



A numerical method for computing minimal surfaces in arbitrary dimension

Thomas Cecil *

ICES, University of Texas, University Station, C0200 Austin, TX 78712, USA

Received 28 June 2004; received in revised form 10 December 2004; accepted 10 December 2004
Available online 8 February 2005

Abstract

In this paper we propose a numerical method for computing minimal surfaces with fixed boundaries. The level set method is used to evolve a codimension-1 surface with fixed codimension-2 boundary in \mathbb{R}^n under mean curvature flow. For $n = 3$ the problem has been approached in D.L. Chopp, 1993 and L.-T. Cheng [D.L. Chopp, Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.* 106(1) (1993) 77–91 and L.-T. Cheng, The level set method applied to geometrically based motion, materials science, and image processing, UCLA CAM Report, 00-20] using the level set method, but with a more complicated boundary conditions. The method we present can be generalized straightforward to arbitrary dimension, and the framework in which it is presented is dimension independent. Examples are shown for $n = 2, 3, 4$.

© 2005 Elsevier Inc. All rights reserved.

1. Introduction

Given a fixed codimension-2 boundary Γ in \mathbb{R}^n , we would like to find a codimension-1 surface S of minimal surface area that takes Γ as its boundary. If we let S be the set $\{x|\varphi(x) = 0\}$ for a function $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$, then the surface area to be minimized can be written as

$$A = \int_{\mathbb{R}^n} |\nabla \varphi| \delta(\varphi) dx. \quad (1)$$

Applying the method of gradient descent to (1) we arrive at the evolution PDE

$$\varphi_t = \delta(\varphi) \nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right). \quad (2)$$

* Tel.: +512 232 7761.

E-mail address: tcecil@ices.utexas.edu.

Within the level set framework it is advantageous to avoid the δ function and so the related PDE:

$$\varphi_t = |\nabla\varphi|\nabla \cdot \left(\frac{\nabla\varphi}{|\nabla\varphi|} \right), \quad (3)$$

that also evolves φ towards a minimizer of (1), is studied here. Evolution under (3) is known as mean curvature flow. The basic idea behind our technique is to initialize a surface that passes through Γ and then evolve it to steady state using (3), while forcing it at all times to span Γ .

The applications of minimal surfaces span a wide variety of sciences. The surfaces of chemical structures have been studied [17,4]. The interface between crystals and organic matter in the skeletal element of sea urchins can be described as a minimal surface [21]. Minimal surfaces are found in the ternary mixtures of oil, water and surfactant [29]. They have also been studied in condensed matter physics [19]. Overviews of the areas of physics, chemistry and biology in which minimal surfaces play a role can be found in [1,16].

The study and computation of minimal surfaces has a long history. Classical theory can be found in [8,22]. Some of the first numerical approximations can be found in [9]. There has been much study in the dimension $n = 3$, and there are many finite element approaches [14,12]. In [25] minimal surfaces were approximated by the level sets of functions of least gradient. Mean curvature flow was used in [10] to compute stable minimal surfaces using finite elements on surfaces. A network of marker particles is used in the works [2,7,30], and non-parametric representations were used in [15,13]. See [11] for a more detailed listing.

Here we have chosen to use a level set method because of its flexibility when handling topological changes, especially in higher dimensions. While it requires an extra dimension of storage to track the function away from the level set of interest, this extra storage can usually be reduced by using locally adaptive methods with higher resolution near the interface [26]. Level set methods for computing minimal surfaces were also employed in [6,5].

A method proposed in [6] uses a similar level set framework as our method for $n = 3$, but requires complicated boundary conditions for φ near Γ . These boundary conditions require the user to find the intersection of various lines and planes with Γ . Thus, an analytic representation of Γ must be given or constructed to find these intersections. Also, the boundary may be unable to avoid an inconsistent construction. The generalization of that method to higher dimensions is also not available. The method in [5] a modified energy is minimized which fixes the solution in a band near the codimension-2 boundary of the minimal surface.

Our method is similar to the method in [6,5] away from Γ , using finite difference methods for the mean curvature motion equation [23,24]. However, near Γ we use a different technique. To form the spatial derivatives on the right side of (3) we use a radial basis function (RBF) reconstruction of φ with stencil points that lie exactly on Γ . Then this reconstruction is differentiated to find the needed spatial derivatives. Therefore no analytic construction of Γ is needed, and the user only needs to know data points on the minimal surface boundary Γ . Because the RBF reconstruction is dimension independent, the method easily generalizes to arbitrary dimension and we have obtained results in \mathbb{R}^4 .

