition. Judy and Motta [7] developed a customizable electrodeposition process for fabrication of small metal structures on a substrate. As pictured schematically in Figure 1.1, two plates of metal (nickel) are placed in an electroplating solution, and voltage is applied across the two plates. This causes Ni ions to separate from the anode and deposit on the cathode. By insulating areas of metal on the cathode, they can control where the metal gets deposited. Altering the exposed metal allows them to produce different shapes. The patterns of exposed metal are called seed layer patterns. For complicated seed layer patterns, it may be difficult to predict the final shape of the object. Thus the experimental procedure may require some trial and error to determine the seed pattern that attains a desired shape. One application of this process is the fabrication of neural probes used to stimulate the brain in Parkinson's disease research [6]. When used in preclinical experiments involving small animal models (e.g., rat or mouse), the probes should be very small (e.g., a few 100 μm) so they can be inserted into the brain with precision and minimal

*Received by the editors October 26, 2007; accepted for publication (in revised form) March 19, 2008; published electronically January 28, 2009.

<http://www.siam.org/journals/siap/69-4/70650.html>

[†]Level Set Systems, Los Angeles, CA 90272 (pthiyana@gmail.com).

[‡]Department of Mathematics, University of California, Los Angeles, CA 90095-1555 (caflisch@math.ucla.edu). The research of this author was supported by ARO through grant DAAD19-02-1-0336 and by the National Science Foundation (NSF) through grant DMS-0707557.

[§]Innovative Micro Technology, Goleta, CA 93117 (pmotta@ieee.org).

[¶]Electrical Engineering Department, University of California, Los Angeles, CA 90095-1594 (jack.judy@ucla.edu). The research of this author was supported in part by the NSF-supported UCLA Neuroengineering IGERT (grant 9972802) and the National Institutes of Health (NIH)-funded UCLA Morris K. Udall Center of Excellence for Parkinson's Disease Research (grant NS38367).

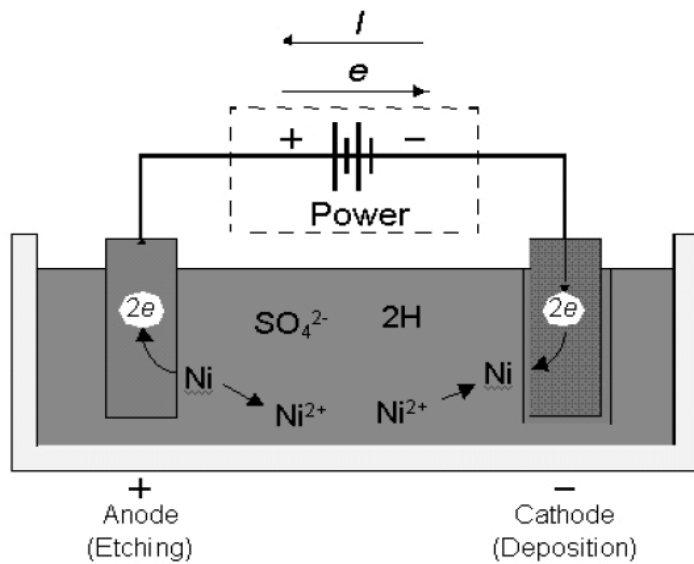


FIG. 1.1. *Experimental apparatus for plating a metal surface.*

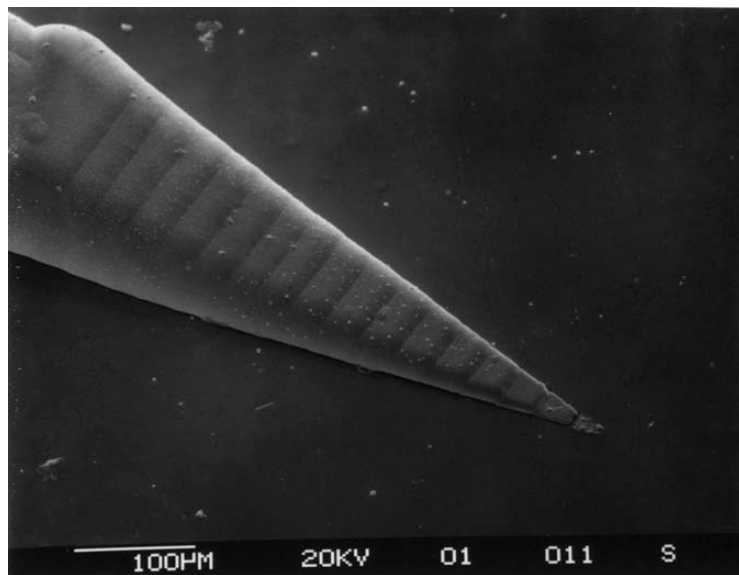


FIG. 1.2. *Experimentally formed micromachined probe shaft using a customizable electroplating process [7].*

damage [7]. Figure 1.2 shows an example of a needle-shaped object produced in [7].

A mathematical model of electrodeposition may simplify and accelerate the design of seed layer patterns and reduce the number of experimental trials needed to attain a desired shape. Our model will employ the simplifying assumption that the surface grows uniformly in the normal direction. We will not consider nonuniform growth or possible diffusion of the growing surface. Experimental parameters, such as

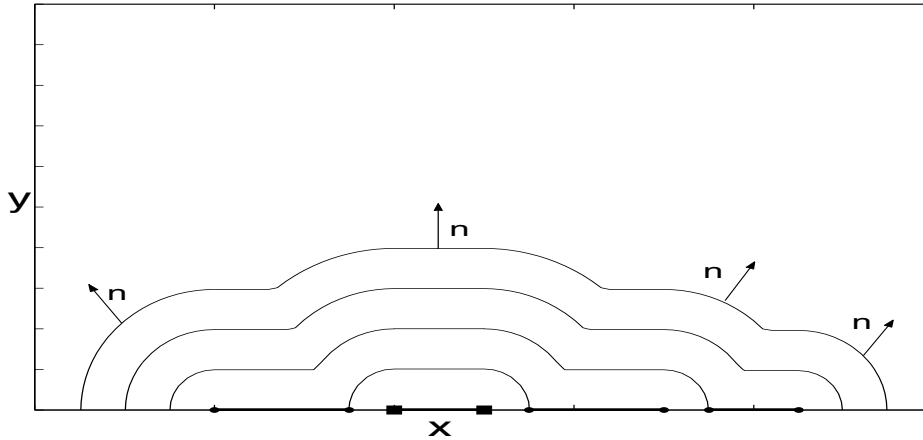


FIG. 1.3. Stages of growth of the seed layer pattern. The segment with squares on the ends is powered and the segments with circles on ends are unpowered.

temperature, will be also be ignored. These simplifications allow for a solution which is computationally fast and enable us to solve the inverse problem of determining a seed layer pattern that will produce a given shape.

Figure 1.3 shows a two-dimensional (2D) cross section of an object at various times as it grows uniformly in the normal direction, and indicates how the seed layer pattern controls the final shape. Initially, the segment with squares on the ends is a powered metal layer and the segments with circles on the ends are metal layers that are isolated from the power source. Although the object grows from the powered segment, it eventually touches an unpowered segment, which then becomes powered and starts to grow with the surface. As each unpowered segment is contacted by the growing object, it becomes powered and starts to grow.

The remainder of this paper is organized as follows: Section 2 describes the level set method that is the principal computational technique of our simulation method. Sections 3 and 4 describe the simulation method in two and three dimensions, respectively. The inverse design method in two and three dimensions is presented in sections 5 and 6, respectively. The methods for two and three dimensions are presented separately, since the 2D method is considerably simpler than the three-dimensional (3D) method. Conclusions and further work are discussed in section 7.

2. Level set method. The level set method is a way to represent complex geometric interfaces or shapes that evolve in time, with several nice properties, including the capability to easily merge shapes. The idea behind the level set method is to represent the interface (boundary) $\Gamma(t)$ of the object $\Omega(t)$ implicitly as the zero level set of a function ϕ , usually referred to as a level set function [8] (i.e., $\Gamma(t) = \{\mathbf{x} : \phi(\mathbf{x}, t) = 0\}$). We also set $\phi(\mathbf{x}, t) < 0$ inside $\Omega(t)$, and $\phi(\mathbf{x}, t) > 0$ outside $\Omega(t)$. This representation of $\Gamma(t)$ allows complicated topological changes, provides information about the interface, and makes computation of the evolving interface straightforward. For velocity \mathbf{v} of the boundary $\Gamma(t)$, the level set function ϕ describes the evolution of $\Gamma(t)$ if ϕ satisfies

$$(2.1) \quad \phi_t + \mathbf{v} \cdot \nabla \phi = 0,$$

$$(2.2) \quad \{\mathbf{x} : \phi(\mathbf{x}, 0) = 0\} = \Gamma(0),$$

with the additional condition that ϕ is negative inside $\Omega(0)$ and positive outside $\Omega(0)$.

