A simple scheme for volume-preserving motion by mean curvature

Steven J. Ruuth[*] Brian T.R. Wetton[†]

September 13, 2002

Abstract

In this article, we present a diffusion-generated approach for evolving volume-preserving motion by mean curvature. Our algorithm alternately diffuses and sharpens characteristic functions to produce a normal velocity which equals the mean curvature minus the average mean curvature. This simple algorithm naturally treats topological mergings and breakings and can be made very fast even when the volume constraint is enforced to double precision (or more). Two dimensional numerical studies are provided to demonstrate the convergence of the method for smooth and nonsmooth problems.

Key words: Diffusion-generated motion, volume-preserving, mean curvature

1 Introduction

There are many phenomena in which sharp interfaces form, persist and propagate. Modeling these processes often leads to equations of motion for a surface moving with a normal speed that depends on the surface geometry.

Motion by mean curvature is one of the fundamental models for interface motion. If we consider a collection of disjoint surfaces Γ_i, then under mean curvature motion they will shrink to enclose zero volume in finite

[*]Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (sruuth@sfu.ca). The work of this author was partially supported by an NSERC Canada grant.

[†]Department of Mathematics, University of British Columbia. (wetton@math.ubc.ca). The work of this author was partially supported by an NSERC Canada Grant.

time. However, if the normal velocity is modified from the mean curvature κ to $\kappa - \kappa_a$ where κ_a is the average mean curvature[1], then the total volume enclosed by the surfaces remains constant. This *volume-preserving mean curvature motion* arises physically as a limit of a nonlocal model for binary alloys [2, 16]. Note that this motion is closely related to the general phenomena of "Ostwald ripening" (or "survival of the fattest"), which is of general interest in statistical physics, and provides a model for the dissolution-precipitation dynamics of precipitate crystals suspended in a saturated solution. Furthermore, the model has importance within the mathematical theory of curvature flows, as the simplest model problem with nontrivial limiting behavior. To avoid ambiguity with flows that preserve surface area and interface length we follow the usual convention of referring to 2D motions of the type $v_n = \kappa - \kappa_a$ as volume-preserving rather than area-preserving.

Due to the theoretical and practical interest in this underlying flow, an interesting variety of computational techniques have been brought to bear on volume-preserving motion by mean curvature.

Front tracking methods [3] for volume-preserving mean curvature flow explicitly approximate the motion of the interface rather than a level set of some higher dimensional function. These methods are especially well-suited for curves in two dimensions that never cross but are rather difficult to implement whenever complicated topological changes occur, especially in more than two dimensions. We further note that front tracking methods explicitly calculate κ and κ_a, which can be involved in three dimensions.

Phase field methods, on the other hand, automatically treat changes in topology. They have the advantage that they do not need to calculate κ or κ_a. Unfortunately these methods can be too expensive for practical computation [12] because they represent the interface as an internal layer and thus require an extremely fine mesh (at least locally) to resolve this layer. We note that a variety of advances have been made for certain curvature-dependent flows (including volume-preserving flow) by carrying out a careful dynamic mesh refinement [13]. However, these gains in efficiency naturally come at the price of additional algorithmic detail, especially in three dimensions.

It is also noteworthy that level set methods [14] have been used to evolve interfaces according to $v_n = \kappa - \kappa_a$ [15, 24]. To treat volume-preserving motion by mean curvature these methods explicitly calculate κ and κ_a. These

[1] In 3D denote surface i by Γ_i and let $|\Gamma|$ be the total surface area. Similarly, in 2D denote curve i by Γ_i and let $|\Gamma|$ be the total curve length. The average mean curvature is defined as

$$\kappa_a = \frac{1}{|\Gamma|} \sum_i \int_{\Gamma_i} \kappa dA.$$

calculations are reasonably straightforward on a uniform grid [15]. Level set methods are efficient and naturally treat topological mergers and pinch-off, but have the disadvantage that they sometimes suffer from significant volume loss in poorly resolved regions of the flow. Unfortunately, volume-preserving flow is quite naturally under-resolved since the volume of small phase regions should be accounted for as they (very rapidly) disappear. See also [4, 23] for some other interesting examples of volume-conserving motions (but with different surface dynamics) that have been treated using level set methods.

