Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational Physics 215 (2006) 392–401

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Short note

An improved geometry-aware curvature discretization for level set methods: Application to tumor growth

Paul Macklin *, John Lowengrub

Department of Mathematics, 103 MSTB, University of California, Irvine, CA 92697, USA

Received 25 July 2005; received in revised form 2 November 2005; accepted 11 November 2005
Available online 22 December 2005

Abstract

An advantage of using level set methods for moving boundary problems is that geometric quantities such as curvature can be readily calculated from the level set function. However, in topologically challenging cases (e.g., when two interfaces are in close contact), level set functions develop singularities that yield inaccurate curvatures when using traditional discretizations. In this note, we give an improved discretization of curvature for use near level set singularities. Where level set irregularities are detected, we use a local polynomial approximation of the interface to construct the level set function on a local subgrid, where we can accurately calculate the curvature using the standard 9-point discretization. We demonstrate that this new algorithm is capable of calculating the curvature accurately in a variety of situations where the traditional algorithm fails and provide numerical evidence that the method is second-order accurate. Examples are drawn from modified Hele-Shaw flows and a model of solid tumor growth.

MSC: 65; 65M06; 92C05

Keywords: Moving boundary problems; Level set method; Multiphase Hele-Shaw flows; Tumor growth; Finite differences; Curvature discretization

1. Introduction

Many important physical problems involve the motion of free boundaries or interfaces with velocities dependent upon curvature. For instance, in Hele-Shaw multiphase flows, tumor growth, and crystal growth, the motion of interfaces depends nonlocally upon the derivatives of curvature. Therefore, the stable and accurate computation of curvature is paramount when simulating such systems. This is particularly important in regions where interfaces are in near contact.

Level set methods have been used with good success to implicitly track moving interfaces and automatically detect topology changes in these problems [8–10,12,13]. However, level set functions develop discontinuities in their derivatives near regions of topological change, making the curvature discretization problematic. In [6,7],

[*] Corresponding author.
E-mail addresses: pmacklin@math.uci.edu (P. Macklin), lowengrb@math.uci.edu (J. Lowengrub).
URLs: http://math.uci.edu/~pmacklin (P. Macklin), http://math.uci.edu/~lowengrb (J. Lowengrub).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.11.016

it was demonstrated that if the curvature is computed without regard for the local geometry by using standard centered difference algorithms as proposed in [8–10,12,13], then the curvature becomes oscillatory and inaccurate for interfaces in near contact. This may lead to sudden spikes in the curvature error. Refining the mesh near topology changes may delay the onset of these problems but cannot eliminate them. For example, we show a case in which mesh refinement delays the formation of spikes but actually increases the spike magnitude. In our example, this can lead to the blow-up of the solution when using a traditional curvature discretization. Furthermore, due to computational cost, mesh refinement cannot be continued indefinitely as the distance between interfaces approaches zero.

In [6,7], a complicated curvature discretization was given that addressed the accurate approximation of curvature in complex 2D geometries in the context of a nonlinear model of tumor growth. In this paper, we introduce a simpler and more robust geometry-aware curvature discretization that can be extended to 3 dimensions. We calculate the curvature using standard level set methods when the level set function is sufficiently smooth. Otherwise, our method works by first constructing a properly-oriented (least squares, quadratic) polynomial approximation of the interface through a point. With this curve, we create a local level set function with which to compute the curvature by a standard discretization on a local subgrid.

In our work, we have found that using a local level set function is more robust in calculating the curvature than directly differentiating an interpolating spline [6,7]. Alternative methods of representing the curve (e.g., B-splines; see [1,4,11]) can be used together with our method, but we find that quadratic least squares polynomial approximations are easy to implement and sufficient for second-order accuracy.

Our method calculates the curvature accurately in a variety of difficult topological situations (e.g., merging interfaces, drop fragmentation), as we demonstrate in examples of modified Hele-Shaw multiphase flow and in vivo tumor growth. In the Hele-Shaw example, we present numerical evidence of second-order convergence. Furthermore, we also demonstrate that in this example, the traditional curvature discretization fails in a way that is worsened by decreasing the mesh size. Our method is generally applicable to any level set model involving morphological changes or interfaces in near contact, e.g., multiphase flows (e.g., [13]), dendritic crystal growth (e.g., [3]), and image processing (e.g., [8,12]).

2. Overview

Traditional level set methods (e.g., [8–10,12,13]) compute curvature as

$$\kappa = \nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_x\varphi_y\varphi_{xy} + \varphi_{yy}\varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{3/2}}, \tag{1}$$

where φ is an approximation of the signed distance function to the interface Γ. On a Cartesian grid, this divergence is generally calculated at node points by using a 9-point stencil with centered differences for all the partial derivatives.