In the case of electrodeposition, the object $\Omega(t)$ grows normal to itself at a constant speed (i.e., the velocity is $\mathbf{v} = \nu \mathbf{n}$, in which ν is a constant and \mathbf{n} is the outward unit normal vector to Γ). By rescaling time, we may take $\nu = 1$. Since the gradient $\nabla\phi$ of ϕ is in the outward normal direction \mathbf{n} , then $\mathbf{v} \cdot \nabla\phi = |\nabla\phi|$ and the PDE for ϕ becomes

$$(2.3) \quad \phi_t + |\nabla\phi| = 0.$$

A solution of (2.3) satisfying (2.2) is the function

$$(2.4) \quad \phi(\vec{x}, t) = d(\vec{x}) - t,$$

where $d(\vec{x})$ is the signed distance function for the initial object $\Omega_0 = \Omega(0)$ with boundary Γ_0 . The function $d(\vec{x})$ is defined as

$$(2.5) \quad d(\vec{x}) = \begin{cases} -\min_{\vec{s} \in \Gamma_0} |\vec{x} - \vec{s}|, & x \in \Omega_0, \\ 0, & x \in \Gamma_0, \\ \min_{\vec{s} \in \Gamma_0} |\vec{x} - \vec{s}|, & x \notin \Omega_0, \end{cases}$$

and satisfies $|\nabla d| = 1$. It follows that $|\nabla\phi| = 1$ and $\phi_t = -1$ and that the level set $\phi = 0$ is the boundary Γ_0 , so that ϕ solves (2.3).

For this model, it will be necessary to take the union of two objects and merge their boundaries. This can be done as follows: Let ϕ_1 and ϕ_2 be level set functions representing the interfaces $\Gamma_1 = \partial\Omega_1$ and $\Gamma_2 = \partial\Omega_2$, respectively. Then $\phi = \min(\phi_1, \phi_2)$ is a level set function representing the merged interface $\Gamma = \partial\Omega$ of the combined object $\Omega = \Omega_1 \cup \Omega_2$. It is important to note that merging two signed distance functions d_1 and d_2 will produce a level set function that is not necessarily a signed distance function (i.e., if $\phi = \min(d_1, d_2)$); then it is possible that $|\nabla\phi| \neq 1$ at some points inside the merged object. However, $|\nabla\phi(\vec{x})| = 1$ for \vec{x} in the exterior of the interface (i.e., where $\phi(\vec{x}) > 0$). Since the interface is moving in the outer normal direction, the interface motion is still correctly described by the equation $\phi_t + 1 = 0$.

Previous applications of the level set method are found in [5, 9] for electrodeposition and in [1, 2, 3, 4] for more general deposition and materials processing.

3. Forward problem in 2D.

3.1. General idea of the solution for the forward 2D problem. In the 2D case, the domain is $[a_1, a_2] \times [0, b_2]$. The seed layer is composed of powered and unpowered line segments placed on the x -axis, as illustrated in Figure 1.3. Suppose that there is only one powered line segment, given as $[p, q]$, and the unpowered line segments are given as $[L_i, R_i]$ for $1 \leq i \leq N$. Construct a level set function $\phi_0(x, y)$ that represents the powered line segment $[p, q]$ and solve (2.3) for $\phi(x, y, t)$, as described in section 2. The interface will grow from the powered line segment in the normal direction. After a time interval t , the interface comes in contact with (or has passed over) one of the unpowered line segments $[L_j, R_j]$ if $\phi(L_j, 0) \leq 0$ or $\phi(R_j, 0) \leq 0$. When this occurs, the j th unpowered segment becomes powered and starts to grow with the rest of the interface, which is simulated as follows:

1. Construct a level set function $\psi(x, y)$ representing the line segment $[L_j, R_j]$.
2. Merge the two interfaces by setting $\phi(x, y) = \min(\phi, \psi)$.

Now continue with the evolution starting from with the updated ϕ until another unpowered segment is contacted by the growing interface, then repeat the two steps

above. For the case of multiple powered segments, steps similar to connecting an unpowered segment to the rest of the interface are performed. Suppose there are M powered segments, given by $[p_i, q_i]_{i=1}^M$ with corresponding level set functions $\phi_i(x, y)$. Then the level set function for all of the powered segments is $\phi_0(x, y) = \min_{1 \leq i \leq M} \{\phi_i(x, y)\}$, which provides the initial condition for the level set PDE (2.3).

3.2. Implementation of the solution for the forward 2D problem. In order to solve this problem computationally, space and time are discretized, with a uniform grid in space. If $[L, R]$ is a line segment, then a distance function representing it is

$$(3.1) \quad d_{[L,R]}(x, y) = \begin{cases} \sqrt{(x-L)^2 + y^2}, & x < L, \\ y, & L \leq x \leq R, \\ \sqrt{(x-R)^2 + y^2}, & x > R. \end{cases}$$

As described in section 2, the solution at discrete times $t_n = ndt$ is $\phi^{n+1} = \phi^n - dt$. The time step is chosen to be $dt = 0.1dx$, which ensures that the unpowered segments get turned on at accurate times. The initial data ϕ^0 is the distance function to the powered line segments. Contact with unpowered line segments is performed by the method of section 3.1.

Summarizing the above, if the powered segments are $[p_i, q_i]_{i=1}^M$ and the unpowered segments are $[L_i, R_i]_{i=1}^N$, the algorithm for forward growth is as follows:

1. Compute $\phi_0(x, y) = \min_{1 \leq i \leq M} (d_{[p_i, q_i]}(x, y))$.
2. Set $n = 0$. Repeat the following until the final growth time T_g is reached (i.e., when $n * dt = T_g$).
 - Set $\phi^{n+1} = \phi^n - dt$.
 - Find all unpowered segments that become powered during this time interval and merge them into the interface. For each unpowered segment $[L_k, R_k]$ do the following:
 - If $\phi^{n+1}(L_k, 0) \leq 0$ or $\phi^{n+1}(R_k, 0) \leq 0$, then compute $d_{[L_k, R_k]}(x, y)$ and set $\phi^{n+1}(x, y) = \min(\phi^{n+1}(x, y), d_{[L_k, R_k]}(x, y))$.
 - Set $n = n + 1$.

4. Forward problem in 3D.

4.1. General idea of the solution for the forward 3D problem. In the 3D case, the domain is $[a_1, a_2] \times [b_1, b_2] \times [0, c_2]$. The seed layers are on the $z = 0$ plane and can now take on any 2D shape, as illustrated in Figure 4.1. For an arbitrary shaped seed layer, initial construction of a level set function is a key step and is discussed in the section 4.3.

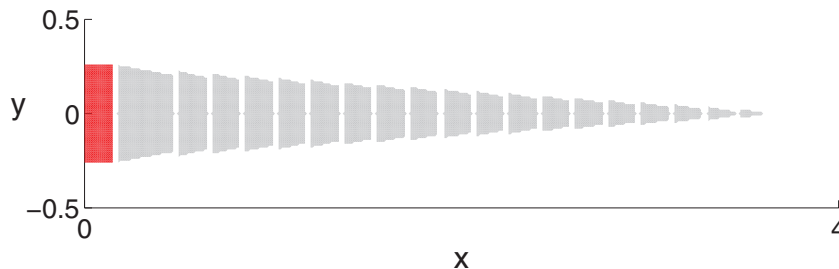


FIG. 4.1. Seed layer used to grow the 3D needle-shaped object in Figure 1.2. The rectangle is powered and the trapezoids are the unpowered layers.

Subsequent evolution of the level set function is essentially the same as in the 2D case: Suppose a level set function, $\phi_0(x, y, z)$, representing the powered layers is given. Using $\phi_0(x, y, z)$ as the initial condition, ϕ is evolved in time as specified by the level set PDE. The interface grows from the powered layer(s) in the normal direction. When the interface comes in contact with unpowered layers, they become powered and start to grow with the interface. In 3D, the boundaries of the layers consist of curves rather than points and some unpowered layers may have multiple boundaries, such as a ring shaped layer. Let γ be the boundary of one of the unpowered layers P . If $\min_{\gamma} \phi \leq 0$, then that unpowered layer has been contacted by the growing interface, it becomes powered, and it starts to grow with the interface. Merger of the layer P with the interface is performed through the following steps:

1. Construct a level set function $\psi(x, y, z)$ representing the layer P .
2. Merge the layer into the interface by setting $\phi = \min(\phi, \psi)$.

Continue with the evolution of ϕ until the growing interface contacts another unpowered layer, then repeat the two steps above.

4.2. Implementation of forward 3D method. In order to solve this problem computationally, discretize the space and time domains using a uniform grid in space. Since the seed layers lie in the plane $z = 0$, the 3D distance function $d_{3D}(x, y, z)$ is related to the 2D distance function $d_{2D}(x, y)$ within the plane $z = 0$ by

$$(4.1) \quad d_{3D}(x, y, z) = \sqrt{[\max(0, d_{2D}(x, y))]^2 + z^2}.$$

For an arbitrarily shaped seed layer U , we construct an approximate distance function $d_{2D}(x, y)$ by approximating P by an N -sided polygon with vertices $\{(x_i, y_i)\}_{i=1}^N$. A method to construct distance functions for polygons is discussed in section 4.3. Detecting contact requires evaluation of the level set function ϕ at the boundaries of the unpowered layer. In the numerical method, this is checked at a discrete set of points along the polygonal boundary. If $\phi \leq 0$ on any of the discrete points on the boundary, the unpowered polygonal layer has been contacted by the interface, becomes powered, and starts to grow with the interface. This is done by performing the following two steps:

1. Construct a distance function $d(x, y, z)$ representing the layer.
2. Take the union of the two interfaces by setting $\phi = \min(\phi, d)$.