In this paper, we propose a novel algorithm for treating volume-preserving mean curvature flow which is based on the diffusion-generated motion algorithm originally proposed by Merriman, Bence and Osher [11, 12]. Our simple approach automatically evolves surfaces with a normal speed equal to $\kappa - \kappa_a$ without ever directly computing the mean curvature or the average mean curvature. Remarkably, this nonlocal motion can be produced merely by following the level set that preserves volumes in the diffusion-generated framework. Topological merger and breakage are automatically handled without any special algorithmic procedures. Accurate, efficient discretizations are possible using adaptive resolution and fast Fourier transform techniques [19]. Finally, the algorithm has the advantage that it inexpensively enforces the volume constraint to double precision (or more, if need be). We remark that some of the basic ideas in this paper first appeared as part of an earlier dissertation. See [17] for details.

The outline of the paper follows. In Section 2 we review the diffusion-generated motion by mean curvature algorithm [11, 12] and show how a simple modification gives a volume-preserving flow. Efficient discretizations are also discussed. Section 3 validates our approach with a number of numerical tests. Finally, Section 4 gives a variety of extensions and directions for future work.

2 Diffusion-Generated Motion

A semi-discrete algorithm for following interfaces propagating with a normal velocity equal to the mean curvature was introduced by Merriman, Bence and Osher [11, 12]. This algorithm has been extended to motions which also involve a curvature-independent component [11, 10]. In this section we review these basic methods and use them as building blocks for deriving volume-preserving motion by mean curvature. Fast spatial discretizations are also discussed.

2.1 Motion by Mean Curvature

The diffusion-generated motion algorithm introduced in [11, 12] is a surprisingly simple procedure for evolving the boundary of a region with a normal speed equal to mean curvature. If the initial region has characteristic function χ, the updated region at a time Δt is

$$\left\{\mathbf{x} : \chi * K(\mathbf{x}) > \frac{1}{2}\right\}$$

where K is a Gaussian of width $\sqrt{\Delta t}$ [8, 20]. "Diffusion-generated" refers to the fact that convolution with the Gaussian kernel can be realized by solving the heat equation for a time Δt, with χ as initial data. Using this observation we obtain the diffusion-generated motion by mean curvature algorithm:

ALGORITHM **DGM**
GIVEN: An initial region R.

BEGIN
  (1) Initialize: Set χ equal to the characteristic function for the region R.
  (2) Repeat for all steps:
    (a) Diffuse: Starting from χ, evolve $\bar{\chi}$ for a time Δt according to $\bar{\chi}_t = \nabla^2 \bar{\chi}$.
    (b) Sharpen: $\chi = \begin{cases} 1 & \text{if } \bar{\chi} > 1/2 \\ 0 & \text{otherwise} \end{cases}$
END

The location of the interface is given by the $1/2$ level set of $\bar{\chi}$.

Diffusion causes a curvature-dependent blurring of the set boundary and a formal analysis of the diffusion equation [11, 10, 12] shows this should result in motion by mean curvature as the time step goes to zero. In the special case of smooth interfaces, formal asymptotics show a first order convergence rate in the position of the front [18]. Moreover, rigorous proofs demonstrate that this simple algorithm converges to motion by mean curvature in rather general settings involving topological change [5, 1, 9]. We further note that diffusion-generated motion could serve as a definition of curvature motion in extremely general settings, since it applies to the boundary of any measurable set, for which curvature would otherwise have no meaning.

Diffusion-generated motion has a number of interesting extensions [22, 21]. These include a direct extension to affine motion [10, 18], anisotropic curvature motion [20, 9] and the motion of multiple junctions [11, 12, 18]. Extensions to achieve interface motions similar to the threshold dynamics type cellular automata are also available [22]. These methods are based on continuous convolutions rather than discrete sums and naturally provide a numerically and analytically tractable link between cellular automata models and the smooth features of pattern dynamics.

Particularly relevant to the derivation of volume-preserving schemes will be the the extension to affine motion described below.

2.2 Affine Motion

The extension of the ALGORITHM **DGM** to a normal velocity which equals the mean curvature plus a constant (*affine motion*),

$$v_n = a + \kappa$$

is also straightforward. This motion is obtained using the threshold value of

$$\frac{1}{2} - \frac{1}{2} a \sqrt{\frac{\Delta t}{\pi}} \tag{1}$$

instead of the usual value of 1/2 [10].

Interestingly, an expansion of the normal velocity of the front may be carried out to show that the rate of convergence is first order for smooth interfaces. See [10, 17, 18] for the relevant details.