Suppose we have a level set function φ defined on a Cartesian grid with mesh points given by $X \times Y = \{x_i\}_{i=1}^{M} \times \{y_j\}_{j=1}^{N}$ and we require the curvature $\kappa(x,y)$ at an interior point $(x, y) \in [x_i, x_{i+1}) \times [y_j, y_{j+1})$. If the level set function is sufficiently smooth to compute the curvature $\kappa(x_k, y_\ell)$ at each mesh point $(x_k, y_\ell) \in \{x_k\}_{k=i-1}^{i+2} \times \{y_\ell\}_{\ell=j-1}^{j+2}$, then we can accurately compute $\kappa(x, y)$ by calculating the curvature at these 16 mesh points (with the 9-point stencil) and using bicubic interpolation.[1] If the level set function is only sufficiently smooth to compute the curvature $\kappa(x_k, y_\ell)$ at each of the four mesh points $(x_k, y_\ell) \in \{x_i, x_{i+1}\} \times \{y_j, y_{j+1}\}$, then we proceed with bilinear interpolation instead. It often occurs that the level set function is insufficiently smooth to allow even a bilinear interpolation. (See Section 3 for a measure of smoothness.) When two interfaces are in close contact (generally 5–7 nodes apart or less), the derivatives of φ become inaccurate and develop discontinuities in the region between the interfaces. Our method provides a means to deal with this situation accurately.

We first detect regions where the traditional curvature discretization fails. In these regions, we find a least squares quadratic, properly-oriented curve $\gamma(s)$ approximating the interface Γ near the point (x, y) where we

[1] In our testing, this gives a second-order accurate curvature [6,7].

desire the curvature. We have found that directly differentiating γ to obtain the curvature is not robust, as γ is sensitive to errors in the parameterization. Instead, we construct a local level set function $\widehat{\varphi}$ about (x, y) and use the standard 9-point stencil on a locally refined subgrid to discretize the curvature.

3. Detecting regions where the traditional curvature fails

In [6,7], we found that if we defined a level set quality function by

$$Q(x,y) = |1 - |\nabla\varphi|| \tag{2}$$

and set a threshold η, then (x, y) is near a singularity of the level set function φ whenever $Q(x, y) \geqslant \eta$; we compute $\nabla\varphi$ using centered finite differences. In our testing, we found that using $\eta = 0.004$ reliably identified such regions without yielding false positives.

Suppose we wish to calculate the curvature at (x_i, y_j) using the standard 9-point discretization. If $Q(x, y) \geqslant \eta$ at any $(x,y) \in \{x_k\}_{k=i-1}^{i+1} \times \{x_\ell\}_{\ell=j-1}^{j+1}$, then the level set function is not smooth enough to accurately discretize the curvature at (x_i, y_j), and we use our geometry-aware algorithm instead.

4. Approximating the interface with proper orientation

Let (x, y) be contained in the mesh square $[x_i, x_{i+1}) \times [y_j, y_{j+1})$, with $\varphi(x, y) = 0$. We seek to construct an accurate approximation $\gamma(s) = (x(s), y(s))$, where s is arclength, of the interface Γ near (x, y) with proper orientation. Let $\mathbf{x_3} = (x_3, y_3) = (x, y)$, and let $s_3 = 0$ such that $\gamma(s_3 = 0) = \mathbf{x_3}$.

We choose points $\mathbf{x_2} = (x_2, y_2)$ and $\mathbf{x_4} = (x_4, y_4)$ where the interface Γ intersects the mesh immediately surrounding $\mathbf{x_3}$. To improve the stability of the approximating curve we seek to construct, we choose these points to be at least $\frac{1}{10}\Delta x$ away from $\mathbf{x_3}$. Similarly, we choose a point $\mathbf{x_1} = (x_1, y_1)$ where the mesh surrounding $\mathbf{x_2}$ intersects Γ and is at least $\frac{1}{10}\Delta x$ from $\mathbf{x_2}$, and $\mathbf{x}\ _\mathbf{5} = (x_5, y_5)$ is similarly chosen to be close to $\mathbf{x_4}$. See Fig. 1.

We choose the ordering $(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5})$ such that when traversing the curve in the direction of increasing arclength s, the region where $\varphi < 0$ is on the left side of the curve. The orientation of the curve can readily be determined by examining the cross product of $\mathbf{x_4} - \mathbf{x_3}$ and $\mathbf{y} - \mathbf{x_3}$, where $\mathbf{y}$ is a point off the curve. See Fig. 2.

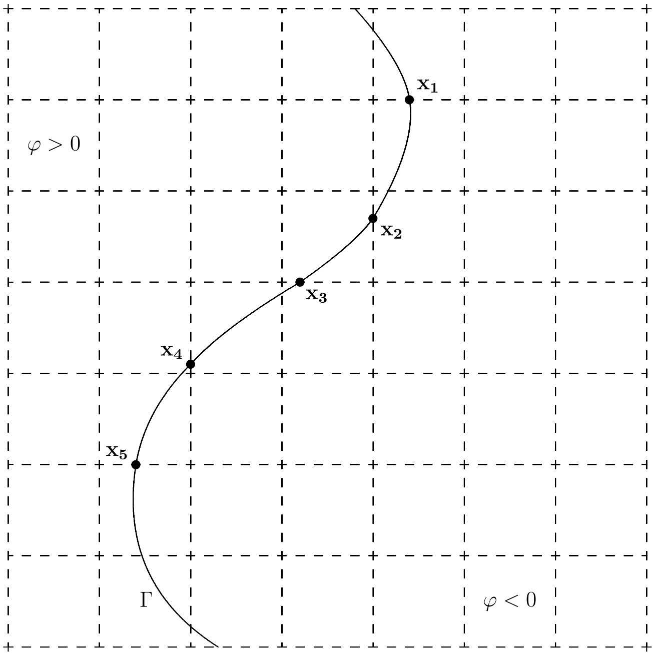

Fig. 1. Finding points on Γ near $\mathbf{x_3}$.