As described in section 2 and the 2D case, the iteration to evolve the interface is $\phi^{n+1} = \phi^n - dt$, where ϕ^0 is the distance function representing the powered layer(s). The boundary points of each unpowered layer are checked for contact each time step.

Summarizing the above, let $\{P_i^p\}_{i=1}^M$ be the set of powered polygonal layers and $\{P_i^u\}_{i=1}^N$ the set of unpowered polygonal layers of a given seed layer pattern. The following steps detail forward growth in three dimensions:

1. Set $\phi^0(x, y, z) = \min_{1 \leq i \leq M} (d_i(x, y, z))$ in which $d_i(x, y, z)$ is the distance function (see section 4.3) for the i th powered polygonal layer P_i^p .
2. Set $n = 0$. Repeat the following until the final growth time T_g is reached (i.e., when $n * dt = T_g$).
 - Set $\phi^{n+1} = \phi^n - dt$.
 - Check all of the unpowered polygonal layers that have not already been merged into the growing shape for contact by the interface. If $\phi^{n+1} \leq 0$ at one of the discretized boundary points of an unpowered polygonal layer, it has been contacted by the interface.
 - Suppose the k th unpowered polygonal layer P_k^u has been contacted

by the interface. Compute $d_k(x, y, z)$, and set

$$\phi^{n+1}(x, y, z) = \min(\phi^{n+1}(x, y, z), d_k(x, y, z)).$$

- Repeat the above step for any other newly contacted unpowered layer(s).
- Set $n = n + 1$.

4.3. Constructing signed distance functions for polygons. Suppose that $\{(x_k, y_k)\}_{k=1}^N$, $(x_k, y_k) \in [a_1, a_2] \times [b_1, b_2]$, are the vertices of a polygon traversed in either a clockwise or counterclockwise direction, with $(x_N, y_N) = (x_1, y_1)$. The goal is to create a signed distance function $d(x, y)$ whose zero level set is the polygon boundary. Since the level set computations are performed on a grid, it is only necessary to compute the distance function on the discrete set of grid points. The first step is to create a distance function $d_{k,k+1}(x, y)$ for each line segment $[(x_k, y_k), (x_{k+1}, y_{k+1})]$ (see the appendix for the formula). Note that $d_{k,k+1}(x, y) \geq 0$, since the line segment has no interior. Set $d_0(x, y) = \min_k \{d_{k,k+1}(x, y)\}$ to get a distance function $d_0(x, y)$ for the polygon defined on the grid. However, $d_0(x, y)$ is not a signed distance function since it takes positive values inside the polygon. The remaining step is to identify which grid points are inside the polygon, and negate the value of d_0 at those points.

The Jordan curve theorem states that a point is in the interior of a bounded region if the half line from that point to infinity intersects the boundary of the region an odd number of times. Conversely, the point is in the exterior of the region if the half line intersects the boundary an even number of times. The half line from the point in question to infinity can be taken in any direction. For the grid point (u, v) , a natural choice is the horizontal half line $((-\infty, v), (u, v)]$. Since the polygon should be completely contained in the domain $[a_1, a_2] \times [b_1, b_2]$, only the horizontal line segment $[(a_1, v), (u, v)]$ needs to be checked for intersections with the polygon boundary. Since the boundary of the polygon is composed of line segments, it suffices to check for intersections between the horizontal line segment $[(a_1, v), (u, v)]$ and the polygon boundary segments.

In summary, the steps to form a signed distance function for a polygon on a grid are as follows:

1. Set $d_0(x, y) = \min_{1 \leq k < N} (d_{k,k+1}(x, y))$ in which $d_{k,k+1}(x, y)$ is the distance function from the line segment between vertices k and $k + 1$.
2. For each grid point (u, v) , do the following:
 - Count the number of intersections of the line segment $[(a_1, v), (u, v)]$ with the polygon sides.
 - If the number of intersections is odd, set $d_0(x_i, y_j) = -d_0(x_i, y_j)$.

Once the 2D distance function $d_0(x, y)$ has been computed; the 3D distance function for the polygonal layer is $d_0(x, y, z) = \sqrt{[\max(d_0(x, y), 0)]^2 + z^2}$.

4.4. Results for forward 3D method. Figure 4.1 shows the seed layers that were used to grow the 3D needle shaped object pictured in Figure 1.2. The leftmost (rectangular) layer was powered and the others were unpowered initially. Figure 4.2 shows the results from simulation, using the method described in section 4.2, applied to the seed layer pattern given in Figure 4.1. The simulated object (Figure 4.2) is in excellent agreement with the object that was grown in the laboratory (Figure 1.2).

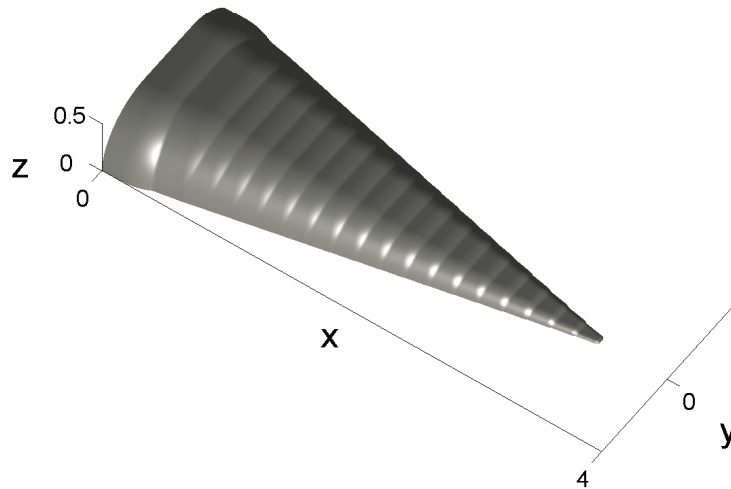


FIG. 4.2. Simulated growth of the needle seed layer pattern of Figure 4.1, using the algorithm described in section 4.2.

5. Inverse problem in two dimensions.

5.1. Problem description and solution. For the inverse problem, a shape is given, and the goal is to determine the seed layer pattern that would approximate that shape. In addition, we find that some shapes are not attainable.

Let the desired final shape Γ be given by $y = F(x)$, contained in the domain $[a_1, a_2] \times [0, b_2]$. The results of section 3 show that the ends of a line segment grow as quarter circles. The first step in solving the inverse problem is to approximate the curve Γ by an envelope of circles that are tangent to Γ and that have centers on the x -axis. Assume that $F \in C^1([a_1, a_2])$. For $s \in [a_1, a_2]$, the circle tangent to F at the point $(s, F(s))$ has center $(c_s, 0)$ and radius R_s given by the formulas

$$(5.1) \quad c_s = F'(s)F(s) + s,$$

$$(5.2) \quad R_s = F(s)\sqrt{[F'(s)]^2 + 1}.$$

A discrete representation of the curve Γ is given by points $\{(x_i, F(x_i))\}_{i=1}^N$ for a given set of points $x_i \in [a_1, a_2]$. Equations (5.1)–(5.2) then determine centers $\{(c_i, 0)\}_{i=1}^N$ and radii $\{R_i\}_{i=1}^N$ for a set of circles $\{C_i\}_{i=1}^N$ that are tangent to F at the points $(x_i, F(x_i))$. Remove any circle C_i whose center is not in the domain (i.e., $c_i \notin [a_1, a_2]$) to get a possibly smaller sets of centers $\{(c_j, 0)\}_{j=1}^M$ and radii $\{R_j\}_{j=1}^M$, with $M \leq N$. See Figure 5.1 for an example of an envelope of circles tangent to the function $F(x) = \frac{1}{2}\sqrt{1-x}$.

For simplicity, we assume that initially there is only a single powered point. It starts growing as a half circle whose radius is equal to the time that point has been powered (since the normal velocity is 1). More generally, the radius of a circle in the envelope is equal to the total time over which the circle should be powered. It follows that the total growth time $t = T_G$ should equal the maximal radius of the circles (i.e., $T_G = \max_j \{R_j\}$). Furthermore, since the j th circle should grow for time R_j , then its start time should be $T_j = T_G - R_j$. The circle C_j gets powered at time T_j if its

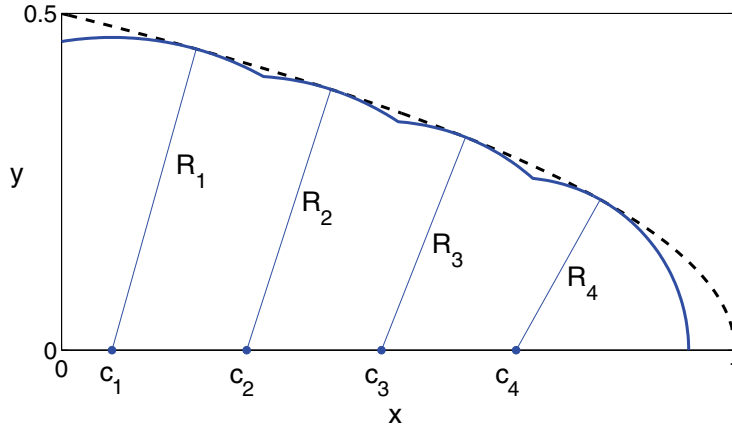


FIG. 5.1. An envelope of 4 circles are used to form an approximation of $y = F(x)$.