2.3 Volume-Preserving Motion

We now provide a simple modification of diffusion-generated motion by mean curvature which generates the desired volume-preserving flow.

As is clear from the previous section, an approximation to this flow is possible by thresholding at

$$\frac{1}{2} - \frac{1}{2} \kappa_a(t) \sqrt{\frac{\Delta t}{\pi}} \tag{2}$$

rather than the usual value of 1/2. By the nature of the flow, this choice must preserve volume (to leading order). *Because there exists a unique threshold λ that preserves volume*[2] *it is clear that the desired threshold value (2) must be λ.* In smooth problems, we expect this approximation to be first order in Δt since the threshold value (1) used in this derivation gives a first order approximation to volume-preserving flow.

Applying this idea to diffusion-generated motion gives a simple algorithm for computing solutions to the nonlocal model. We just make the following replacements to step 2(b) in the Algorithm **DGM**:

[2] The diffused χ is smooth, so the volume enclosed by a level set ℓ, $V(\ell) = \text{Volume}(\{\vec{x} : \bar{\chi} < \ell\})$ is an invertible function of ℓ.

- Find the level set that preserves phase volumes; i.e., determine the value λ satisfying $V(\lambda) = V_0$ where V_0 is the initial phase volume. Solving for λ may be accomplished by a variety of line search algorithms. We have found that a particularly fast, simple and reliable approach is to use secant method with initial guesses coming from the previous search[3]. Moreover, volume calculations are trivial when a Fourier spectral discretization is used since the volume is given by the leading Fourier coefficient c_{00} (c_{000} in 3D).

- Carry out step 2(b) of the Algorithm **DGM** using the threshold value λ rather than the usual choice of 1/2.

Making these changes gives the algorithm for volume-preserving motion by mean curvature:

ALGORITHM **VP-DGM**
GIVEN: An initial region R.

BEGIN
(1) Initialize:
(a) Set χ equal to the characteristic function for the region R.
(b) Set V_0 equal to the volume of R.
(2) Repeat for all steps:
(a) Diffuse: Starting from χ, evolve $\bar{\chi}$ for a time Δt according to $\bar{\chi}_t = \nabla^2 \bar{\chi}$.
(b) Determine the threshold value that preserves the volume of the set: I.e., find a λ so that $|\text{Volume}(\{\vec{x} : \bar{\chi} < \lambda\}) - V_0| < \epsilon$.
(c) Sharpen: $\chi = \begin{cases} 1 & \text{if } \bar{\chi} > \lambda \\ 0 & \text{otherwise} \end{cases}$
END

The location of the interface is given by the λ level set of $\bar{\chi}$.

2.4 Spatial Discretization

A standard finite difference discretization of diffusion-generated motion is extremely easy to implement and is suitable for qualitative studies of certain curvature-dependent flows [11, 12] but has the disadvantages that it is susceptible to numerical artifacts [12] and is prohibitively expensive when accurate solutions are sought [19]. Fortunately, much faster results can be obtained using a simple spectral method on adaptive grids [19].

[3]The simulations detailed in Section 3 required an average of between 1.0 and 3 secant iterations per step to preserve volume to 13 decimal digits. Note that the fewest iterations were required when small time steps were applied. This make sense since the appropriate level λ varies little between successive time steps when Δt is small.

To begin, a method is needed to solve the heat equation,

$$\chi_t = \nabla^2 \chi$$

repeatedly over intervals of length Δt. Following [19], this is accomplished using a Fourier cosine tensor product. Notice that χ is initially discontinuous so it will contain a high frequency error from truncating the Fourier series. However, we only require χ after a time Δt. After a time Δt, high frequency error modes have been damped out. Since the problem is linear, the various modes do not interact. Thus *there is never a need to approximate the high frequency components of* χ. We remark that the decay of high frequency modes depends only on the timestep size—it is independent of the threshold value λ. Thus a Fourier approximation is an excellent choice, because far fewer basis functions are required than might otherwise be expected.

The thresholding step is also straightforward. Using the usual orthogonality conditions, it is easy to show that the Fourier coefficients of the characteristic function in 2D after thresholding are

$$c_{ij} = (2-\delta_{i0})(2-\delta_{j0}) \iint_{R_t} \cos(\pi i x)\cos(\pi j y)\; dxdy \tag{3}$$

where $R_t = \{\vec{x} : \chi(\vec{x}, t) > \lambda\}$ is the approximation to the phase we are following and

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases}$$

is the usual Kronecker delta function.