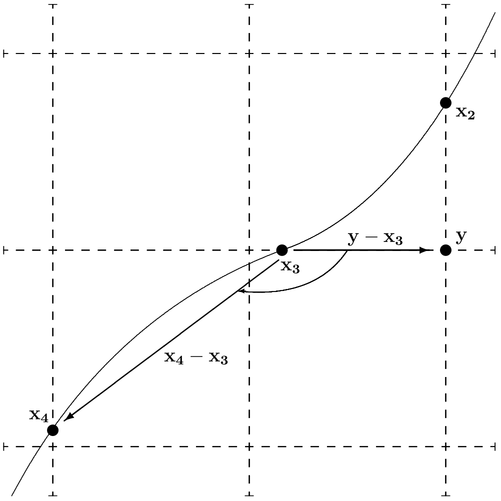

Fig. 2. Determining the orientation of $(\mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4})$: Notice that the z-component of $(\mathbf{y}-\mathbf{x_3})\times(\mathbf{x_4}-\mathbf{x_3})$ is negative, so $\mathbf{y}$ is on the left side of the curve in this orientation.

We choose arclengths s_1, s_2, s_4, and s_5 such that $\gamma(s_i)=\mathbf{x_i}$, for $1 \leqslant i \leqslant 5$. We then approximate these arclengths by using the linear distances between the points. That is, moving backward along the curve from $\mathbf{x_3}=\gamma(s_3)$, $s_3=0$, $s_2=-|\mathbf{x_3}-\mathbf{x_2}|$, and $s_1=s_2-|\mathbf{x_2}-\mathbf{x_1}|$; moving forward, $s_4=|\mathbf{x_4}-\mathbf{x_3}|$ and $s_5=s_4+|\mathbf{x_5}-\mathbf{x_4}|$.

Finally, let $x(s)$ and $y(s)$ be the least squares quadratic curves fitted to $\{(s_i,x_i)\}_{i=1}^{5}$ and $\{(s_i,y_i)\}_{i=1}^{5}$, respectively. We reset the constant coefficients such that $\gamma(0)=\mathbf{x_3}=(x,y)$. Notice that because $\gamma(s)$ is only used to construct the level set function on a local subgrid, it does not affect the position of the actual contour of the original level set function. Although we do not show it here, the resulting curve approximates Γ very well; this is reflected in our numerical tests in later sections.

5. Constructing a new local level set and computing the curvature

Lastly, we construct a local level set function near $\mathbf{x_3}=(x,y)$. For a fixed $\delta>0$, let $\widehat{X}=\{x-\delta,x,x+\delta\}$ and $\widehat{Y}=\{y-\delta,y,y+\delta\}$, so that $\widehat{X}\times\widehat{Y}$ is a 3×3 grid centered at $\mathbf{x_3}$. See Fig. 3. Let $\widehat{\varphi}$ be the local level set function on $\widehat{X}\times\widehat{Y}$. We can choose any desired mesh size $\delta\leqslant\Delta x$ and $\delta\leqslant\Delta y$.

For each point $\widehat{\mathbf{x}}=(\widehat{x}_i,\widehat{y}_j)\in\widehat{X}\times\widehat{Y}$, we set $\widehat{\varphi}_{i,j}$ equal to the signed distance between $\widehat{\mathbf{x}}$ and γ. (The sign is determined based upon whether $\widehat{\mathbf{x}}$ is on the left or right side of the curve.) The end result is a local construction of φ on a refined subgrid that avoids level set singularities. We compute the curvature $\kappa(x,y)$ using the standard centered differences for $\widehat{\varphi}_x$, $\widehat{\varphi}_{xx}$, $\widehat{\varphi}_{xy}$, $\widehat{\varphi}_{yy}$, and $\widehat{\varphi}_y$ on the subgrid $\widehat{X}\times\widehat{Y}$.

We tested with $\delta=\Delta x$, $\frac{1}{10}\Delta x$, $\frac{1}{100}\Delta x$, and $\frac{1}{1000}\Delta x$. In general, we found that all these values worked equally well when using a quadratic $\gamma(s)$. This is because the essential feature of our technique is that it removes the nearby second interface and locally rebuilds the level set function accordingly. However, smaller values tended to give more accurate results when two interfaces were in extremely close contact (less than one mesh point apart). In our remaining work, we chose $\delta=\frac{1}{1000}\Delta x$.

6. Numerical examples

6.1. Two drops merging under modified Hele-Shaw flow

Let Γ be an interface describing the boundary of two circular drops of radius 1 centered at

$$(2.5\cos\theta, 2.5\sin\theta) \quad \text{and} \quad (-2.5\cos\theta, -2.5\sin\theta), \tag{3}$$

respectively, where $\theta\in[0, 2\pi)$ is fixed. Let φ be a level set function for Γ, and let Γ evolve with normal velocity

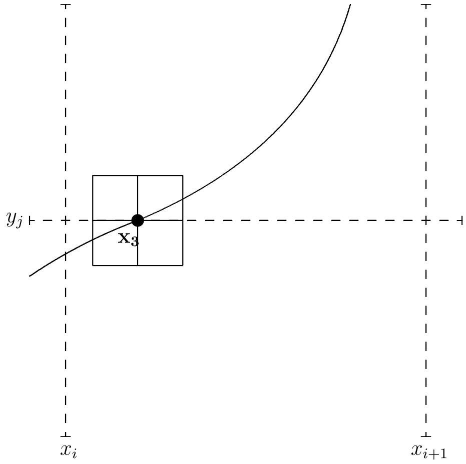

Fig. 3. The local subgrid near $\mathbf{x_3}$, with $\delta = \frac{1}{8}\Delta x$. In our tests, we used $\delta = \frac{1}{1000}\Delta x$.

$$V = 1 - \mathbf{n} \cdot [\nabla p] \quad \text{if } \varphi(x) = 0, \tag{4}$$

where $[\nabla p]$ is the jump in the pressure gradient from the inside to the outside of the drops. The pressure p solves

$$\nabla^2 p = 0 \quad \text{if } \varphi(x) < 0, \tag{5}$$

$$[p] = \kappa \quad \text{if } \varphi(x) = 0, \tag{6}$$

$$p = 0 \quad \text{if } \varphi(x) > 0, \tag{7}$$

where (6) is the Laplace–Young boundary condition, and the surface tension (the coefficient of the curvature) is non-dimensionalized to 1. The level set function φ is updated via

$$\varphi_t + V_{\text{ext}} |\nabla \varphi| = 0, \tag{8}$$

where V_{ext} is an extension of V off of Γ.

Under these equations, both circles expand outward at a constant speed of 1 since $p \equiv \frac{1}{1+t}$ inside the drops and $p \equiv 0$ outside the drops. At $t = 1.5$, the drops merge and thus become non-circular. Consequently, at this time the curvature is no longer constant along the interface and instead takes large values near the intersection. The pressure p inside the drops is then no longer constant, steep pressure gradients emerge, and the Hele-Shaw-like term of the velocity dominates near the intersection of the circles.

To study the convergence behavior of our curvature technique, we solved this example with $\theta = 13°$ on a computational domain of $[-6, 6] \times [-6, 6]$ with $\Delta x = \Delta y = 0.10$ (low resolution), $\Delta x = \Delta y = 0.05$ (medium resolution), and $\Delta x = \Delta y = 0.025$ (high resolution) using the level set/ghost fluid method as described in [7]. In Fig. 4, we show the medium-resolution results ($\Delta x = 0.05$), where the interfaces are plotted every 0.75 time units from $t = 0$ to $t = 2.25$, and the arrows indicate the direction of growth. In the left plot, we show the results when using the traditional 9-point curvature discretization. The singular curvature between the merging interfaces creates steep and noisy false pressure gradients that prevent the merger of the drops. This behavior of the traditional algorithm is also seen in the low-resolution study ($\Delta x = 0.10$), and at high resolution ($\Delta x = 0.025$), the traditional curvature algorithm becomes so inaccurate that the simulation is unable to continue past $t = 1.486$. This is a non-trivial example where the traditional curvature discretization was inaccurate and led to incorrect simulation behavior, and decreasing Δx exacerbated the problem.

On the right side of Fig. 4, we show the same simulation using our geometry-aware curvature discretization at medium resolution. (The high-resolution results are indistinguishable to graphical resolution.) Level set singularities between the merging interfaces are first detected at $t = 1.33$, and the discretization adapts accordingly. The drops merge at approximately $t = 1.48$, very close to the exact time of $t = 1.50$. Immediately after the merger, sharp cusps form in the interface that are smoothed out due to surface

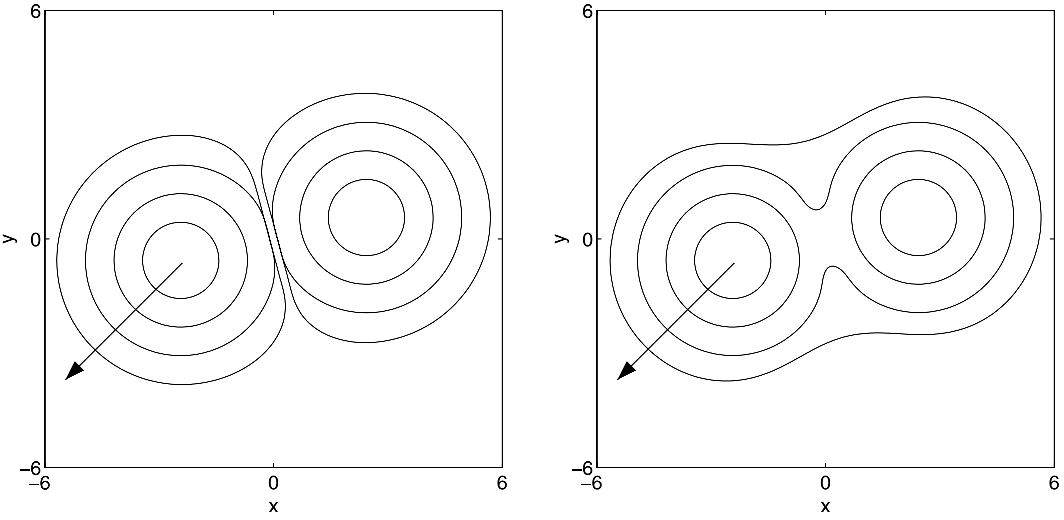

Fig. 4. Comparison of methods for merging drops under modified Hele-Shaw flow at medium resolution. Left: Traditional curvature discretization. Right: Geometry-aware curvature discretization. Times shown: $t = 0.0, 0.75, 1.5, 2.25$.

tension. This demonstrates that our geometry-aware curvature routine is robust and accurate even in situations involving interfaces with high curvature. Our curvature algorithm also performs well at low and high resolutions: the drops merge at $t = 1.47$ for the low-resolution study and at $t = 1.49$ for the high-resolution study, and the drops coalesce in a Hele-Shaw-dominated manner thereafter.

In Fig. 5, we examine the maximum curvature error for the two curvature discretizations. In the left plot, we show the maximum curvature error for the traditional curvature discretization. At low resolution (dotted curve), the error for the traditional discretization has a spike at $t = 1.32$. Afterwards, the interface flattens out in the near-contact regions, the drops fail to merge, and the simulation tends to the wrong solution. Because the drops flatten rather than merge, the computed curvature is bounded away from the correct value; consequently, the error curve levels off after this initial spike. At medium resolution (dashed curve), the occurrence of this error spike is delayed until $t = 1.39$, but the magnitude of the spike increases; after the spike, the

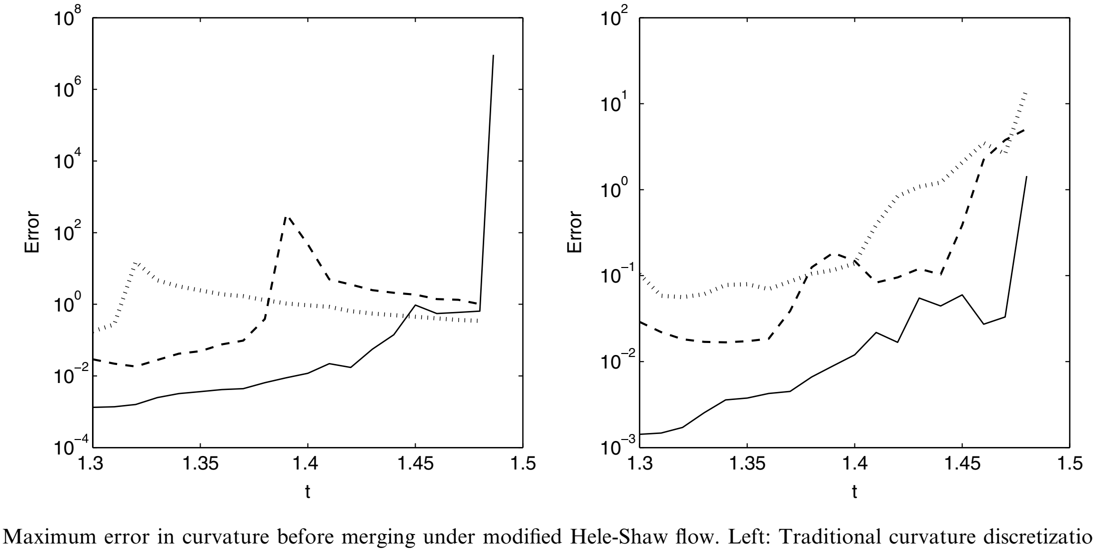

Fig. 5. Maximum error in curvature before merging under modified Hele-Shaw flow. Left: Traditional curvature discretization. Right: Geometry-aware curvature discretization. Dotted ($\Delta x = 0.10$), dashed ($\Delta x = 0.05$), solid ($\Delta x = 0.025$).

medium resolution study behaves similarly to the low-resolution study. At high resolution (solid curve), the appearance of the error spike is further delayed until $t = 1.486$, but its magnitude grows to $\sim 10^7$. Refining the computational mesh delays the occurrence but exacerbates the magnitude of the problems inherent in the traditional curvature discretization; at high resolution, the error spike is so severe that the simulation is unable to continue.

In the right plot in Fig. 5, we show the maximum curvature error for our geometry-aware discretization. The dotted curve gives the error at low resolution, the dashed curve the medium-resolution error, and the solid curve corresponds to high resolution. Overall, each mesh refinement improves the accuracy of our geometry-aware curvature discretization, and our geometry-aware method never experiences the large error spikes that characterize the traditional discretization. In the last several time steps, the interfaces are in very close contact (under two mesh lengths), making the conditions for accurately calculating the curvature very difficult. Even for these times, mesh refinement improves the accuracy.