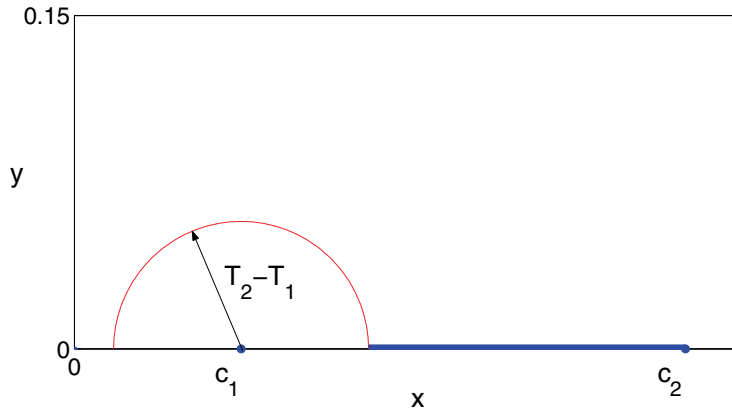


FIG. 5.2. Shows the first circle at time T_2 and the unpowered segment that would power the second circle at time T_2 .

center is the endpoint of a segment that is contacted at time T_j by the circle that is growing from the center $(c_j, 0)$ to its left or right that was powered before T_j .

This is illustrated in Figures 5.1–5.3. In the example shown in Figure 5.1, R_1 is the maximum radius, so $T_G = R_1$, $T_1 = 0$ and the point $(c_1, 0)$ should be powered initially. The point $(c_2, 0)$ should be powered at time T_2 . Figure 5.2 shows the first circle at time T_2 turning on the point $(c_2, 0)$ by contacting the left edge of the segment $[\ell, c_2]$ at time T_2 with $\ell = c_1 + T_2 - T_1$. Figure 5.3 shows the interface at time T_3 , including the growth of the unpowered segment calculated to turn on the second circle.

To summarize the procedure in general for finding the unpowered segment for the k th circle centered at $(c_k, 0)$, with start time T_k , it is necessary to find the closest circle to the k th circle with an earlier start time. Denote the closest circle as the p th circle with center $(c_p, 0)$ and start time T_p . If $c_p < c_k$, then the unpowered segment is on the left side of c_k and is given by $[\ell, c_k]$, where $\ell = c_p + T_k - T_p$. If $c_p > c_k$, then the unpowered segment is on the right side of c_k and is given by $[c_k, r]$, where $r = c_p - (T_k - T_p)$.

Three sufficient conditions on the curve $\Gamma = \{y = F(x)\}$ must be satisfied to

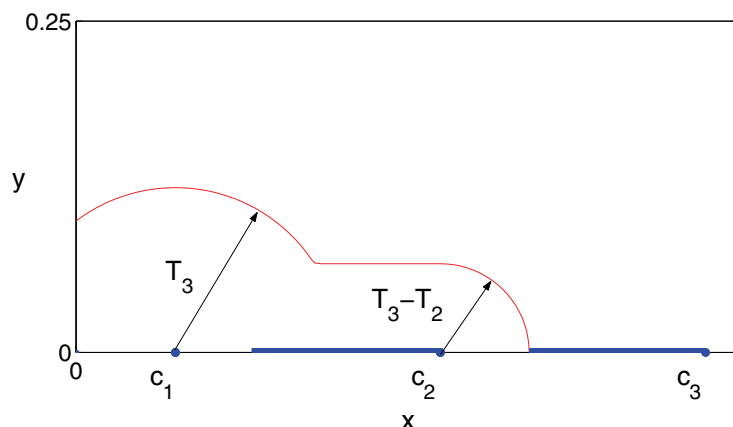


FIG. 5.3. Shows the interface at time T_3 , and the unpowered segment that powers the third circle at time T_3 .

enable a solution to the inverse problem:

1. $F \in C^1([a_1, a_2])$.
2. Γ is the envelope of circles with centers on the x -axis. Each circle intersects Γ at tangent points from below (i.e., the circle lies below Γ).
3. F can have only one local maxima, or if it has multiple local maxima, they must all have the same value.

Although there may exist a seed layer pattern for a given shape that is only in $C([a_1, a_2])$ (but still satisfies the other two conditions), the given algorithm requires that the derivative be continuous as well. The second condition is a necessary one. If that condition is violated, then the shape given by $F(x)$ will not be attainable. Figure 5.4 shows the function $F(x) = \frac{1}{4} \sin(\pi x^2)$, and a circle tangent to it at $(0.8, F(0.8))$ that also crosses F at another point, making that shape unattainable. The present model allows for powered points to be powered by an external source only at the same time. For example, there cannot be one point x_1 powered at time $t = 0$ and another point x_2 independently powered (meaning not powered by contact of the growing surface) at some later time. The third condition is therefore necessary since it ensures that an approximation of the given shape can be attained without needing powered points that start at different times.

5.2. Justification for the construction of the unpowered segments. In constructing the unpowered segments, it was assumed that the point $\ell = c_p + T_k - T_p$ was in between the two circle centers c_p and c_k (this assumes that the contact point is to the left of c_k , but the proof is similar if it were to the right). Here it will be shown that this is always the case if the attainability conditions of section 5.1 are satisfied. Showing that $\ell \in [c_p, c_k]$ is the same as showing that $\ell = c_p + T_k - T_p \leq c_k$, or

$$T_k - T_p \leq c_k - c_p.$$

Suppose this were not true, so that $T_k - T_p > c_k - c_p$. Using the relationships $T_k = T_g - R_k$ and $T_p = T_g - R_p$ gives $T_k - T_p = R_p - R_k$. Then $R_p - R_k > c_k - c_p$, or

$$R_p > d + R_k,$$

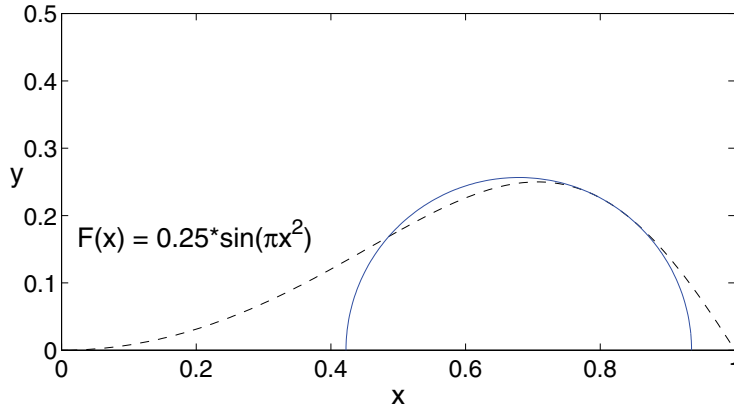


FIG. 5.4. Example of a shape that is not attainable. The circle is tangent to F at $(0.8, F(0.8))$, but it crosses F at another point. This function is not attainable using the given procedure for calculating the seed layer pattern.

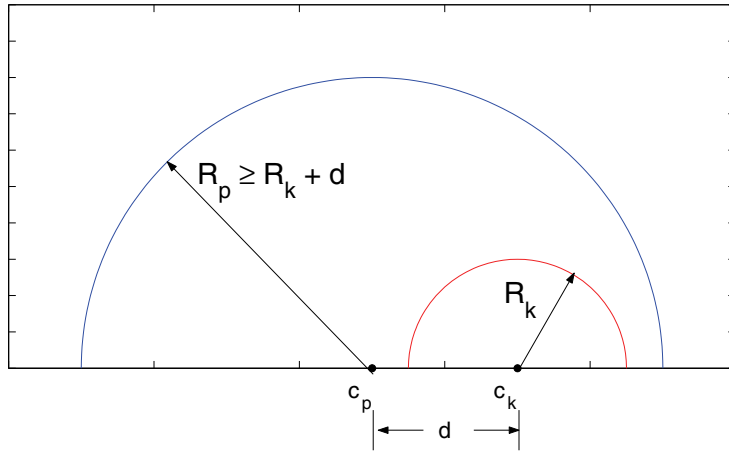


FIG. 5.5. The figure shows that the circle centered at c_p completely encloses the circle centered at c_k if $T_k - T_p \geq c_k - c_p$.

where $d = c_k - c_p$ is the distance between the two circle centers. This means that the circle centered at c_p has a radius so large that it completely encloses the circle centered at c_k (see Figure 5.5). This is impossible, since both circles are tangent to F from below by the second attainability condition in section 5.1. This contradiction shows that $\ell = c_p + T_k - T_p \leq c_k$.

5.3. Results of inverse 2D problem method. Figure 5.6 shows the solution of the inverse problem for the function $F(x) = x(x - 1)^2$. For solution of the inverse problem, a single powered point and eight unpowered segments are allowed. The figure shows the desired function (dashed line), the calculated seed layer pattern, and the resulting growth from the seed layer pattern. The result is in good agreement with the target curve. Better agreement could be achieved by allowing more segments in the seed layer pattern.

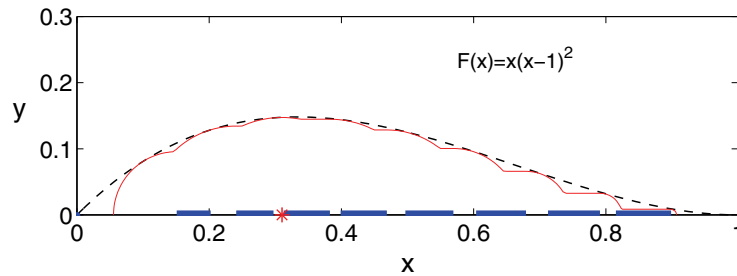


FIG. 5.6. The desired shape is the dotted curve. The seed layer was calculated using the inverse procedure, and the growth of the seed layer is shown by the solid curve. The star is the powered point, and the line segments are the unpowered segments. Nine circles were used for the approximation.