The outline of the paper is as follows: we begin with the description of the evolution procedure away from Γ . Secondly, we discuss the RBF reconstruction and the procedure for evolution near Γ . Finally, numerical examples in \mathbb{R}^n , $n = 2, 3, 4$ are shown.

2. Evolution procedure away from Γ

2.1. Grid construction

First, we describe the procedure for constructing the computational domain. Given a fixed, compact, codimension-2 boundary in \mathbb{R}^n we find an n -dimensional cube, $\Omega \supset \Gamma$, such that $\|\Gamma - \partial\Omega\| > \epsilon$, where ϵ is

the size of a few (3 or 4) grid cells. This buffering is to ensure that the stencils used in calculating the terms in (3) do not cross $\partial\Omega$. Given Ω we discretize it using a uniform grid with distance between nodes = dx , and call this discretized set $\bar{\Omega}$.

2.2. Evolution of mean curvature flow

We treat the evolution of (3) using the method of lines. The time derivatives are calculated using TVD Runge–Kutta schemes [24]. The CFL condition used is

$$dt \left(\frac{2}{dx^2} + \frac{2}{dy^2} + \frac{2}{dz^2} \right) \leq 1. \tag{4}$$

The spatial derivatives are calculated using central finite differences. The curvature term can be written as

$$k \equiv \left[\sum_{i=1}^n \varphi_{x_i x_i} \left(\sum_{\substack{j=1 \\ j \neq i}}^n \varphi_{x_j}^2 \right) - \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \varphi_{x_i x_j} \varphi_{x_i} \varphi_{x_j} \right] / |\nabla \varphi|^3. \tag{5}$$

Second order finite differencing applied to (5) at a point x_0 results in a stencil S_0 of size 3^n , which consists of all the points $\{y \mid \|y - x_0\|_\infty \leq dx\}$. Finite differencing is also applied to the term $|\nabla \varphi|$. Also, $|\nabla \varphi|$ is regularized by adding a small, $O(dx^2)$, term to it, and we minimize the magnitude of the allowable radius of curvature on the grid to be $1/|k| \leq dx$.

When we say a point x_0 is “away from Γ ,” we mean:

1. There are no points $y \in \Gamma$ that lie in the cube $\{y \mid \|y - x_0\|_\infty < dx\}$, so that the convex hull of the stencil used in advancing (3) does not cross Γ .
2. That all points of S_0 are part of $\bar{\Omega}$. We say this here because in the next section we will explain how certain points of $\bar{\Omega}$ are removed from the computational domain if they are too close to Γ .

We use Neumann BCs $\partial u / \partial n = 0$ on $\partial\Omega$. For some examples these boundary conditions are necessary as they impose symmetry across the boundary, but for some problems they are arbitrary as the region of interest lies completely within the domain.

2.3. Reinitialization

Another procedure that must be applied at every timestep is reinitialization of φ to a distance function. The reason for these repeated applications is to avoid the bunching of levelsets that can occur near Γ , as noted in [6]. Thus reinitialization keeps the level sets well spaced and avoids problems in evolution that can occur when $|\nabla \varphi|$ is too large or small. This is done by evolving the PDE

$$\varphi_t + s(\varphi)(|\nabla \varphi| - 1) = 0, \tag{6}$$

where s is a smoothed version of the signum function. As this is done every timestep we only compute a few iterations of (6). See [24] for details on this computation on uniform grids.

3. Evolution procedure near Γ

In this section we describe the evolution of (3), (6) on the set of all points that are not “away from Γ .”

3.1. Grid adjustment

Because of the possibility that a point $x_0 \in \overline{\Omega}$ could be very close to Γ we remove the set of points $B \equiv \{x \mid \|x - \Gamma\|_2 < d\}$ from $\overline{\Omega}$, where $d < dx$. We use $d = 0.5 dx$ in practice. This is done so that CFL condition is not over restrictive. In general if the smallest radius of curvature that can be resolved on the grid is r_{\min} , then we can expect that we need a CFL condition of $dt \leq O(r_{\min} dx)$. So if we allow points on Γ that are too close, $O(dx^2)$, to points on the uniform grid, then we could end up in situations where $r_{\min} \approx dx^2$, yielding a CFL condition of $dt \leq O(dx^3)$, which is too restrictive to be practical. Similarly, we do not oversample Γ so that points on Γ lie too close together. Thus our final computational grid consists of $(\overline{\Omega} \setminus B) \cup \Gamma$.