In Fig. 6, we show the order of convergence of our geometry-aware discretization:

$$\frac{\log\left(\frac{\max\ \text{error}_{\Delta x=0.025}}{\max\ \text{error}_{\Delta x=0.050}}\right)}{\log\left(\frac{0.025}{0.050}\right)}. \tag{9}$$

Note that we obtain second-order convergence or better for almost all times. The medium-resolution study first detects level set irregularity at $t = 1.33$, and the high-resolution study detects irregularity starting at $t = 1.44$; after our algorithm detects level set irregularity, it begins to use our geometry-aware discretization. Thus, between $t = 1.33$ and $t = 1.43$, the medium-resolution study uses the geometry-aware discretization between the merging interfaces while the high-resolution study continues to use the traditional 9-point stencil. In this time interval, the interfaces approach one another, the traditional curvature discretization loses accuracy for the high-resolution study, and the order of convergence steadily falls. After $t = 1.43$, the high-resolution simulation begins to use our geometry-aware discretization, and the order of convergence is restored to second-order or better until $t = 1.48$, at which time the drops begin to merge.

In the next section, we present examples that demonstrate the behavior of our adaptive curvature algorithm in the context of solid tumor growth. We shall see that the traditional curvature discretization again becomes inaccurate, leading to incorrect predictions on the behavior of the tumor growth models. Thus, the shortcomings of the traditional curvature discretization negatively impact the scientific investigation of tumor growth.

6.2. Necrotic in vivo tumor growth

Let φ be a level set function whose zero level set denotes the boundary Γ of an avascular tumor growing into a surrounding, non-cancerous tissue. This models the early stage of in vivo growth before angiogenesis occurs. Let $R = \max\{\text{dist}(\mathbf{x}, 0) : \mathbf{x} \in \Gamma\}$, and let $D_R = \{\mathbf{x} : |\mathbf{x}| \leqslant R + 1\}$ be a region containing the tumor and

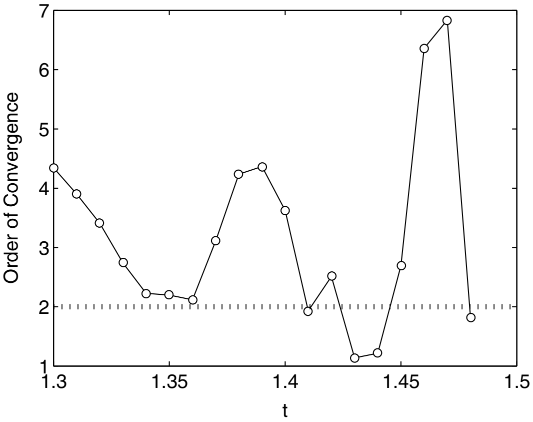

Fig. 6. Order of convergence for the geometry-aware curvature for the modified Hele-Shaw problem.

the non-cancerous tissue immediately surrounding the tumor where there is no blood vasculature. Outside D_R, the healthy tissue is assumed to have a pre-existing network of blood vessels.

Let c denote the non-dimensionalized nutrient concentration within the tumor and the surrounding tissue. Outside D_R, the blood vasculature delivers sufficient nutrient that c is constant. Within D_R, the nutrient diffuses and is consumed as it enters the tumor interior. Where the nutrient level drops below a threshold value N, the tumor cells become necrotic, start to die, and are broken down by enzymes. The proliferating tumor cells generate an internal (oncotic) pressure p that pushes the tumor boundary outward with normal velocity V via Darcy's law. The enzymatic breakdown of necrotic tumor tissue is modeled by a local decrease in the pressure that slows growth. Cell-to-cell adhesive forces are modeled by a curvature boundary condition on Γ. The non-cancerous tissue in D_R is assumed to be close enough to the tumor to be affected by the pressure changes within the tumor, and the pressure is assumed to be constant outside of D_R. Accordingly, the nutrient concentration c satisfies

$$\nabla^2 c = c \quad \text{if } \varphi(\mathbf{x}) < 0, \tag{10}$$

$$[c] = 0 \quad \text{if } \varphi(\mathbf{x}) = 0, \tag{11}$$

$$D\nabla^2 c = 0 \quad \text{if } \varphi(\mathbf{x}) > 0 \text{ and } \mathbf{x} \in D_R, \tag{12}$$

$$[c] = 0 \quad \text{if } \mathbf{x} \in \partial D_R, \tag{13}$$

$$c = 1 \quad \text{if } \mathbf{x} \notin D_R, \tag{14}$$

the oncotic pressure p is governed by

$$\nabla^2 p = G \cdot G_N \quad \text{if } \varphi(\mathbf{x}) < 0 \text{ and } c < N, \tag{15}$$

$$\nabla^2 p = -Gc \quad \text{if } \varphi(\mathbf{x}) < 0 \text{ and } c \geqslant N, \tag{16}$$

$$[p] = \kappa \quad \text{if } \varphi(\mathbf{x}) = 0, \tag{17}$$

$$\mu\nabla^2 p = 0 \quad \text{if } \varphi(\mathbf{x}) > 0 \text{ and } \mathbf{x} \in D_R, \tag{18}$$

$$[p] = 0 \quad \text{if } \mathbf{x} \in \partial D_R, \tag{19}$$

$$p = 0 \quad \text{if } \mathbf{x} \notin D_R, \tag{20}$$

and the normal velocity of the tumor boundary is given by Darcy's law:

$$V = -\mathbf{n} \cdot \nabla p \quad \text{if } \varphi(\mathbf{x}) = 0. \tag{21}$$