6. Inverse problem in three dimensions.

6.1. A continuous approach. The desired shape Γ is defined by the function $z = F(x, y)$ for (x, y) in the substrate domain $[a_1, a_2] \times [b_1, b_2]$. The goal is to find a seed layer pattern of polygonal layers that will grow to form an approximation of Γ . Instead of a discrete seed layer, consider a continuous seed layer, in which each point in the substrate can be powered at any time. Suppose that this continuous seed layer can grow a shape that exactly matches the given shape Γ at time T_g . This continuous seed layer can then be described by a function $T(x, y)$, which is defined to be the time at which power should be applied to the point (x, y) . Although a continuous seed layer may be impractical, it is a useful theoretical construct.

Let $t_j \in (0, T_g)$ for $j = 1, \dots, n$ be an increasing set of times such that $0 = t_0$, and define the contour curves

$$(6.1) \quad \gamma_i = \{(x, y) \in [a_1, a_2] \times [b_1, b_2] : T(x, y) = t_i\}.$$

Also, define β_i to be the curve that results from growing γ_i in the outward normal direction (i.e., the direction of increasing T) for time $(t_{i+1} - t_i)$.

The initial powered “layer” S_0 is defined to be the curve (or point) γ_0 . For $m > 0$, the seed layer S_m is then defined to be the region in between the two curves γ_m and β_{m-1} . With this definition, we find that S_m is contacted by the growing surface and turns on at time t_m as desired. By its definition, S_0 is powered at time t_0 . The rest follows by iteration: Since the curve γ_{m-1} becomes powered at time t_{m-1} it grows outward at normal speed 1, and at time t_m it hits the curve β_{m-1} which turns on S_m , which is the region in between β_{m-1} and γ_m . This is illustrated in Figure 6.1, which shows two contours γ_m and γ_{m-1} of some function $T(x, y)$. Also shown is β_{m-1} , which is the resulting curve of growing γ_{m-1} in the normal direction for time $t_m - t_{m-1}$.

All of the unpowered layers are constructed in the same way. When the above method is discretized, the powered curve γ_0 will usually not be a curve, but individual points. From a fabrication viewpoint, both individual points and curves as powered layers are not desirable, and it is better to have an actual polygonal powered layer. One simple approach to do this will be discussed later.

6.2. Discretizing and implementing the continuous approach. There are several tasks involved in solving the inverse problem computationally. Each of the following sections describes a task and a method to complete the task. Then, in section 6.2.7, the methods are brought together to give the solution to the inverse problem.

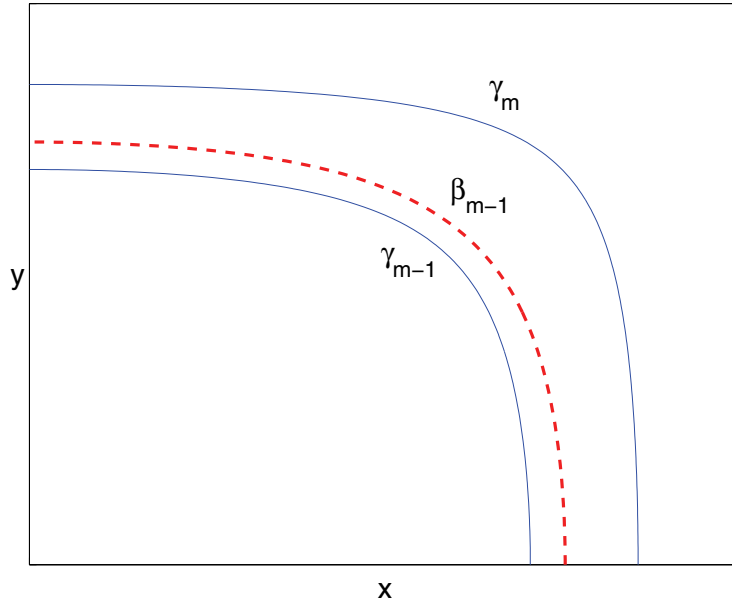


FIG. 6.1. Two contours γ_{m-1} and γ_m of $T(x, y)$ are shown as solid curves. β_{m-1} is shown as the dashed curve. The unpowered layer is the region in between β_{m-1} and γ_m .

6.2.1. Relationship between $T(x, y)$ and $F(x, y)$. A key component of the previous section was knowing $T(x, y)$ for a given shape Γ defined by $z = F(x, y)$. An equivalent condition will be stated and used here. Consider a seed layer that is only a single powered point. Since the growth is uniform in the normal direction with speed 1, the powered point will grow as a hemisphere, with a radius equal to the time the point has been powered. Let the given shape $z = F(x, y)$ satisfy the following conditions:

1. $F \in C^1([a_1, a_2] \times [b_1, b_2])$.
2. Γ can be formed as an envelope of hemispheres (with centers on the $z = 0$ plane) that are tangent to Γ from below (i.e., the sphere lies below the surface $z = F(x, y)$).
3. F has only one local maximum, or if it has multiple local maxima, F has the same value at all local maxima.

Under the above conditions, there should exist a function $R(x, y)$, $(x, y) \in [a_1, a_2] \times [b_1, b_2]$, that gives the radius of the sphere with center (x, y) that is tangent to the given shape F . A continuous seed layer can be defined by $R(x, y)$, since $R(x, y)$ is the length of time that a point (x, y) should be powered. Define $T_g = \max_{(x, y)} \{R(x, y)\}$, which is the length of time that power should be applied to the seed layer. Then $T(x, y) = T_g - R(x, y)$ is the time at which the point (x, y) should become powered. Therefore, finding $T(x, y)$ is equivalent to finding $R(x, y)$. By considering the equation of a sphere with center (x, y) on the $z = 0$ plane, the relationship between $R(x, y)$ and F is given by the following three equations:

$$(6.2) \quad x = F_x(x_s, y_s)F(x_s, y_s) + x_s,$$

$$(6.3) \quad y = F_y(x_s, y_s)F(x_s, y_s) + y_s,$$

$$(6.4) \quad R(x, y) = \sqrt{(x - x_s)^2 + (y - y_s)^2 + [F(x_s, y_s)]^2}.$$

The point $(x_s, y_s, F(x_s, y_s))$ is the point of tangency of the sphere to the surface $z = F(x, y)$.

6.2.2. Computing $T(x, y)$. The first step in numerically solving the inverse problem is to discretize the domain of the substrate $[a_1, a_2] \times [b_1, b_2]$. A uniform grid is used. Let $N_x + 1$ be the number of grid points in x and $N_y + 1$ the number of grid points in y . Then the grid sizes are $dx = (a_2 - a_1)/N_x$ and $dy = (b_2 - b_1)/N_y$, and the grid points are $\{(x_i, y_j) = (a_1 + i * dx, b_1 + j * dy)\}$ for $0 \leq i \leq N_x$ and $0 \leq j \leq N_y$. The radius value $R(x_i, y_j)$ on the grid point (x_i, y_j) is found by solving the nonlinear equations (6.2)–(6.3). Numerical experiments indicate that the following simple functional iteration converges fairly quickly:

$$(6.5) \quad x_s^{n+1} = x_i - F_x(x_s^n, y_s^n)F(x_s^n, y_s^n),$$

$$(6.6) \quad y_s^{n+1} = y_j - F_y(x_s^n, y_s^n)F(x_s^n, y_s^n).$$

A good initial guess is $x_s^0 = x_i$ and $y_s^0 = y_j$: Once (x_s, y_s) is found, (6.4) is used to compute $R(x_i, y_j)$. This process is repeated to compute R for each grid point in the discretized domain. Then the growth time is determined by $T_g = \max_{\{0 \leq i \leq N_x, 0 \leq j \leq N_y\}} \{R(x_i, y_j)\}$ and the function T is given by $T(x_i, y_j) = T_g - R(x_i, y_j)$ on the discretized domain.

6.2.3. Calculating contour curves. Although any conventional methods can be employed to calculate the contour curves of $T(x, y)$, a simple method is presented here. Let $T(x, y)$ be given on the grid points defined in section 6.2.2. Linear interpolation is used to find the points of a contour curve $\gamma = \{(x, y) : T(x, y) = \alpha\}$ on a triangulated grid. Consider the lower triangular cell with corners (x_i, y_j) , (x_{i+1}, y_j) , and (x_{i+1}, y_{j+1}) . If γ passes through that lower triangular grid cell, then γ can be approximated by a line segment $L_{i,j}^\ell$ inside the lower triangular grid cell. Linear interpolation can determine the endpoints of the line segment $L_{i,j}^\ell$, which could be on the horizontal grid cell boundary $[(x_i, y_j), (x_{i+1}, y_j)]$, the vertical grid cell boundary $[(x_{i+1}, y_j), (x_{i+1}, y_{j+1})]$, or the diagonal grid cell boundary $[(x_i, y_j), (x_{i+1}, y_{j+1})]$. Every lower triangular grid cell is checked to see if γ passes through it, and if so a line segment $L_{i,j}^\ell$ is computed. Similarly, the upper triangular grid cells with corners (x_i, y_j) , (x_i, y_{j+1}) , and (x_{i+1}, y_{j+1}) are also checked to see if they contain part of γ , and if so, a line segment $L_{i,j}^u$ is computed. The collection of all line segments $L_{i,j}^\ell$ and $L_{i,j}^u$ is then a representation of the contour curve γ . Connectivity of this curve is guaranteed, since the end of one line segment corresponds to the beginning of another line segment. Figure 6.2 illustrates the triangulated grid and the points in the triangular grid cell boundaries that define the ends of the line segments.