To complete the discretization, the integrals (3) must be evaluated. These are accurately and efficiently treated using adaptive quadrature methods with unequally spaced fast Fourier transforms. Note that both of these algorithmic components are reasonably straightforward to implement: Adaptive quadrature can be carried out using standard quadtree methods (or octrees in 3D) while unequally spaced FFT packages are available commercially or can be directly coded from recent journal articles. See [19] for details.

3 Numerical Experiments

In this section, we examine the numerical convergence rate of volume-preserving motion by mean curvature for some smooth and nonsmooth flows. In all cases the volume was preserved to 13 digits. In the smooth case, we shall compare our results against the exact (analytically derived) solution while in the nonsmooth simulation our comparisons will be against a well-resolved front tracking calculation [3].

3.1 A Smooth Test Problem

To test how well the algorithm approximates the volume-preserving flow $v_n = \kappa - \kappa_a$, we first consider the smooth motion of two circles with initial radii 0.2 and 0.15. See Figure 1. Using the ALGORITHM **VP-DGM**, the area enclosed by the smaller circle after a time $t = 0.02$ was compared to the exact answer[4] of 0.0445079. The results from a number of experiments are reported in Table I below. These suggest a first order error for this smoothly varying problem.

3.2 A Nonsmooth Test Problem

To examine the numerical convergence for an initially nonsmooth shape we consider the evolution of a square under volume-preserving motion by mean curvature (see Figure 1). Using the ALGORITHM **VP-DGM**, the maximum distance of the computed curve to the exact curve was computed after a time $t = 0.0075$. The exact curve was taken to be the result from a tracking algorithm [3] with a very fine time step and spatial discretization. The results from a number of experiments are reported in Table II below. In this nonsmooth problem, the observed convergence improves towards first order as Δt decreases.

3.3 Topological Change

Examples involving changes in topology are also naturally handled by the method. To test how well the algorithm approximates the volume-preserving flow $v_n = \kappa - \kappa_a$, we consider the nonsmooth motion of a circle and two ellipses shown in Figure 3. Using the ALGORITHM **VP-DGM**, the maximum distance of the computed curve to the exact curve was computed after a time $t = 0.0125$. Similar to the previous example, the exact curve was taken to be the result from a tracking algorithm [3] with a very fine time step and spatial discretization. Table III reports on the results from a number of experiments. Here we observe numerical convergence, although the convergence rate is unclear.

We remark that qualitative studies for much more complicated initial conditions are also possible. See, e.g., Figure 4 for a “many-bubble” simulation.

[4]This exact result was obtained by numerical integration of the coupled ODE for the cell radii since the curves stay circles as they evolve.

4 Extensions and Directions for Future Work

In this work, we have presented a diffusion-generated approach for evolving volume-preserving mean curvature flow, $v_n = \kappa - \kappa_a$. Our method naturally treats topological mergings and breakings and can be made very fast even when the volume constraint is enforced to double precision. We have carried out a number of two-dimensional convergence studies to investigate the numerical convergence of the method and note that the algorithm may be applied in three dimensions using the discretizations described in [19].

A rather interesting extension of the volume-preserving algorithm is to the motion of multiple regions or junctions. The original diffusion-generated motion by mean curvature algorithm [11, 12] was introduced to evolve (symmetric) junctions of moving surfaces with a normal speed equal to the mean curvature. It is natural to consider enforcing volume constraints on the motion (once again by a secant approach) to obtain a means for finding shapes having the least surface area enclosing prescribed volumes. A test starting from two connected squares gives the standard double-bubble shown in Figure 5. A proof that this is the correct minimizer is given in [6]. (See [7] for the 3D minimizer and [24] for a variety of related calculations using a variational level set approach.) It is interesting to note that the junction angles between different surfaces can also be varied in diffusion-generated motion [18], so minimizations of weighted surface areas enclosing prescribed volumes may also be considered using this approach.

Another class of extensions arise when the diffusion step is replaced by a convolution with a nonsymmetric kernel (cf. [20]). Well-defined volume-preserving algorithms can be obtained in a variety of ways, the most obvious of which is to sharpen according to the contour that preserves volume. We have not pursued this extension in any detail but expect that it should correspond to minimizing some anisotropic surface energy while preserving volume.

One might also contemplate achieving volume-preserving motion by mean curvature by combining the ideas of the paper with other methods for interfacial dynamics, such as cellular automata, phase field or level set methods. Note that in the level set case an accurate reinitialization step would be key to the overall success of the algorithm.