Here, ∇p is computed on the interior side of the tumor, G is a parameter that relates to the relative proliferation rate of the tumor cells, G_N is a parameter that governs the rate of tumor cell breakdown in necrotic regions, D is the nutrient diffusivity in healthy tissue, and μ is the cellular mobility in healthy tissue. This tumor growth model is an extension of current models given in [2,6,7,15] and will be further investigated in a future work [5]. See [14] for a different approach using diffuse interface modeling.

In Fig. 7, we solve this system with a random initial shape, $\Delta x = \Delta y = 0.08$, $G = 20.0$, $G_N = 1.0$, $D = 1.0$, $\mu = 1.0$ and $N = 0.35$. In the left column, we solve using the traditional curvature discretization, and in the right column, we use our new geometry-aware discretization. Time increases from top to bottom in 0.2 increments from $t = 0$ to $t = 0.6$. In the simulations, widespread fragmentation of the tumor occurs, and the remaining small tumor nodules move away from one another. The fragmentation is due to the combined effects of the diffusing nutrient concentration, selective proliferation in the high-nutrient regions (the nutrient is highest on ∂D_R), and the variable pressure outside the tumor. This will be explored at length in a forthcoming paper [5].

Notice that significant tumor fragmentation occurs for both curvature discretizations, but the discretizations yield significantly different results on the times of fragmentation; the shape, size, and location of fragments; and whether or not the fragments contain necrotic regions. This is important in the study of malignant tumors, where the fragmentation of tumor masses may lead to the development of metastases. Furthermore, the location and quantity of necrotic tumor cells has a great impact on the development of blood vessels in tumors (angiogenesis) [15], and so the failures of the traditional curvature discretization may lead to erroneous predictions of the morphology and the subsequent vascular development of a tumor.

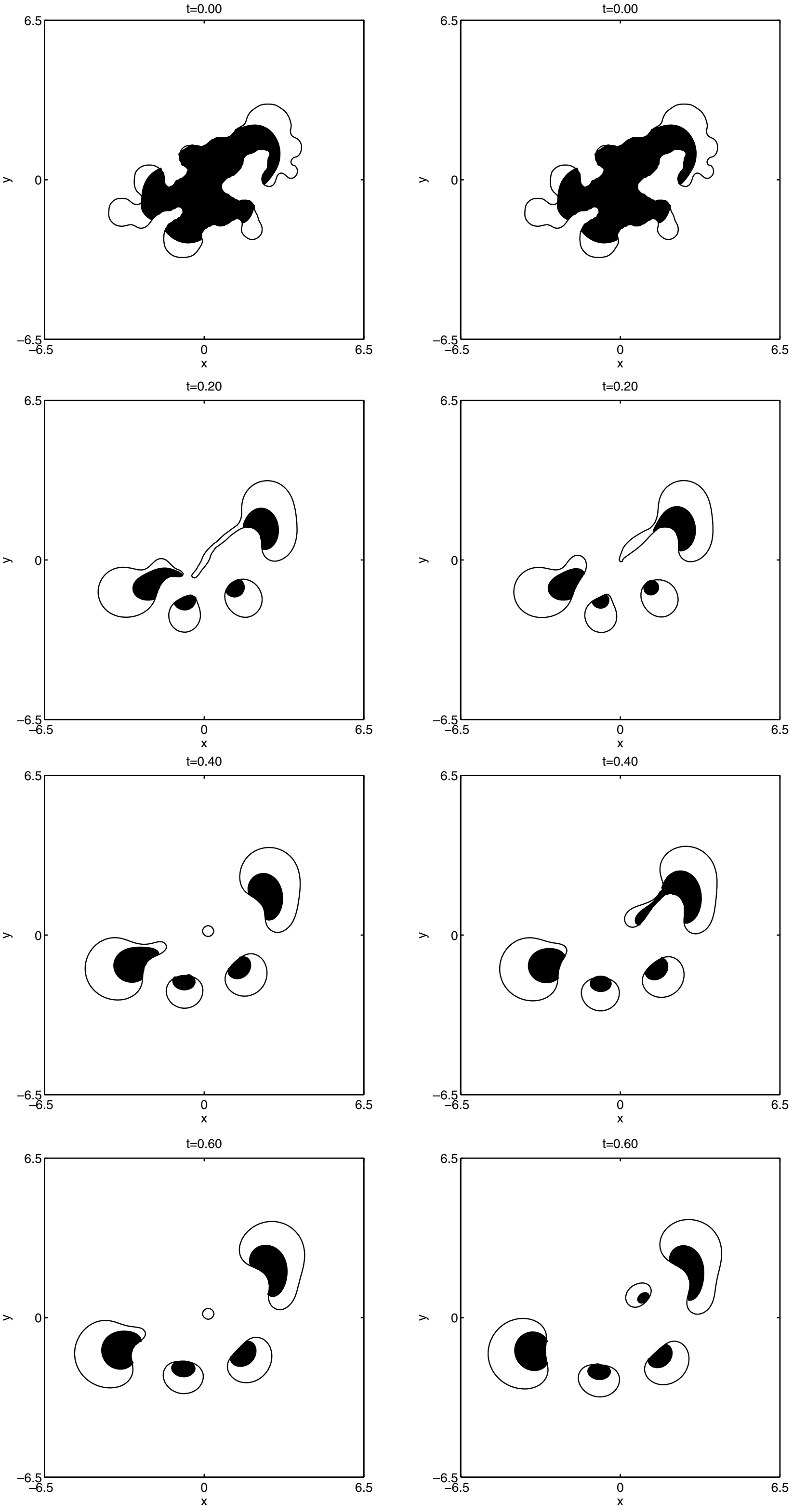

Fig. 7. Comparison of methods for necrotic in vivo tumor growth. The left column uses the traditional curvature discretization; the right column uses our new geometry-aware discretization. Time increases from top to bottom in 0.2 increments from $t = 0.0$ to $t = 0.6$. The dark regions indicate necrotic regions where the tumor cells are dying due to lack of nutrient.

7. Conclusions

We have developed an improved geometry-aware discretization of curvature for use near level set singularities. In our method, we first detect regions where the traditional curvature discretization fails. Then, we find a least squares, oriented quadratic polynomial approximation of the interface centered at the point where we desire the curvature. A local level set function is constructed, and a standard 9-point stencil is used on a local subgrid to discretize the curvature.

We have demonstrated that for complex geometries (e.g., interfaces in near contact), the traditional curvature discretization produced results that were not improved by mesh refinement, whereas our geometry-aware algorithm was second-order accurate and robust. Examples were given for modified Hele-Shaw flow and in vitro tumor growth. In the tumor growth example, it was demonstrated that an accurate and robust curvature discretization is critical for the accurate modeling of the biophysical properties of evolving tumors. Our method is generally applicable to any level set model involving morphological changes or interfaces in near contact, e.g., multiphase flows, dendritic crystal growth, and image processing.

Lastly, we note that the geometry-aware curvature discretization developed here can be extended to three dimensions by finding an approximating surface $\gamma(s_1, s_2) = (x(s_1, s_2), y(s_1, s_2), z(s_1, s_2))$ and constructing a $3 \times 3 \times 3$ local level set function. We also note that this method could be used to improve the accuracy of normal vector discretizations near level set singularities; this is currently under study.

Acknowledgements

The authors gratefully thank Vittorio Cristini and Steven Wise for enlightening discussions concerning this work. The authors also thank the Network and Academic Computing Services (NACS) at the University of California at Irvine (UCI) and the departments of mathematics and biomedical engineering at UCI for generous computing resources. We thank the National Science Foundation (mathematics division) for partial support.

References

[1] C. de Boor, A Practical Guide to Splines, Springer-Verlag, New York, 1978.

[2] V. Cristini, J.S. Lowengrub, Q. Nie, Nonlinear simulation of tumor growth, J. Math. Biol. 46 (2003) 191–224.

[3] F. Gibou, R. Fedkiw, R. Caflisch, S. Osher, A level set approach for the numerical simulation of dendritic growth, J. Sci. Comput. 19 (2003) 183–199.

[4] W. Li, S. Xu, G. Zhao, L.P. Goh, Adaptive knot placement in B-spline curve approximation, Comput. Aided Des. 37 (8) (2005) 791–797.

[5] J.S. Lowengrub, P. Macklin, Nonlinear simulation of in vitro and in vivo tumor growth, in preparation.

[6] P. Macklin, Numerical simulation of tumor growth and chemotherapy, M.S. Thesis, University of Minnesota School of Mathematics, September 2003.

[7] P. Macklin, J.S. Lowengrub, Evolving interfaces via gradients of geometry-dependent interior Poisson problems: Application to tumor growth, J. Comput. Phys. 203 (1) (2005) 191–220.

[8] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York, NY, 2002, ISBN 0-387-95482-1.

[9] S. Osher, R. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2) (2001) 463–502.

[10] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.

[11] A.E. Segall, M.J. Sipics, The influence of interpolation errors on finite-element calculations involving stress-curvature proportionalities, Finite Elem. Anal. Des. 40 (13-14) (2004) 1873–1884.

[12] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, New York, NY, 1999, ISBN 0-521-64557-3.

[13] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, Ann. Rev. Fluid Mech. 35 (2003) 341–372.

[14] S. Wise, H. Frieboes, V. Cristini, Three-dimensional simulation of the growth of a multispecies tumor via a diffuse interface approach, Bull. Math. Biol., in review.

[15] X. Zheng, S.M. Wise, V. Cristini, Nonlinear simulation of tumor necrosis, neo-vascularization and tissue invasion via an adaptive finite-element/level set method, Bull. Math. Biol. 67 (2005) 211–259.