6.2.4. Evolving a curve in the normal direction. One step in section 6.1 for finding an unpowered layer requires growing a contour curve of $T(x, y)$ in the outward normal direction, which is performed using the level set method. Let γ be the contour $T(x, y) = \alpha$ that is to be moved in the outward normal direction for time τ . Using the method discussed in section 6.2.3, compute a set of line segments $\{L_k\}$ that represents the contour curve γ . Following steps similar to those outlined in the section 4.3, a distance function representing the polygonal approximation of γ is computed as $\psi(x, y) = \min_k \{d_k(x, y)\}$, where $d_k(x, y)$ is the distance function for the line segment L_k (a formula is given in the appendix). Note that ψ will not be a signed distance function, since $d_k(x, y) \geq 0$. In general, γ will segment the domain into two parts: the interior region (where $T(x, y) < \alpha$) and the exterior region (where

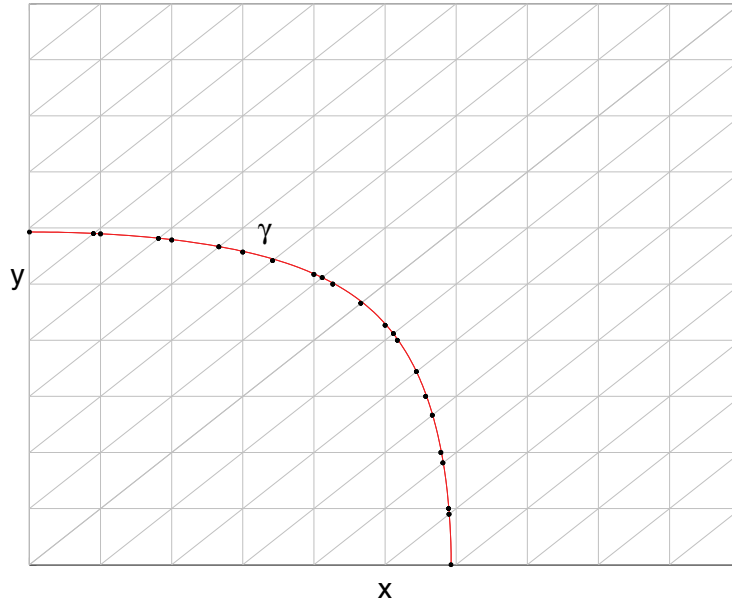


FIG. 6.2. The circles are the endpoints of the line segments used for the discretization of the curve γ .

$T(x, y) > \alpha$). To make a signed distance function, the value of ψ in the interior region must be negated. Therefore, a signed distance function ϕ for the curve γ is defined as

$$\phi(x, y) = \begin{cases} -\psi(x, y) & \text{if } T(x, y) < \alpha, \\ \psi(x, y) & \text{otherwise.} \end{cases}$$

Then the curve from growing γ in the normal direction for time τ is the zero level set of $\phi(x, y, 0) - \tau$. Again, the methods in section 6.2.3 can be used to form a polygonal approximation of the new curve.

6.2.5. Forming an unpowered layer from β and γ . The final step in the construction of a layer S_m is to connect the two curves, β_{m-1} and γ_m , that define it in the $z = 0$ plane. Let $\{L_k^\beta\}$ and $\{L_k^\gamma\}$ be the set of line segments representing the curves β_{m-1} and γ_m , respectively. The unpowered layer is the region between these two curves consisting of points (x, y) with $t_{m-1} < T(x, y) < t_m$. In the simplest case, β_{m-1} and γ_m are each single connected curves. If they are closed curves, then S_m is the region between them. If they are not closed, then each of them has two endpoints that should lie on the boundary of the domain. In this case, one can easily identify the matching endpoints (by the condition $t_{m-1} < T(x, y) < t_m$) and form a curve connecting them along the domain boundary. In general, γ and β may have multiple disjoint pieces that define multiple layers (in such a case, these layers would all have the same start time). A more sophisticated approach to forming layers may have to be used to connect complicated curves, but this has not proved to be necessary in the numerous examples we have simulated.

6.2.6. The powered “layer.” In the presentation above, the initially powered layer consists of a single point or curve that is discretized to a set of points. The powered points should have start time $t = 0$, so they are the points on the grid

$\{(x_k^P, y_k^P)\}_{k=1}^{N_P}$ that satisfy $T(x_k^P, y_k^P) = 0$. In order to calculate the first unpowered layer, the powered points will have to be grown for time t_1 . The signed distance function for a point is (u, v) is $d_{(u,v)}(x, y) = \sqrt{(x-u)^2 + (y-v)^2}$. To grow the powered points for time t_1 , the distance function for the powered points must first be computed as

$$(6.7) \quad d(x, y) = \min_{k=1, \dots, N_P} \{d_{(x_k^P, y_k^P)}(x, y)\}.$$

Then, the zero level set of $d(x, y) - t_1$ is the curve resulting from growing the powered points for time t_1 .

6.2.7. Constructing the seed layer pattern. This section brings together all of the methods in the previous sections to formulate in detail a solution to the inverse problem. Let the substrate domain be $[a_1, a_2] \times [b_1, b_2]$, and assume that the surface $z = F(x, y)$ satisfies conditions (1)–(3) in section 6.2.1.

- Discretize the domain using a uniform grid, where $N_x + 1$ is the number of grid points in x and $N_y + 1$ is the number of grid points in y .
- Compute $R(x, y)$ on the grid using the method described in section 6.2.2. Define $T_g = \max_{\{0 \leq i \leq N_x, 0 \leq j \leq N_y\}} \{R(x_i, y_j)\}$.
- Set $T(x, y) = T_g - R(x, y)$ on the grid.
- Define a positive integer n , and choose a set of times $\{t_i\}_{i=1}^n$ such that $0 = t_0 < t_1 < t_2 < \dots < t_n < T_g$.
- Find the powered points of the seed layer $\{(x_k^P, y_k^P)\}_{k=1}^{N_P}$ that satisfy $T(x_k^P, y_k^P) = 0$.
- Using the technique given in section 6.2.6, compute the distance function $d(x, y)$ for the powered points.
- Compute the parameterization of $\beta_0 = \{(x, y) : d(x, y) = t_1\}$ using the method in section 6.2.3.
- Compute the parameterization of $\gamma_1 = \{(x, y) : T(x, y) = t_1\}$ using the method in section 6.2.3.
- Use the method in section 6.2.5 to form the first unpowered layer that is in between β_0 and γ_1 .
- For $m = 2, \dots, n$, repeat the following:
 - Grow γ_{m-1} in the outward normal direction for time $(t_m - t_{m-1})$ using the method in section 6.2.4 and label the resulting curve β_{m-1} .
 - Compute the parameterization of $\gamma_m = \{(x, y) : T(x, y) = t_m\}$ using the method in section 6.2.3.
 - Use the method in section 6.2.5 to construct the m th unpowered layer that is in between β_{m-1} and γ_m .

6.2.8. Justification for the construction of the unpowered layers. In the construction of the unpowered layers, it was assumed that the curve β_{m-1} is between γ_{m-1} and γ_m (i.e., that when γ_{m-1} moves in the normal direction for time $t_m - t_{m-1}$, it does not cross γ_m). We now prove this, assuming that the shape $z = F(x, y)$ satisfies the attainable conditions in section 6.2.1.

From any point (x^{m-1}, y^{m-1}) on γ_{m-1} , consider the line segment extending in the outer normal direction to a point (x^m, y^m) on the curve γ_m , and define the distance $d = |(x^{m-1}, y^{m-1}), (x^m, y^m)|$. The map from (x^{m-1}, y^{m-1}) to (x^m, y^m) is one-to-one if the curves are smooth and the distance between them is sufficiently small. Since the curve β_{m-1} is defined by moving γ_{m-1} a distance $t_m - t_{m-1}$, it suffices to show that $t_m - t_{m-1} \leq d$.

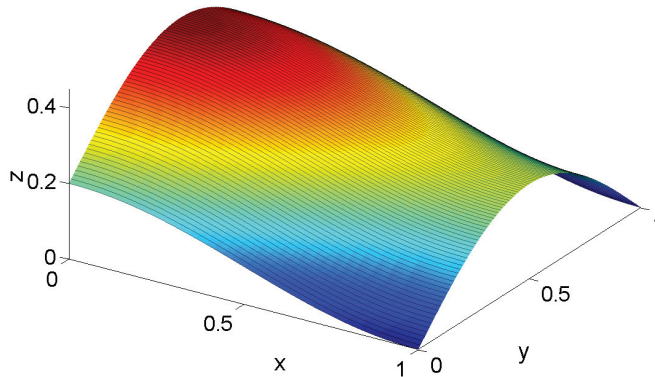


FIG. 6.3. The desired shape is given by $F(x, y) = 0.1 \cos(\pi x) + 0.25 \sin(\pi y) + 0.1$. The color map is a function of height and is used only to aid visualization of the surface.

Suppose that the opposite is true, so that $t_m - t_{m-1} > d$. By condition 2 in section 6.2.1, the sphere S_{m-1} centered at (x^{m-1}, y^{m-1}) with radius $R^{m-1} = T_g - t_{m-1}$ and the sphere S_m at (x^m, y^m) with radius $R^m = T_g - t_m$ are both tangent to the surface $z = F(x, y)$. It follows that $R^{m-1} \geq R^m + d$, which says that S_{m-1} strictly contains S_m , which is not possible if both are tangent from below by condition 2 in section 6.2.1. This contradiction shows that $t_m - t_{m-1} \leq d$, so that the curve β_{m-1} is between γ_{m-1} and γ_m .

6.2.9. A powered polygonal layer. The algorithm outlined in section 6.2.7 constructs the powered layer as points rather than the preferred polygonal shape. A simple way to construct a powered polygonal layer is to set the first unpowered layer as powered. The final growth time would then have to be reduced by t_1 , the start time of the first unpowered layer. Doing this will increase the error of the final shape, but this error can be reduced by choosing t_1 to be small.

6.2.10. Results. Figures 6.3–6.5 show a prescribed shape $F(x, y) = 0.1 \cos(\pi x) + 0.25 \sin(\pi y) + 0.1$, the seed layer pattern calculated using the steps in section 6.2.7, and the result of the forward growth of the calculated seed layer pattern, respectively. Figures 6.6–6.8 show similar results for the prescribed shape $F(x, y) = -(x - 0.5)^2 - (y - 0.5)^2 + 0.25$. Both examples were computed on the domain $[0, 1] \times [0, 1]$ with $N_x = N_y = 100$ and $n = 12$. The computational time to construct the seed layer patterns for the above examples was about 5 seconds.

6.3. Error analysis. In this section, a computational analysis of the accuracy of the inverse algorithm of section 6.2 is presented using the two test problems in section 6.2.10 for various values of the number of grid points (N_x and N_y) and the parameter n , which determines the number of unpowered layers ($\geq n$). In order to resolve the seed layers, the number of grid points should be larger than the number of layers.

The accuracy of the inverse solution is measured by the error in the subsequent forward growth. The forward problem is performed on the computational domain $[0, 1] \times [0, 1] \times [0, 0.5]$, with a fixed number of grid points $N_x^F = 100$, $N_y^F = 100$, and $N_z^F = 50$, with grid sizes $dx^F = \frac{1}{N_x^F + 1}$, $dy^F = \frac{1}{N_y^F + 1}$, and $dz^F = \frac{0.5}{N_z^F + 1}$. Most of the error in the forward growth itself is removed, since the start times of the layers

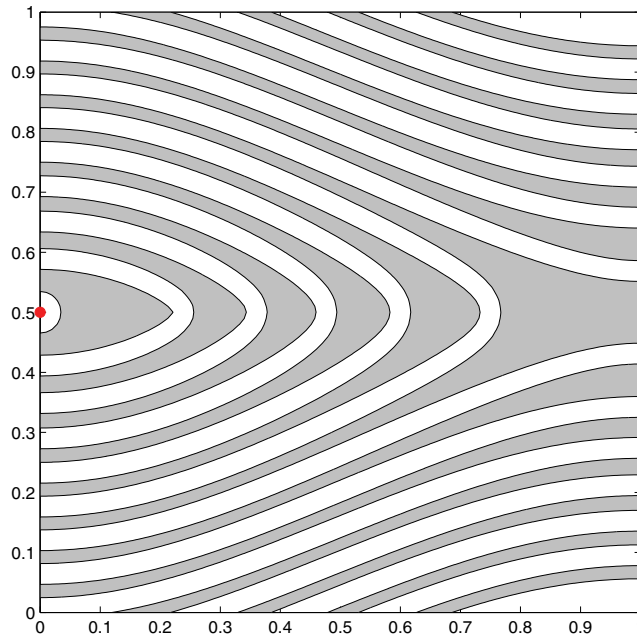


FIG. 6.4. The calculated polygonal seed layer pattern for the shape given by $F(x, y) = 0.1 \cos(\pi x) + 0.25 \sin(\pi y) + 0.1$, shown in Figure 6.3. The dot is the powered point.

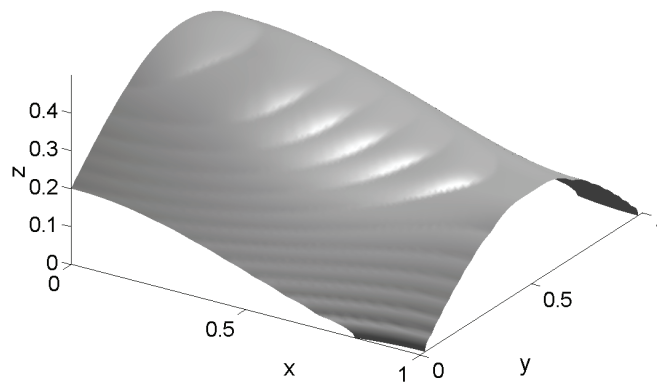


FIG. 6.5. The forward growth of the calculated polygonal seed layer pattern in Figure 6.4. It compares well to the desired shape in Figure 6.3.

are all known. The resulting level set function $\phi(x, y, z)$ is converted to a function $z = F^\phi(x, y)$ defined on the fixed forward grid using linear interpolation. A numerical L^2 error between F^ϕ and F is computed as

$$(6.8) \quad \|F - F^\phi\| = \sqrt{\sum_{i=0, j=0}^{N_x^F, N_y^F} |F(x_i, y_j) - F^\phi(x_i, y_j)|^2 dx^F dy^F}.$$

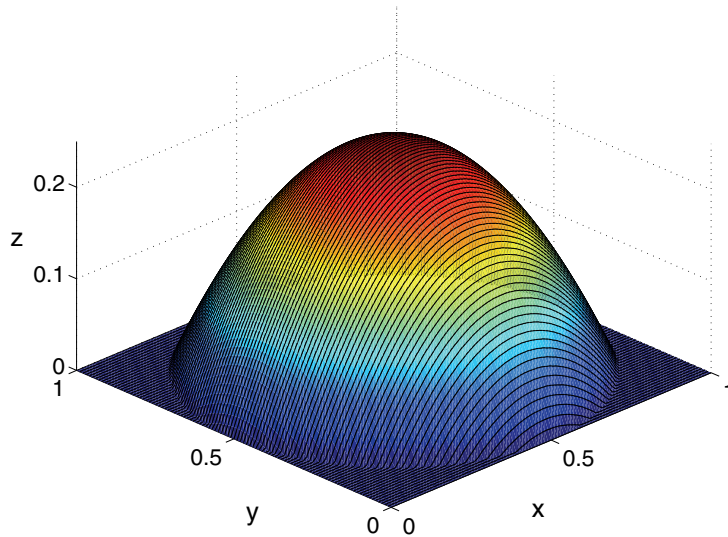


FIG. 6.6. The desired shape is given by $F(x, y) = -(x - 0.5)^2 - (y - 0.5)^2 + 0.25$. The color map is a function of height and is used only to aid visualization of the surface.

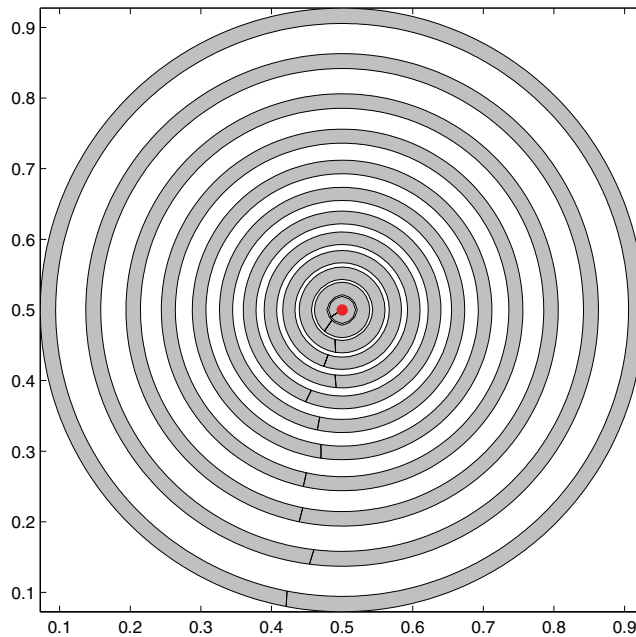


FIG. 6.7. The calculated polygonal seed layer pattern for the shape given by $F(x, y) = -(x - 0.5)^2 - (y - 0.5)^2 + 0.25$, shown in Figure 6.6. The dot in the center is the powered point.

Three values of the number of grid points ($N_x = N_y = 50, 100, 200$) and three values for the number of start times ($n = 10, 20, 40$) are used for the inverse algorithm. More grid points make the boundaries of the layers smoother, and more start

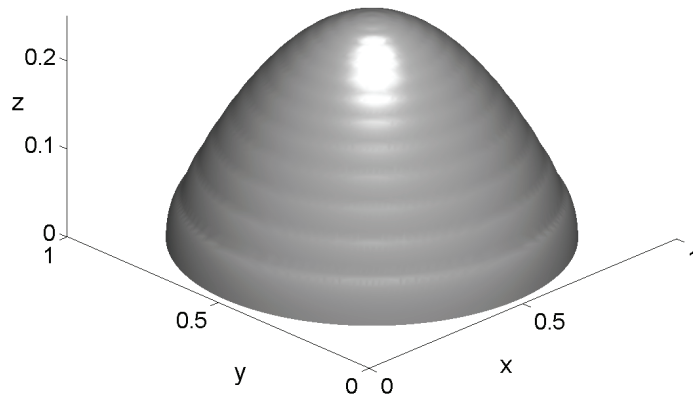


FIG. 6.8. The forward growth of the calculated polygonal seed layer pattern in Figure 6.7. It compares well to the desired shape in Figure 6.6.

TABLE 6.1

The L^2 errors for different values of $N_x = N_y$ and n for the function $F(x, y) = 0.1 \cos(\pi x) + 0.25 \sin(\pi y) + 0.1$.

n	$\ F - F^\phi\ , N_x = 50$	$\ F - F^\phi\ , N_x = 100$	$\ F - F^\phi\ , N_x = 200$
10	0.00425832	0.00423406	0.00422753
20	0.00201249	0.00198800	0.00198200
40	0.00076916	0.00074762	0.00074282

TABLE 6.2

The L^2 errors for different values of $N_x = N_y$ and n for the function $F(x, y) = -(x - 0.5)^2 - (y - 0.5)^2 + 0.25$.

n	$\ F - F^\phi\ , N_x = 50$	$\ F - F^\phi\ , N_x = 100$	$\ F - F^\phi\ , N_x = 200$
10	0.00234398	0.00231595	0.00230947
20	0.00092564	0.00089596	0.00088921
40	0.00036701	0.00034218	0.00034155

TABLE 6.3

Runtimes for different values of $N_x = N_y$ and n for the function $F(x, y) = 0.1 \cos(\pi x) + 0.25 \sin(\pi y) + 0.1$.

n	Runtime, $N_x = 50$	Runtime, $N_x = 100$	Runtime, $N_x = 200$
10	0.421	2.714	20.159
20	0.762	5.188	38.746
40	1.462	10.195	76.701

times produce more unpowered layers in the seed layer. Tables 6.1 and 6.2 show the computed errors for the functions $F(x, y) = 0.1 \cos(\pi x) + 0.25 \sin(\pi y) + 0.1$ and $F(x, y) = -(x - 0.5)^2 - (y - 0.5)^2 + 0.25$, respectively. As the data indicate, the error has very little dependence on $N_x = N_y$ but has nearly linear dependence on n^{-1} . Table 6.3 shows the run times for the function $F(x, y) = 0.1 \cos(\pi x) + 0.25 \sin(\pi y) + 0.1$ with different numbers of grid points and different values of n . All computations were performed on a PC with an Intel M1.6GHz processor with 1.25GB of RAM.

Examining Tables 6.1 and 6.3 shows there is no benefit to using more grid points than needed for resolution of the layers. Using a larger n , however, does improve the accuracy of the seed layer, with only a linear increase in computational time.

7. Conclusions and future work. The method presented here may face material limitations. An accurate fit to a desired shape can result in seed layer elements that are thin strips. The limit on the minimum geometry of the seed layer elements is governed by the photolithographic technology used to pattern them. Typically this ranges from about 1 μm , with readily available optical lithographic systems, to substantially less than 100 nm with electron-beam lithography and industry-leading optical photolithography systems.

Surface roughness could also limit the seed layer geometry. The process used to define the seed layer typically consists of a combination of photolithography, physical vapor deposition (PVD) (e.g., evaporation or sputtering), and chemical etching. The metal films deposited by PVD can be very thin (< 100 nm) and are very smooth. The roughness of electrodeposited films can be highly variable and are subject to the specific process and recipe used. To obtain smoother electrodeposition, one can add chemicals (i.e., brighteners) and perform periodic current reversal.

The solution of the forward problem using the level set method depends on construction of a global distance function from the powered and (initially) unpowered segments. There are several ways to construct this distance function. Our method uses a time discretization that directly mimics the physical evolution of the front. It has some advantages in that additional physics could be easily included. An alternative method in both two and three dimensions would directly construct the distance function from the geometry of the segments.

One additional feature that can be introduced into seed layer patterns is solid boundaries. These solid boundaries are made of an insulated material and prevent growth of the metal shapes beyond the boundaries. This allows even more shape possibilities. Future work would include these solid boundaries in the model for forward growth and would attempt to incorporate them into the inverse procedure. Also, the current model is quite simple, and future work would add more physics into the model, such as nonuniform growth or perhaps some diffusion of the growing surface.

Appendix. The distance function for a line segment. Let the line segment be given as $[(x_1, y_1), (x_2, y_2)]$. Without loss of generality, we shall assume that $x_1 < x_2$. There are 4 cases, depending on the slope of the line segment, which we denote as

$$m = \frac{y_2 - y_1}{x_2 - x_1}.$$

1. $m = \pm\infty$.

$$d(x, y) = \begin{cases} \sqrt{(x - x_1)^2 + (y - y_b)^2} & \text{if } y < y_b, \\ |x - x_1| & \text{if } y_b \leq y \leq y_t, \\ \sqrt{(x - x_1)^2 + (y - y_t)^2} & \text{if } y > y_t, \end{cases}$$

where $y_b = \min(y_1, y_2)$ and $y_t = \max(y_1, y_2)$.

2. $m = 0$.

$$d(x, y) = \begin{cases} \sqrt{(x - x_1)^2 + (y - y_1)^2} & \text{if } x < x_1, \\ |y - y_1| & \text{if } x_1 \leq x \leq x_2, \\ \sqrt{(x - x_2)^2 + (y - y_1)^2} & \text{if } x > x_2. \end{cases}$$

3. $0 < m < \infty$.

$$d(x, y) = \begin{cases} \sqrt{(x - x_2)^2 + (y - y_2)^2} & \text{if } y \geq (-\frac{1}{m}(x - x_2) + y_2), \\ \sqrt{(x - x_1)^2 + (y - y_1)^2} & \text{if } y \leq (-\frac{1}{m}(x - x_1) + y_1), \\ \sqrt{(x - x_m)^2 + (y - y_m)^2} & \text{otherwise,} \end{cases}$$

where $x_m = (\frac{1}{m+\frac{1}{m}}) * (mx_1 + \frac{x}{m} + y - y_1)$ and $y_m = m * (x_m - x_1) + y_1$.

4. $-\infty < m < 0$.

$$d(x, y) = \begin{cases} \sqrt{(x - x_2)^2 + (y - y_2)^2} & \text{if } y \leq (-\frac{1}{m}(x - x_2) + y_2), \\ \sqrt{(x - x_1)^2 + (y - y_1)^2} & \text{if } y \geq (-\frac{1}{m}(x - x_1) + y_1), \\ \sqrt{(x - x_m)^2 + (y - y_m)^2} & \text{otherwise,} \end{cases}$$

where $x_m = (\frac{1}{m+\frac{1}{m}}) * (mx_1 + \frac{x}{m} + y - y_1)$ and $y_m = m * (x_m - x_1) + y_1$.

REFERENCES

- [1] D. ADALSTEINSSON AND J. A. SETHIAN, *A level set approach to a unified model for etching, deposition, and lithography. 1. Algorithms and two-dimensional simulations*, J. Comput. Phys., 120 (1995), pp. 128–144.
- [2] D. ADALSTEINSSON AND J. A. SETHIAN, *A level set approach to a unified model for etching, deposition, and lithography. 2. 3-dimensional simulations*, J. Comput. Phys., 122 (1995), pp. 348–366.
- [3] D. ADALSTEINSSON AND J. A. SETHIAN, *A level set approach to a unified model for etching, deposition, and lithography. 3. Redeposition, reemission, surface diffusion, and complex simulations*, J. Comput. Phys., 138 (1997), pp. 193–223.
- [4] S. CHEN, M. KANG, B. MERRIMAN, R. E. CAFLISCH, C. RATSCH, R. FEDKIW, M. F. GYURE, AND S. OSHER, *Level set method for thin film epitaxial growth*, J. Comput. Phys., 167 (2001), pp. 475–500.
- [5] D. JOSELL, D. WHEELER, W. H. HUBER, J. E. BONEVICH, AND T. P. MOFFAT, *A simple equation for predicting superconformal electrodeposition in submicrometer trenches*, J. Electrochem. Soc., 148 (2001), pp. C767–C773.
- [6] P. LIMOUSIN, P. KRACK, P. POLLAK, A. BENAZZOUC, C. ARDOUIN, D. HOFFMANN, AND A.-L. BENABID, *Electrical stimulation of the subthalamic nucleus in advanced Parkinson's disease*, New England J. Medicine, 339 (1998), pp. 1105–1111.
- [7] P. S. MOTTA AND J. W. JUDY, *Multielectrode microprobes for deep brain stimulation fabricated with a customizable 3-D electroplating process*, IEEE Trans. Biomed. Engrg., 52 (2005), pp. 923–933.
- [8] S. OSHER AND R. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, New York, 2003.
- [9] D. WHEELER, D. JOSELL, AND T. P. MOFFAT, *Modeling superconformal electrodeposition using the level set method*, J. Electrochem. Soc., 150 (2003), pp. C302–C310.