We conclude by posing a challenge to the mathematical community. Although diffusion-generated motion has been the subject of many rigorous theoretical studies [1, 5, 8, 9] a rigorous treatment of our nonlocal algorithm has proven elusive. We feel that this simple algorithm would be an excellent target for a convergence proof.

References

[1] G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM Journal on Numerical Analysis*, 32(2):484–500, 1995.

[2] L. Bronsard and B. Stoth. Volume preserving mean curvature flow as a limit of a nonlocal Ginzburg-Landau equation. *SIAM J. on Mathematical Analysis*, 28(4):769–807, 1997.

[3] L. Bronsard and B.T.R. Wetton. A numerical method for tracking curve networks moving with curvature motion. *J. Comput. Phys.*, 120(1):66–87, 1993.

[4] D.L. Chopp. Computation of self-similar solutions for mean curvature flow. *Experimental Mathematics*, 3(1):1–15, 1994.

[5] L.C. Evans. Convergence of an algorithm for mean curvature motion. *Indiana University Mathematics Journal*, 42:553–557, 1993.

[6] J. Foisy, M. Alfaro, J. Brock, N. Hodges, and J. Zimba. The standard double soap bubble in r^2 uniquely minimizes perimeter. *Pacific J. Math.*, 159(1):47–59, 1993.

[7] J. Hass, M. Hutchings, and R. Schlafly. The double bubble conjecture. *Electron. Res. Announc. Amer. Math. Soc.*, 1(1):98–102, 1995.

[8] H. Ishii. A generalization of the Bence, Merriman and Osher algorithm for motion by mean curvature. In A. Damlamian, J. Spruck, and A. Visintin, editors, *Curvature Flows and Related Topics*, pages 111–127. Gakkôtosho, Tokyo, 1995.

[9] H. Ishii, G.E. Pires, and P.E. Souganidis. Threshold dynamics type schemes for propagating fronts. *TMU Mathematics Preprint Series*, 4, 1996.

[10] P. Mascarenhas. Diffusion generated motion by mean curvature. CAM Report 92-33, University of California, Dept. of Math, Los Angeles, 1992.

[11] B. Merriman, J. Bence, and S. Osher. Diffusion generated motion by mean curvature. In J.E. Taylor, editor, *Computational Crystal Growers Workshop*, pages 73–83. American Mathematical Society, Providence, Rhode Island, 1992. Also available as UCLA CAM Report 92-18, April 1992.

[12] B. Merriman, J. Bence, and S. Osher. Motion of multiple junctions: a level set approach. *J. Comput. Phys.*, 112(2):334–363, 1994.

[13] R.H. Nochetto, M. Paolini, and C. Verdi. A dynamic mesh algorithm for curvature dependent evolving interfaces. *J. Comput. Phys.*, 123(2):296–310, 1996.

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

[15] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M.J. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.

[16] J. Rubinstein and P. Sternberg. Nonlocal reaction-diffusion equations and nucleation. *IMA J. Appl. Math.*, 48:248–264, 1992.

[17] S.J. Ruuth. *Efficient Algorithms for Diffusion-Generated Motion by Mean Curvature*. PhD thesis, University of British Columbia, Vancouver, Canada, 1996.

[18] S.J. Ruuth. A diffusion-generated approach to multiphase motion. *J. Comput. Phys.*, 145:166–192, 1998.

[19] S.J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *J. Comput. Phys.*, 144:603–625, 1998.

[20] S.J. Ruuth and B. Merriman. Convolution generated motion and generalized Huygens' principles for interface motion. *SIAM Journal on Applied Mathematics*, 60(3):868–890, 2000.

[21] S.J. Ruuth and B. Merriman. Convolution-thresholding methods for interface motion. *J. Comput. Phys.*, 169:678–707, 2001.

[22] S.J. Ruuth, B. Merriman, and S. Osher. Convolution generated motion as a link between cellular automata and continuum pattern dynamics. *J. Comput. Phys.*, 151:836–861, 1999.

[23] M. Sussman and S. Uto. A computational study of the spreading of oil underneath a sheet of ice,. CAM Report 98-32, University of California, Dept. of Math, Los Angeles, 1998.

[24] H.-K. Zhao, B. Merriman, S. Osher, and L. Wang. Capturing the behavior of bubbles and drops using the variational level set approach. *J. Comput. Phys.*, 143:495–518, 1998.