3.2. Evolution of mean curvature flow

Firstly, we note that as Γ is part of the minimal surface we set $\varphi(x \in \Gamma, t) = 0 \quad \forall t$.

To evolve (3) at a point x_0 we form a local RBF reconstruction of φ , which we call Φ , and then differentiate Φ to find the spatial derivatives needed in (5) (which include the partial derivatives needed in $|\nabla\varphi|$). This reconstruction is done using a 3^n point stencil, S_0 , found using the method of rays described in [3], with the rays given by $r_k = x_0 + v_k \tau$, $\tau \geq 0$, where v_k are taken to be all points on the unit cubic lattice \mathbb{Z}^n with $\|v_k\|_\infty = 1$. While the size of this stencil grows exponentially large for higher dimensional problems, there is research being done into fast solvers for the interpolation equations. Also, in much higher dimensions the distance from the center point to the farthest vertices of the unit cube is $O(\sqrt{n})$, so perhaps stencils with differently shaped convex hulls would be more appropriate, such as those more closely approximating a hypersphere.

The accuracy of the RBF interpolation is usually inversely proportional to the condition number of the interpolation matrix defining the interpolation problem

$$\Phi(y_i) = \phi(y_i), \quad i = 1, \dots, M, \tag{7}$$

where

$$\Phi(x) := \sum_{j=1}^M \gamma_j \psi(x - y_j), \tag{8}$$

where ψ is a radial basis function (RBF), e.g., $\psi(r) = e^{-\alpha r^2}$, $\{y_i\}$ are the nodes in the M point interpolation stencil. For example as $\alpha \rightarrow 0$ when $\psi(r) = e^{-\alpha r^2}$, the reconstruction Φ converges to standard polynomial interpolation. Also, RBF interpolations are optimal within their *native* spaces [18], which may impose more smoothness than polynomial reconstructions exhibit. The speed of the algorithm will mainly be determined by the speed of the solves of these interpolation equations. However, if storage is not a concern then the coefficients of the derivative terms can be stored initially so that the matrix inversions only need to be done once at the start. This saves considerable time at the expense of storage, assuming that data access is relatively fast.

We will briefly describe the method again here.

Assume we have defined $N = 3^n$ rays, $r_k = x_0 + v_k \tau$, $\tau > 0$, $\|v_k\|_2 = 1$, emanating from x_0 . We then find the neighbor x_j of x_0 that maximizes $V \equiv f(x_0, x_j, v_k, \overline{\Omega}, N)$.

The choice of the stencil preference function f has some flexibility. The general properties it should have are that it should be a non-increasing function of

$$\alpha = \cos^{-1} \left(\frac{x_j - x_0}{\|x_j - x_0\|_2} \cdot v_k \right),$$

and $\beta = \|x_j - x_0\|$ (this norm can be chosen by the user). Also, f will depend on the local density of points, ρ_0 , near x_0 .

For example, in 2 dimensions if we calculate ρ_0 and we have chosen N , then we can define $f = g$ where

$$g \equiv \begin{cases} \cos \alpha, & \text{if } \|x_j - x_0\|_2 < \sqrt{\frac{N}{\pi\rho_0}}, \\ -\infty, & \text{otherwise.} \end{cases}$$

Here we have derived the radius of the support of f , $R = \sqrt{N/(\pi\rho_0)}$, by equating $\rho = N/(\pi R^2)$. Thus we are assuming that the points in the ball $\|x_j - x_0\|_2 < R$ have approximate density ρ_0 . To calculate ρ_0 we can use the number points in the neighboring coarse grid cells of x_0 divided by the total size of those cells. Other examples of f are g^p for $p > 0$, or $f = -\beta$, or combinations of these functions such as g/β or $g - \beta$. In practice we use $f = g - \beta/C$, where C is a scaling constant depending on the mesh size.

Some examples of choices of f are shown in Figs. 1–3. For these examples $v_k = (0,1)$. The function is shown on the left and its contour plot shown on the right in each figure for $0 \leq \alpha \leq \pi/2$. The scale for the x, y axes has been multiplied by 50. Fig. 4 shows an example in 2d of how a single stencil node would be chosen. The contour lines of f are shown, along with $v_0, x_0 = (1,1)$, and candidate stencil nodes $x_j, j = 1:4$. In this example the node that maximizes f is x_1 .

Once S_0 is found we form our reconstruction Φ following the RBF parameter optimization procedure outlined in [3]. Next, approximations to the partial derivatives in (5) are constructed by taking second order central finite differences of Φ on a uniform grid that has minimal distance between nodes = h . In practice we use $h = dx$ or $h = 0.5dx$. This adds a small amount of numerical diffusion to the derivatives that is not present if we were to differentiate Φ exactly.

After the spatial derivatives are calculated, TVD Runge–Kutta time advancement is used to advance the solution.

3.3. Reinitialization

As is done on the points of $\bar{\Omega}$ away from Γ , we evolve (6) for a few iterations after each timestep that (3) is advanced. We use the same technique as is done in Section 3.2 to construct the stencil S_0 , but then instead of using central differencing to approximate derivatives of Φ we use one sided upwind finite differencing where it is needed in the Godunov solver of (6). This means that when the numerical Hamiltonian that is used to approximate $|\nabla\phi|$ calls for $D_{x_i}^+ \phi, D_{x_i}^- \phi$ we will use $(\Phi(x_0 + e_i h) - \Phi(x_0))/h, (\Phi(x_0) - \Phi(x_0 - e_i h))/h$, respectively.

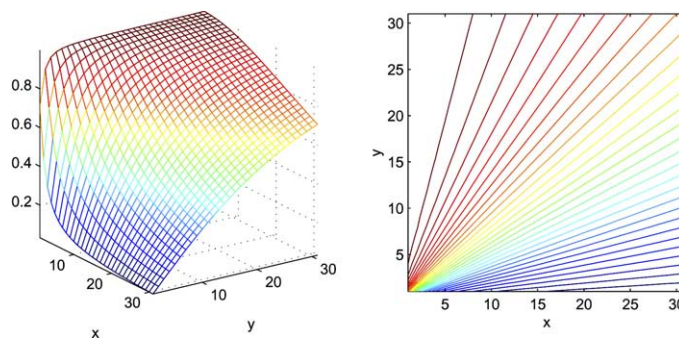


Fig. 1. $f = \cos \alpha$.

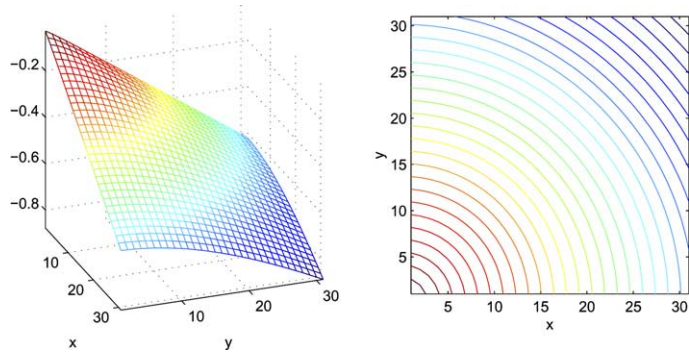


Fig. 2. $f = -\sqrt{x^2 + y^2}$.

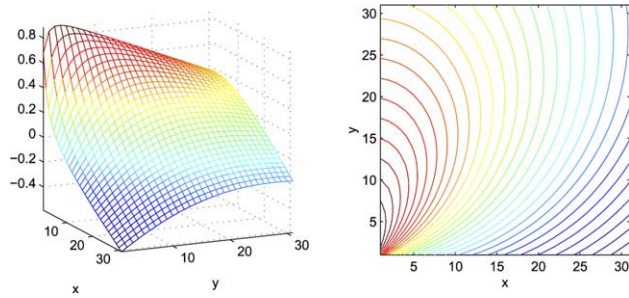


Fig. 3. $f = \cos \alpha - \sqrt{x^2 + y^2}$.

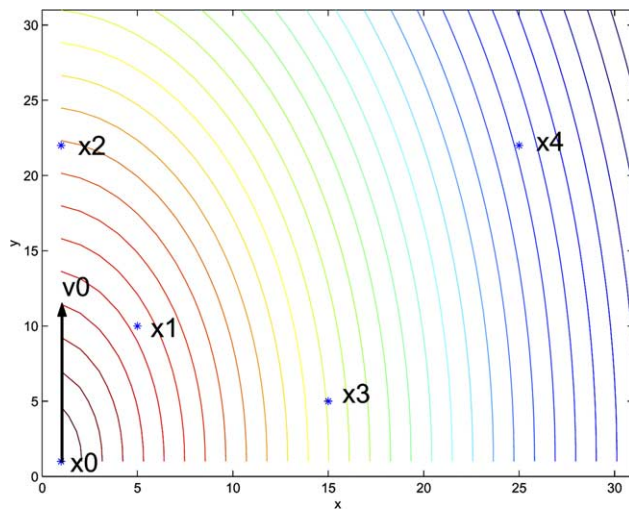


Fig. 4. Example of stencil choice, $f = -\sqrt{x^2 + (y/2)^2}$.

4. Numerical examples

In this section we show numerical results.

Firstly, we discuss an important issue of initialization of ϕ . One way this could be done is described in [5]. This involves taking a large sphere that encloses Γ . The sphere will eventually shrink until its boundary touches Γ where it will stay fixed. Instead, we find an arbitrary surface passing through (or near) Γ and use this as the set $\{\phi = 0\}$ and then reinitialize globally to construct ϕ defined on all of the domain. For the examples we find initial surfaces that extend out of the boundary $\partial\Omega$, so that for all time these surfaces will remain extended through the boundary (because of the Neumann BCs). Thus they will not interfere with the motion of the portion of the surface that is being studied, which usually lies within the convex hull of Γ .

In 2d with Γ being a set of 2 points the minimal surface will be a line, as in Fig. 5. We use a uniform cell width of $dx = 0.04$.

In 3d an example where we can compare our solution with an analytic result is when Γ is given by two circles defined by $x^2 + y^2 = 0.5^2$, $z = \pm 0.277259$. This example was also computed in [6] (see Fig. 6). The initial condition is a cylinder $x^2 + y^2 = 0.5^2$. In this example we use the symmetry of the solution to reduce the computational domain to the space $x, y \in [0, 0.7]$, $z \in [0, 0.35]$, where we use a uniform cell width of $dx = 0.035$. The minimal surface boundary Γ is thus a quarter circle which is discretized using 1000 equispaced points that lie in the computational domain. However, after the stencils are chosen only 191 of these points are used. The number of points on Γ that are used is largely grid dependent. If Γ were shifted so that it was more closely aligned with a specific line or plane of uniform grid points then the number of points used would be greater.

The exact solution is a catenoid with radius $r(z) = 0.4 \cosh(z/0.4)$, whose radius at $z = 0$ is 0.4. In Table 1 we show a convergence estimate based on nested grid refinement. Here the error is measured along the line $x = y$, at $z = 0$.

Note that because of the symmetric nature of the Neumann BCs imposed we are also computing the solution of a catenoid where Γ is given by 2 circles defined by $x^2 + y^2 = 0.5^2$, $z = 0.35 \pm (0.35 - 0.277259)$.

For the 3d and 4d examples, the solutions are interpolated onto a uniform grid at the points that have been removed during computation because of their proximity to Γ .

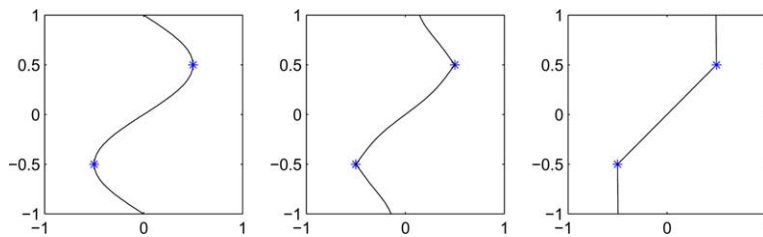


Fig. 5. Minimal surface evolution in 2d with 2 point boundary Γ denoted by the * points. At $t = 0, 0.16, 0.8$.

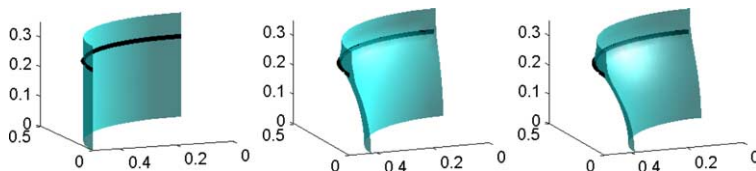


Fig. 6. Minimal surface evolution in 3d with 2 circle boundary Γ denoted by the dark line. At $t = 0, 0.123, 0.367$.

Another 3d example is found by letting Γ lie on Enneper’s minimal surface. For this example we use the parameterization of Enneper’s surface

$$\{x, y, z\} = \left\{ r \cos \theta - \frac{1}{3} r^3 \cos(3\theta), -r \sin \theta - \frac{1}{3} r^3 \sin(3\theta), r^2 \cos(2\theta) \right\}.$$

For Γ we take $r = 1, \theta \in [-\pi, \pi)$ for 2000 equispaced points in θ . The resulting curve resembles the stitching pattern on a baseball. After the stencils are chosen 1618 of these points are used. The computational domain is $[-1.4, 1.4]^3$ and the uniform space step size is $2.8/50$. The initial surface has the same topology as Enneper’s surface, and consists of piecewise planar and cylindrical surfaces that have Gaussian curvature = 0 almost everywhere.

We show two different views of the evolution in Figs. 7, 8 and a comparison with the exact solution in Fig. 9. Only the surface inside Γ should be compared with the exact solution.

In 4d we compute a generalized catenoid solution where Γ is defined as 2 spheres given by $x^2 + y^2 + z^2 = 0.5^2, w = \pm 0.2$. The initial condition is a hypercylinder $x^2 + y^2 + z^2 = 0.5^2$. In this example we use the symmetry of the solution to reduce the computational domain to the space $x, y, z \in [0, 0.6], w \in [0, 0.3]$, where we use a uniform cell width of $dx = 1/30$. The minimal surface boundary Γ is thus an eighth of a sphere that is discretized using 41,692 approximately equispaced points that lie in the computational domain. However, after the stencils are chosen only 8051 of these points are used (see Figs. 10 and 11).

Note that because of the symmetric nature of the Neumann BCs imposed we are also computing the solution of a catenoid where Γ is given by 2 spheres defined by $x^2 + y^2 + z^2 = 0.5^2, w = 0.3 \pm 0.1$.

Table 1
Convergence rate estimate

dx	Error	Rate
0.035	9.504×10^{-3}	
0.0175	2.162×10^{-3}	2.14

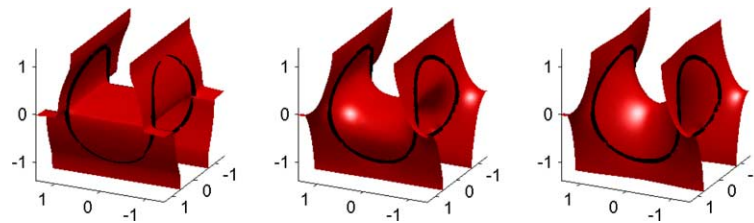


Fig. 7. Minimal surface evolution in 3d with Enneper surface boundary Γ denoted by the dark line. At $t = 0, 0.063, 0.439$.

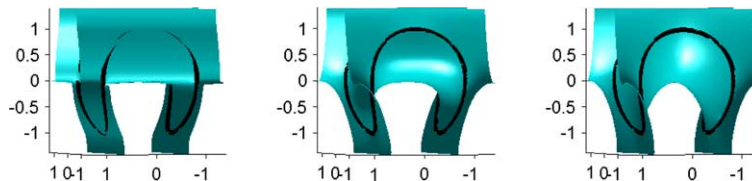


Fig. 8. Minimal surface evolution in 3d with Enneper surface boundary Γ denoted by the dark line. At $t = 0, 0.063, 0.439$.

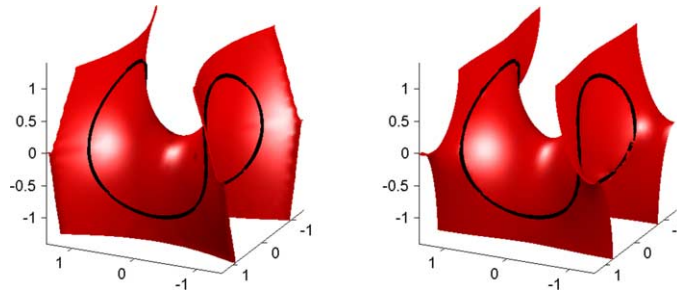


Fig. 9. Minimal surface evolution in 3d with Enneper surface boundary Γ denoted by the dark line. Left: exact solution. Right: computed solution at $t = 0.439$.

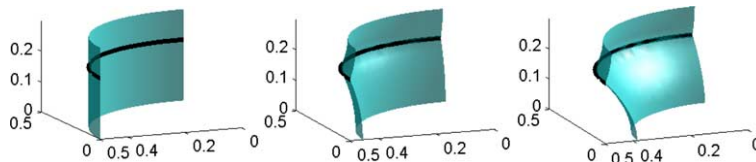


Fig. 10. Minimal surface evolution in 4d with 2 sphere boundary Γ denoted by the dark line. Slices taken at $x = 0$, at $t = 0, 0.034, 0.411$.

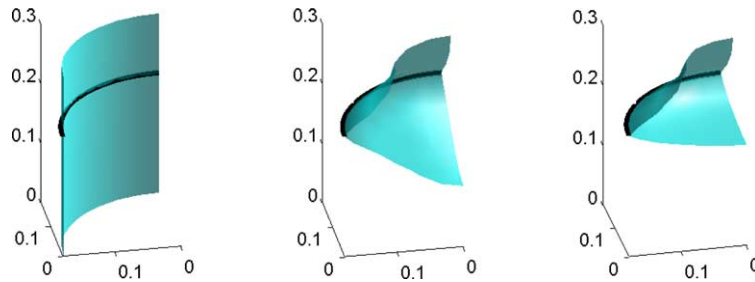


Fig. 11. Minimal surface evolution in 4d with 2 sphere boundary Γ denoted by the dark line. Slices taken at $x = 0.4\bar{6}$, at $t = 0, 0.034, 0.411$.

We show 3d slices of the solution for fixed values of x at various times. Slices taken for fixed y, z values look identical to those shown, and the slices for fixed w values look like spheres.

Note how in the slice taken when $x = 0.4\bar{6}$ that the surface remains a catenoid when $|w| > 0.2$, but has changed topology in the region $|w| < 0.2$.

5. Conclusion

In this paper we introduce a numerical method for computing minimal surfaces in arbitrary dimension that have codimension-2 boundary, Γ , by evolving an initial codimension-1 surface by mean curvature flow. The method uses existing finite differences techniques away from Γ , and a new evolution procedure using radial basis functions near Γ . The framework is described in a way that can be generalized to any dimen-

sion, and computed examples are shown in 2, 3 and 4 dimensions. While uniform grids are used in this paper, it would be advantageous to use an adaptive grid such as those described in [27,20], for higher dimensional problems. The method presented here would still be applicable in these contexts, as it only modifies the underlying evolution scheme near Γ .

Future work can include the study of unstable minimal surfaces. Surfaces with generalized triple points can also be studied, perhaps by the use of multiple level set functions [28]. As computer memory and speed increase higher dimensional problems will also be approached. The method presented can also be applied to other non-linear evolutions with irregular fixed boundaries in arbitrary codimension.

Acknowledgments

This work is supported by NSF Grants DMS-0312222 and ACI-0321917, and an ONR MURI Grant, subcontracted from Stanford University. The author also thanks the referees for their comments.

References

- [1] S. Andersson, S.T. Hyde, K. Larsson, S. Lidin, Minimal surfaces and structures: from inorganic and metal crystals to cell membranes and biopolymers, *Chem. Rev.* (1988) 88.
- [2] K.A. Brakke, The surface evolver, *Exp. Math.* 1 (2) (1992) 141–165.
- [3] T. Cecil, J.L. Qian, S. Osher, Numerical methods for high dimensional Hamilton–Jacobi equations using radial basis functions, *J. Comput. Phys.* 196 (2004) 327–347.
- [4] B. Chen, M. Eddaoudi, S.T. Hyde, M. O’Keeffe, O.M. Yaghi, Interwoven metal-organic framework on a periodic minimal surface with extra-large pores, *Science* (2001) 291.
- [5] L.-T. Cheng. The level set method applied to geometrically based motion, materials science, and image processing. UCLA CAM Report, 00-20.
- [6] D.L. Chopp, Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.* 106 (1) (1993) 77–91.
- [7] C. Coppin, D. Greenspan, A contribution to the particle modeling of soap films, *Appl. Math. Comput.* 26 (4) (1988) 315–331.
- [8] Ulrich Dierkes, Stefan Hildebrandt, Albrecht Küster, Ortwin Wohlrab, *Minimal Surfaces, I,II*, volume 295,296 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1992, Boundary regularity.
- [9] J. Douglas, A method of numerical solution of the problem of Plateau, *Ann. Math.* (2) 29 (1–4) (1927/28) 180–188.
- [10] G. Dziuk, An algorithm for evolutionary surfaces, *Numer. Math.* 58 (6) (1991) 603–611.
- [11] G. Dziuk, J.E. Hutchinson, On the approximation of unstable parametric minimal surfaces, *Calc. Var. Partial Dif.* 4 (1) (1996) 27–58.
- [12] G. Dziuk, J.E. Hutchinson, The discrete Plateau problem: algorithm and numerics, *Math. Comp.* 68 (225) (1999) 1–23.
- [13] D. Greenspan, On approximating extremals of functionals. I. The method and examples for boundary value problems, *ICC Bull.* 4 (1965) 99–120.
- [14] M. Hinze, On the numerical approximation of unstable minimal surfaces with polygonal boundaries, *Numer. Math.* 73 (1) (1996) 95–118.
- [15] R.H.W. Hoppe, Multigrid algorithms for variational inequalities, *SIAM J. Numer. Anal.* 24 (5) (1987) 1046–1065.
- [16] S. Hyde, S. Andersson, K. Larsson, Z. Blum, T. Landh, S. Lidin, B.W. Ninham, *The Language of Shape: The Role of Curvature in Con- Densed Matter: Physics, Chemistry, and Biology*, Elsevier, Amsterdam, 1997, 383 pp.
- [17] S.T. Hyde, S. Andersson, A systematic net description of saddle polyhedra and periodic minimal surfaces, *Z. Krist.* 168 (1–4) (1984) 221–254.
- [18] A. Iske, T. Sonar, On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions, *Numer. Math.* 74 (2) (1996) 177–201.
- [19] R.D. Kamien, T.C. Lubensky, Minimal surfaces, screw dislocations and twist grain boundaries, *Phys. Rev. Lett.* (1999) 82.
- [20] Chohong Min, Local level set method in high dimension and codimension, *J. Comput. Phys.* 200 (1) (2004) 368–382.
- [21] H.U. Nissen, Crystal orientation and plate structure in echinoderm plates, *Science* (1969) 166.
- [22] J.C.C. Nitsche, *Lectures on Minimal Surfaces, Vol. 1. Introduction, Fundamentals, Geometry and Basic Boundary Value Problems* (Trans. Jerry M. Feinberg, with a German foreword), Cambridge University Press, Cambridge, 1989.

- [23] S.J. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12–49.
- [24] S. Osher, R.P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Oxford University Press, 2002.
- [25] Harold R. Parks, Explicit determination of area minimizing hypersurfaces. II, *Mem. Am. Math. Soc.* 60(342) (1986) iv+90.
- [26] Danping Peng, B. Merriman, S. Osher, Hongkai Zhao, Myungjoo Kang, A PDE-based fast local level set method, *J. Comput. Phys.* 155 (2) (1999) 410–438.
- [27] J. Strain, Fast tree-based redistancing for level set computations, *J. Comput. Phys.* 152 (2) (1999) 664–686.
- [28] L. Vese, T. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, *Int. J. Comput. Vision* 50 (3) (2002) 271–293.
- [29] H.G. von Schnering, R. Nesper, The curvature of chemical structures, in: *International Workshop on Geometry and Interfaces*, Aussois, 1990, *J. Phys.* 51(23, Suppl. Colloq. C7) (1990) 383–396.
- [30] H.-J. Wagner, A contribution to the numerical approximation of minimal surfaces, *Computing* 19 (1) (1977/78) 35–58.