Tables

Δt	*Error*	*Conv. Rate*
0.002	-0.003108	-
0.001	-0.001527	1.03
0.0005	-0.000733	1.06
0.00025	-0.000358	1.04
0.000125	-0.000177	1.02

Table I. Errors for the smooth test problem.

Tables (cont)

$\Delta t/0.001875$	*Error*	*Conv. Rate*
1.0000	0.000613	-
0.5000	0.000382	.68
0.2500	0.000228	.75
0.1250	0.000129	.82
0.0625	0.000070	.88

Table II. Errors in the position of the front for the square test problem.

Tables (cont)

$\Delta t/1.5625\mathrm{e}-05$	*Error*	*Conv. Rate*
1.0000	0.0241	-
0.5000	0.0207	.219
0.2500	0.0174	.251
0.1250	0.0136	.355
0.0625	0.0105	.373

Table III. Errors in the position of the front for the merging test problem.

Figure Captions

Figure 1: Initial conditions for a smooth test problem. The solution here remains as two circles and thus the evolution is described by a system of ODEs for the radii.

Figure 2: Evolution of a square under volume-preserving motion by mean curvature.

Figure 3: A merging test problem for the volume-preserving algorithm.

Figure 4: A many-bubble evolution using the volume-preserving algorithm. We take the time step to be $\Delta t = 0.000003125$ in this simulation.

Figure 5: The steady-state double bubble as captured by the method.

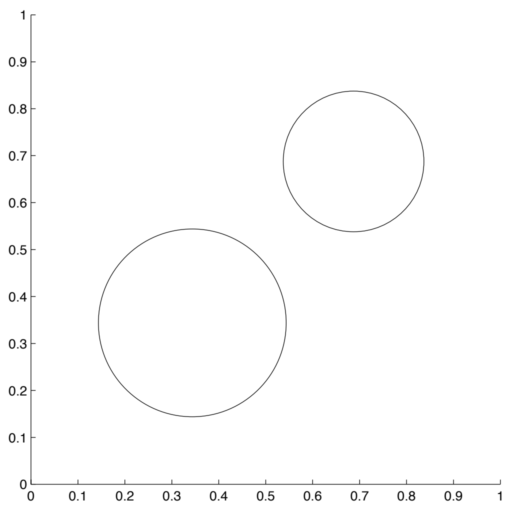

Figure 1:

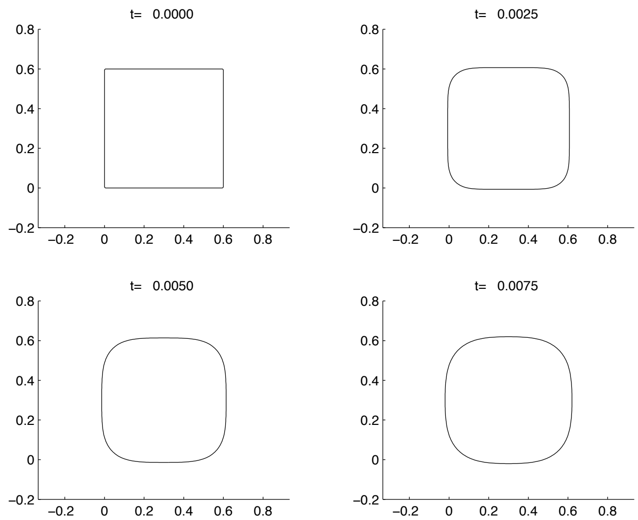

Figure 2:

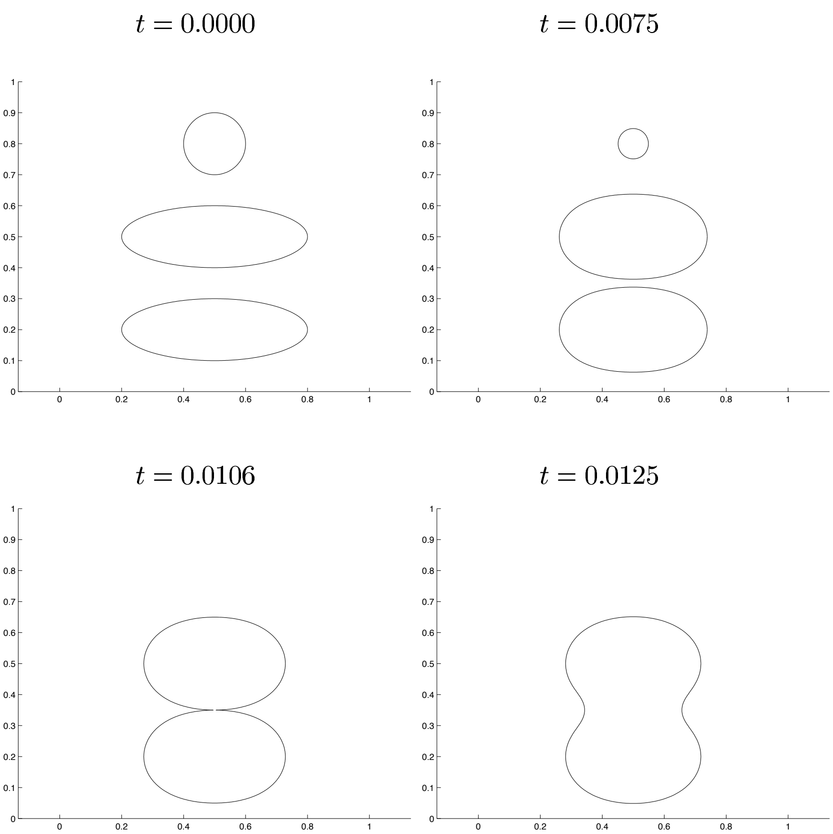

Figure 3:

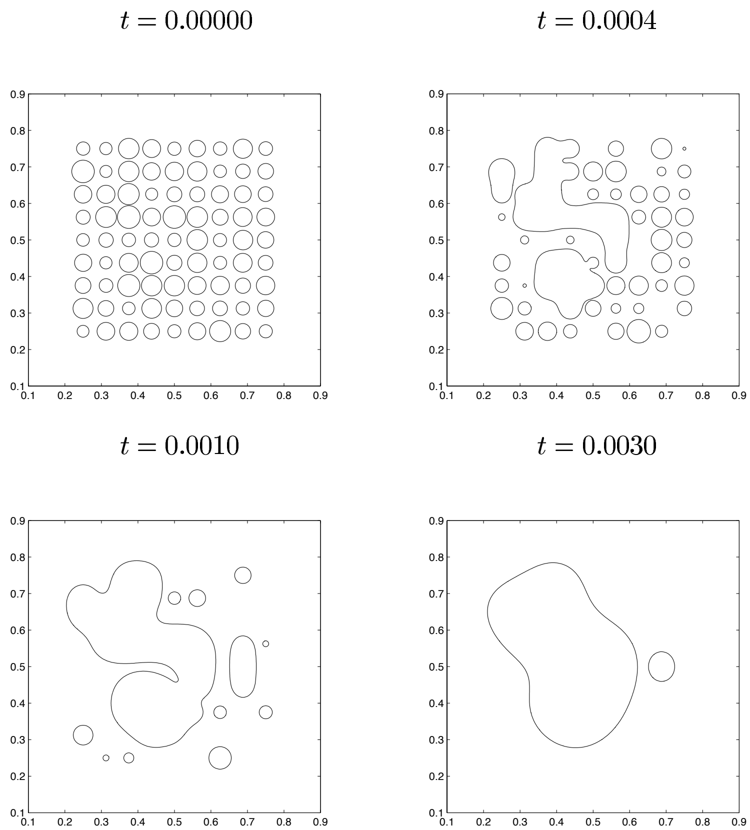

Figure 4:

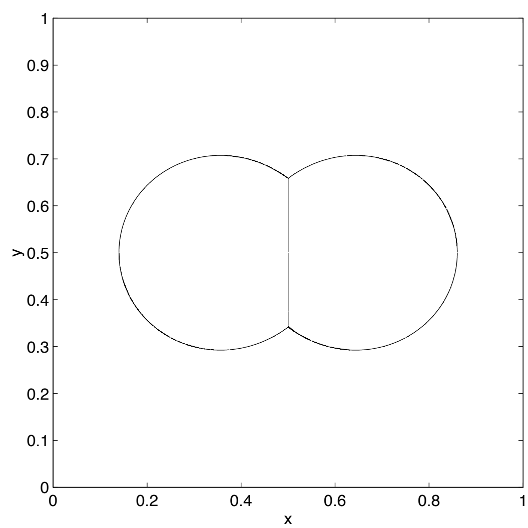

Figure 5: