The Voronoi Implicit Interface Method for computing multiphase physics

Robert I. Saye[a,b] and James A. Sethian[a,b,1]

[a]Department of Mathematics; and [b]Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720

Edited by Leslie Greengard, New York University, New York, NY, and approved October 11, 2011 (received for review July 21, 2011)

We introduce a numerical framework, the Voronoi Implicit Interface Method for tracking multiple interacting and evolving regions (phases) whose motion is determined by complex physics (fluids, mechanics, elasticity, etc.), intricate jump conditions, internal constraints, and boundary conditions. The method works in two and three dimensions, handles tens of thousands of interfaces and separate phases, and easily and automatically handles multiple junctions, triple points, and quadruple points in two dimensions, as well as triple lines, etc., in higher dimensions. Topological changes occur naturally, with no surgery required. The method is first-order accurate at junction points/lines, and of arbitrarily high-order accuracy away from such degeneracies. The method uses a single function to describe all phases simultaneously, represented on a fixed Eulerian mesh. We test the method's accuracy through convergence tests, and demonstrate its applications to geometric flows, accurate prediction of von Neumann's law for multiphase curvature flow, and robustness under complex fluid flow with surface tension and large shearing forces.

multiple interface dynamics | level set methods | foams | minimal surfaces

There are a host of physical problems that involve interconnected moving interfaces, including dry foams, crystal grain growth, mixing of multiple materials, and multicellular structures. These interfaces are the boundaries of individual regions/cells, which we refer to as phases. The physics, chemistry, and mechanics that drive the motion of these interfaces are often complex, and include topological challenges when interfaces are destroyed and created.

Producing good mathematical models that capture the motion of these interfaces, especially at degeneracies, such as triple points and triple lines where multiple interfaces meet, is challenging. Building robust numerical methods to tackle these problems is equally difficult, requiring numerical resolution of sharp corners and singularities, and recharacterization of domains when topologies change. A variety of methods have been proposed to handle these problems, including (*i*) front tracking methods, which explicitly track the interface, modeled as moving segments in two dimensions and moving triangles in three dimensions, (*ii*) volume of fluid methods, which use fixed Eulerian cells and assign a volume fraction for each phase within a cell, (*iii*) level set methods (1), which use an implicit formulation to represent the interface, and treat each region/phase separately, followed by a repair procedure which reattaches the evolving regions to each other (1–3), and (*iv*) diffusion generated motion which combine diffusion via convolution with reconstruction procedures to simulate multiphase mean curvature flow (4). Although there are advantages and disadvantages to each of these approaches, it has remained a challenge to robustly and accurately handle the wide range of possible motions of evolving, highly complex interconnected interfaces separating a large number of phases under time-resolved physics.

In this paper, we present a numerical method for tracking the interface in general multiphase problems. We call this method the Voronoi Implicit Interface Method (VIIM), because it relies on a robust interaction between Voronoi diagrams and implicit interface methods. The method has a variety of important features, including:

- Efficiency and consistency: The method uses a fixed Eulerian mesh, and a single function plus an indicator function to track the entire multiphase system. Phases are coupled together in a consistent fashion, with no gaps, overlaps, or ambiguities.
- Multiple junctions and topological change: Triple points and triple lines, where more than two phases touch, as well as breakage, merger, creation, and disappearance of phases, are all handled naturally. Transition events occur automatically, with no special attention paid to discontinuous topological change.
- Coupling with time-dependent physics: The method uses a physical time step, and complex physics may be solved at each time step and correctly incorporated into the interface evolution. Feedback from the physics affects the interface, and changes in the interface affects the physics.
- Accurate calculation of geometric quantities and extension to any dimension: The method allows one to accurately calculate curvature and geometric constraints as part of the interface evolution, and is fundamentally unchanged, regardless of the dimension of the problems.

After a short background about implicit interface level set methods, we introduce the basic VIIM. We next summarize some of the convergence tests, and then demonstrate the basic method on a collection of geometrical and fluid mechanics problems.

The Voronoi Implicit Interface Method

Background. We begin by recalling the basics of level set methods, introduced by Osher and Sethian (5), which rely in part on the theory of curve and surface evolution given in refs. 6 and 7, and the link between front propagation and hyperbolic conservation laws discussed in ref. 8. Start with an interface Γ separating two phases A and B, and moving with a given speed F in its normal direction. We can interpret this interface as an $n-1$ dimensional hypersurface in $\mathbb{R}^n$, and then embed this interface as the zero level set of a signed distance function $\phi(x)$, so that $\phi(x)$ is the initial (signed) distance to the interface Γ, chosen to be positive inside one phase and negative inside the other. The zero level set $\{\phi = 0\}$ corresponds to the original interface Γ separating A and B. Adding time, we can ensure that the zero level set of this function ϕ always corresponds to the evolving interface Γ through the initial value partial-differential equation given by

$$\phi_t + F|\nabla\phi| = 0.$$

Here, the speed function F may depend on a variety of factors, including local geometry (normal direction and curvature), integral properties (enclosed area/volume), as well as the solution

Author contributions: R.I.S. and J.A.S. designed research, performed research, analyzed data, and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence should be addressed. E-mail: sethian@math.berkeley.edu.

of complex partial differential equations (PDEs), such as fluid flow, material elasticity, diffusion, etc., with jump conditions and source terms supplied by the interface position.

This initial value PDE is approximated using upwind finite difference schemes: In the original formulation presented in ref. 5, the interface is embedded throughout the entire computational domain and hence adds unnecessary labor. More sophisticated versions employ the Narrow Band Level Set Method (9), which adaptively focuses computation in a small thin tube around the moving interfaces, and hence reduces the complexity of the algorithm to roughly the number of elements on the front. For details and a comprehensive review on the level set method, see ref. 1.

Consider now three different regions, or phases, A, B, and C. The standard level set construction, which relies on a signed distance function whose zero level set is the interface of interest, breaks down at triple points where three interfaces meet, because there is no longer an inside and an outside. Various authors have approached this problem by using multiple level set functions, in most cases by simply dividing the region into three separate level set functions, namely, inside A and not inside A, inside B and not inside B, etc., advancing each level set function separately. Although more economical classifications can be formed, they all boil down to using multiple functions in multiple regions.

Voronoi Sets. Given a set of m nodes in $\mathbb{R}^n$, the Voronoi diagram is the decomposition of $\mathbb{R}^n$ into different cells such that all the points in a given cell are closer to one node than any other. The boundaries of these cells are then a set of points, which are equidistant from at least two nodes. This construction may be extended to a set of regions $A_1,\ldots,A_m$ instead of nodes, which may be curves, surfaces, or any other objects: The tessellation of the domain into cells still provides a Voronoi subdivision such that all points in a particular cell are closer to region A_i than to any other, and the boundaries of these cells are equidistant from at least two regions.

Using this idea, we now formulate the VIIM, which is a PDE-based method based on a single function, defined on a fixed Eulerian background mesh, and tracks evolving interface problems of multiple regions in two and three dimensions, regardless of the complexity of the multiply-connected junctions.

The Voronoi Implicit Interface Method. Given a domain in $\mathbb{R}^n$, we imagine a collection of phases such that each point x in the domain is either in a unique phase, or on the boundary between two or more phases: This boundary (which is really a collection of boundaries) will be known as the interface. These are the initial conditions for the problem; we further assume that we are given a speed F (or velocity field $\mathbf{u}$) defined on the interface.

At each point x of the domain, we let $\phi(x)$ be the distance from x to the closest point on the interface: $\phi(x)$ is the embedding of the interface as the zero level set of an unsigned distance function. Furthermore, we create an extension velocity, which defines a speed function $F_{\mathrm{ext}}(x)$ in all of the domain. This extension velocity can be constructed so that it smoothly varies away from the interface itself.

The key idea is to rely on a main feature of level set methods: The level set corresponding to the interface is embedded in a family of nearby level sets. Thus, the motion of the zero level set corresponding to the interface is bracketed by the motion of surrounding level sets. This property is an immediate consequence of a comparison theorem under suitable restrictions on the speed function that moves the neighboring level sets, and these restrictions are often satisfied through the construction of extension velocities as outlined in ref. 10.

We utilize this property to construct the VIIM. The motion of the interface in a multiphase system is determined by nearby level sets, given by the ϵ-level sets of ϕ, where $\epsilon > 0$. Although the interface where multiple phases touch may have high-order junctions such as triple points and triple lines, the ϵ-level sets are hypersurfaces, which exist solely in a single phase and do not contain such degeneracies. The motion of these nearby interfaces controls the evolution of the multiphase interface through the Voronoi construction, as follows.

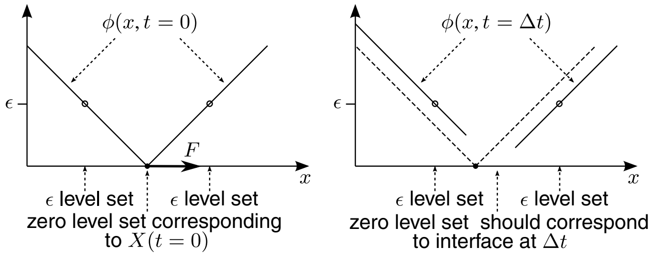


Fig. 1. Evolution of an unsigned distance function ϕ for an interface in one dimension.

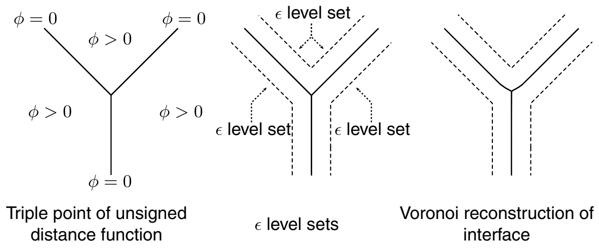
The VIIM method consists of three steps. First, we evolve the unsigned distance function a small time step Δt by solving the governing PDE. For a short period of time, the neighboring level sets will remain nice hypersurfaces. Next, we reconstruct the interface after time Δt as the Voronoi interface of the nearby ϵ-level sets. Obviously, Δt and the choice of the value of ϵ are linked: We need to choose ϵ large enough (or conversely, Δt small enough) so that the ϵ-level sets remain hypersurfaces during the time step. Finally, we use the Voronoi interface to construct a new unsigned distance function at time Δt.

Two Illustrative Examples. We illustrate with two simple examples. Imagine an interface (Fig. 1) in one spatial dimension, which is a single point located at $x = X(t)$ at time t, moving with speed $F(x,t)$, thus $\frac{dX(t)}{dt} = F(X(t),t)$. At $t = 0$, the zero level set of the unsigned distance function ϕ corresponds to the initial location of the point, and is at an extremum of ϕ. However, at the nearby level sets with value ϵ, the distance function is smooth. Hence, we can update the distance function everywhere away from the zero level set in a straightforward manner. What remains is to obtain a suitable definition of the zero level set at time Δt in such a manner that it corresponds to the location of the front at this time. We can do so by defining the zero level set as the point equidistant from the ϵ level set from each side: This single point is the Voronoi set.

In higher dimensions, and in the presence of high-order junctions, this same construction works. In Fig. 2 (*Left*) we see a triple point of an unsigned distance function, which is nonnegative everywhere. In Fig. 2 (*Middle*) we show the ϵ-level sets, and, Fig. 2 (*Right*), the Voronoi interface reconstructed from those ϵ-level sets.

Algorithm and Implementation

Given a multiphase problem with N different phases, our goal is to transform the above into a numerical method. Begin by defining the unsigned distance function $\phi(x)$, which is zero on the boundary of each phase, and measures the positive distance



Triple point of unsigned distance function

ϵ level sets

Voronoi reconstruction of interface

Fig. 2. Reconstruction of a triple point from an unsigned distance function and the Voronoi interface.

to the boundary everywhere else. Let there also be an indicator function $\chi(x)$, which indicates the phase in which the point x is located. Let us further suppose that we are given a speed function F, defined in all of space (and which can depend on position, normal, and curvature, as well as associated physics.) We fix ϵ and Δt. Our goal is to execute the following sequence:

1. Advance the unsigned level set distance function in time by solving the standard level set equation, using forward Euler with a time step Δt depending on F.
2. Reconstruct the unsigned distance function ϕ at the new time step, using the Voronoi interface reconstructed from the ϵ-level sets and the old indicator function χ.
3. Update the speed function F to reflect the appropriate physics, mechanics, etc., by solving the relevant equations of motion. This calculation may involve computing geometric quantities from the Voronoi interface, as well as extracting jump conditions, source terms, etc., from the Voronoi interface as input to PDE solvers, as well as building extension velocities to define the speed within the domain. Loop to 1.

This formulation, by construction, cannot create overlaps or vacuums: Each point in the domain is part of a unique phase at all times. Our reconstruction differs from existing algorithms in that there is neither a no overlap/vacuum condition, nor is there a penalty term that penalizes region of vacuums or overlap. The interface is always well-defined at every time step. Furthermore, the time t in the above algorithm has physical meaning: The interface moves under physical speed F during the time step Δt.

The key to the above is an accurate and fast way to execute step 2, namely to construct the new unsigned distance function from the ϵ-level sets. This procedure involves the computation of distances from the ϵ-level sets on an Eulerian mesh, and so the core tool is an Eikonal equation solver. We make extensive use of the accurate and fast Eikonal reinitialization algorithm developed by Chopp (11), which utilizes bicubic [tricubic in three dimensions (3D)] interpolation to accurately initialize an initial band. In more detail, if a grid cell is identified to contain the zero level set of a function ψ (which need not be a distance function), then ψ is interpolated in that cell using a bicubic/tricubic patch. A Newton solver is then used to find the closest point on the interpolated interface, from which the exact distance to the interpolated interface is computed at the nodes of all such grid cells. Note that this construction does not require explicit construction of the zero level set of ψ. This procedure creates a small initial band that can be input to the efficient Fast Marching Method (12), which is a Dijkstra-like ordered upwind finite difference scheme for solving the Eikonal equation outside this initial band. A different Dijkstra-like control theoretic discretization of the Eikonal equation stemming from optimal control was developed in ref. 13, and we refer the reader to ref. 14 for a detailed discussion and extensions of Fast Marching Methods to more general front propagation problems.

VIIM uses this methodology as follows. First, without explicit construction, by locating cells containing the ϵ-level sets and building bi/tricubic patches, we solve the Eikonal equation with zero boundary value on these sets. With this solution, we locate cells containing the Voronoi set, build bi/tricubic patches, and again solve the Eikonal equation to provide the new unsigned distance function.

Convergence Tests

An extensive convergence analysis of the VIIM has been performed, analyzing convergence for various values of time step Δt, grid size h, and choice of parameter ϵ. One may fix ϵ from the outset, and study convergence as the grid size h vanishes. Here, we instead couple ϵ to the grid size and choose $\epsilon = \alpha h$, where α is a constant. Generally speaking, larger values of ϵ give larger numerical errors, but with judicious programming, α can be taken as any nonnegative number. Here, we summarize some of the key findings.

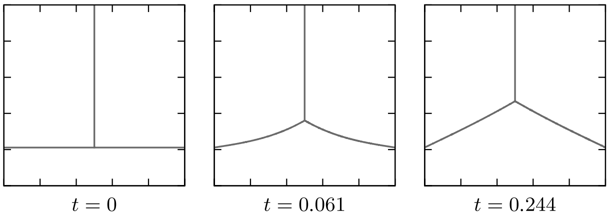


Fig. 3. Evolution of anchored Y-junction.

Convergence Tests for Smooth Interfaces. Consider a problem involving a single interface separating two phases, for example, a circle collapsing under its curvature. In this case, no corners occur in the moving interface. Using second-order finite difference schemes in space and second-order in time, the VIIM gives second-order convergence to the known exact solution.

Convergence Test of T-Junction to Y-Junction. For multiphase problems with more than two phases, and in the presence of triple points, etc., exact solutions are less well-known, and hence we rely on grid convergence. We first analyze the motion of a single T-junction that evolves into a Y-junction under curvature flow, that is, $F = \kappa$. Suitably interpreted, it is well-known that curvature flow applied to a triple point results in 120° angles: This property is known as Young's law, and arises as a natural consequence of interpreting curvature flow as minimizing length (surface area in 3D). We consider Dirichlet boundary conditions, in which the junction is anchored at the boundary of the domain $[0,1]^2$ in two dimensions, and evolve over a time interval $0 \leq t \leq T$. In Fig. 3, we show a snapshot of the evolution, which shows that the junction develops 120° angles, and ultimately converges to a triple point with three straight lines. In Table 1, we show convergence results using grid refinement for various grid sizes, coupling the choice of ϵ to h, where h is the size of one grid cell of a uniform Cartesian grid. Here, $d(\Gamma_1,\Gamma_2) := \frac{1}{T}\int_0^T d_H(\Gamma_1(t),\Gamma_2(t))dt$ is a metric using the Hausdorff distance d_H in space and measures the convergence of the interface evolution in both space and time. The results show first-order convergence. It is important to note that first-order convergence at triple points is probably all that can be expected in this framework.

Convergence Test Using von Neumann–Mullins' Law. Imagine now a large number of phases with a connected network of interfaces and many triple points. von Neumann and Mullins, under the assumption of Young's law (15), showed that if the speed of the interface is a constant multiple γ of curvature,* then the rate of change of area of a particular phase depends on its number of sides, namely

$$\frac{dA}{dt} = 2\pi\gamma\left(\frac{n}{6} - 1\right).$$

We use this result to further test the convergence of the VIIM. Start with 25 randomly positioned phases and apply curvature flow ($\gamma = 1$, grid size 256×256) with periodic boundary conditions (Fig. 4). Each phase grows, shrinks, or conserves its area, with the rate of area change a constant given by the above law. When phases change neighborship with other phases or collapse, the topology of the network changes and may alter the number of

*von Neumann's 1951 paper (16) discussed the rate of change of area as a function of the number of sides due to gas diffusion, and because pressure is taken as constant, the boundaries become circular arcs. Mullins' 1956 paper (17) studied the issue for metal grains, in which the movement of boundaries is governed by local conditions, and produced the same law, without the assumption of circular arcs.

Table 1. Convergence results for the curvature motion in Fig. 3, anchored boundary conditions

	$\epsilon = 2h$		$\epsilon = 4h$		$\epsilon = 6h$	
h	d_h^ϵ	Rate	d_h^ϵ	Rate	d_h^ϵ	Rate
1/64	0.01092	–	0.01626	–	0.01868	–
1/128	0.00631	0.8	0.01019	0.7	0.01322	0.5
1/256	0.00336	0.9	0.00549	0.9	0.00737	0.8
1/512	0.00174	1.0	0.00283	1.0	0.00383	0.9

Here $d_h^\epsilon{:=}d(\Gamma_h^\epsilon,\Gamma_{2h}^\epsilon)$ measures the difference in interface evolution (in space and time) on two successive grid sizes.

sides of each phase. Throughout the entire evolution, it follows that the area of each phase is a piecewise linear function of time. In Fig. 4 we plot the area of a selected set of nine phases. Our results show correct match with von Neumann–Mullins' law throughout the evolution.

Three-Dimensional Convergence Tests. Under curvature flow with Neumann boundary conditions, Fig. 5 shows the evolution of a 3D analogue of a T-junction, which has four different triple lines and a quadruple point. Qualitatively, we see that the surfaces make 120° angles at triple lines, which is one of Plateau's laws on the shapes of soap bubbles in a foam. Using grid refinement to measure convergence, Table 2 shows that the VIIM converges at first order, in both space and time.

Efficiency. Formally, the operation count on the method per time step is $\mathcal{O}(kN \log N)$, where N is the number of grid cells containing the interface, and k is the size of the narrow band. The Voronoi reconstruction takes twice as long as the reinitialization step in the narrow band level set method, because there are two Eikonal solves.

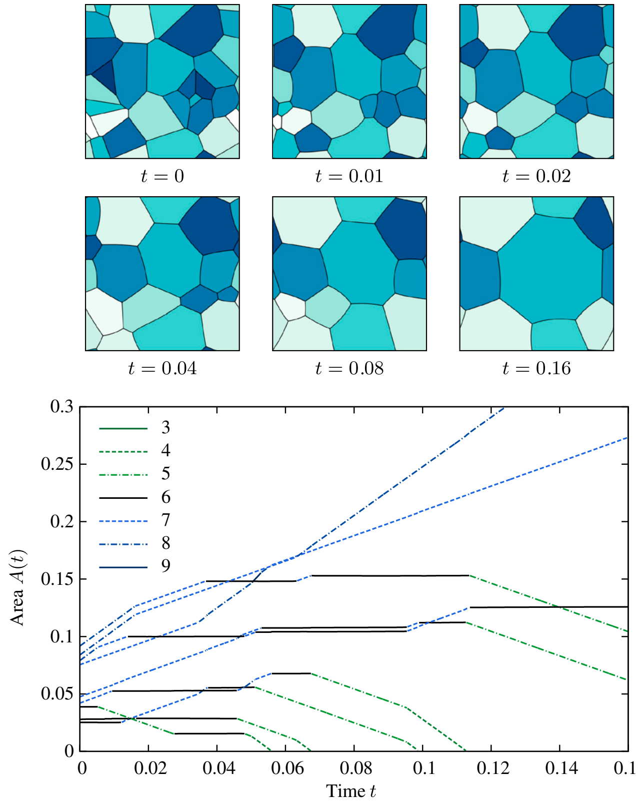


Fig. 4. (*Top*) Curvature flow applied to an initial set of 25 randomly positioned phases. (*Bottom*) Area as a function of time for a selected set of nine phases. According to von Neumann–Mullins' law, the area of a particular phase should be a piecewise linear function, with a derivative that is an affine function of the number of sides of the phase. We have, therefore, colored each part of the trajectories by the number of sides the phase had at that particular time. The slopes of the trajectories is exactly what is predicted by von Neumann–Mullins' law.

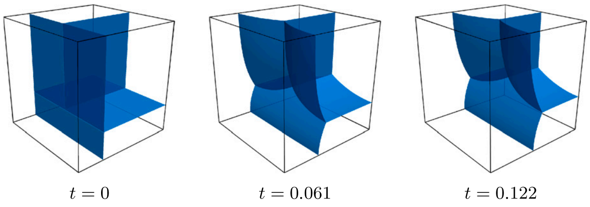


Fig. 5. Evolution of a 3D analogue of a T-junction, moving under curvature flow with zero Neumann boundary conditions.

Geometric Flows with Constraints

By adding a discontinuous source term to the right-hand side of the mean curvature flow equation, we can simulate mean curvature flow with volume conservation. The modified forward Euler step is

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \gamma\kappa^n|\nabla\phi^n| + s^n|\nabla\phi^n|,$$

where

$$s^n(x) = \frac{V_i^0 - V_i^n}{A_i^n \Delta t}, \quad \text{where } x \text{ is in phase } i.$$

Here V_i^n denotes the volume [area in two dimensions (2D)] of phase i at time step n, V_i^0 is the initial volume (area) of phase i, and A_i^n is its surface area (perimeter in 2D) at time step n. In effect, the source term $s^n|\nabla\phi^n|$ grows or shrinks each phase equally around its boundary by an amount that corrects for any mass loss/gain.

Despite each phase potentially growing or shrinking at different rates, the VIIM robustly and smoothly handles this discontinuity and conserves volume almost exactly. Fig. 6 demonstrates the method in 2D and 3D on a set of 100 initially random phases (using zero Neumann boundary conditions). Mean curvature flow minimizes the total length (surface area in 3D) of all interfaces, which, subject to the constraint of area (volume) conservation, eventually attains an equilibrium (Fig. 6). Various topological changes occur, and at all times, triple points (lines) have 120° angles.

Fluid Flow with Permeability

We now use the VIIM to incorporate fluid dynamics into the dynamics of dry foams. A foam is a collection of gas bubbles separated by a liquid component and is considered dry when the liquid makes up less than 10% of the total volume. In the multiphase evolution of dry foams, the following occurs:

- Fluid mechanics of the gas drives the flexible membranes. The flow is incompressible, with jumps in pressure across the membrane due to surface tension effects. Membranes meet at 120° junctions: triple points in 2D and Plateau borders in 3D.
- Membranes may be permeable to gas, which diffuses through the boundary across phases, leading to diffusive coarsening. In general, phases with a large number of faces grow whereas

Table 2. Convergence results for the flow illustrated in Fig. 5

	$\epsilon = 2h$		$\epsilon = 4h$		$\epsilon = 6h$	
h	d_h^ϵ	Rate	d_h^ϵ	Rate	d_h^ϵ	Rate
1/64	0.02705	–	0.05866	–	0.16831	–
1/128	0.01396	1.0	0.02469	1.2	0.03578	2.2
1/256	0.00709	1.0	0.01225	1.0	0.01722	1.1

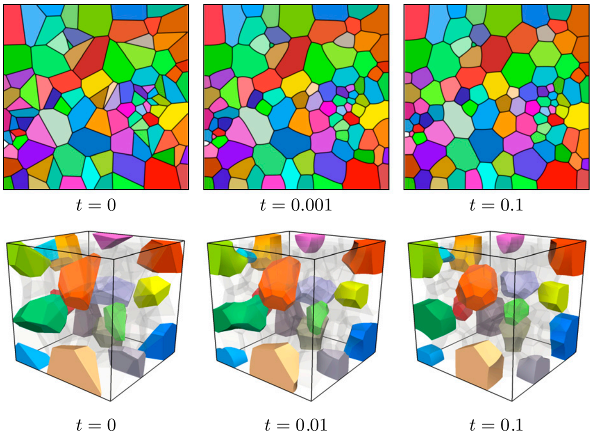


Fig. 6. Mean curvature flow ($\gamma = 1$) with area (volume) conservation on a set of 100 initially random phases (zero Neumann boundary conditions). By the time $t = 0.1$, the solution has approximately attained equilibrium. (*Top*) Two dimensions, simulation computed on a 256×256 grid on the domain $[0,1]^2$. (*Bottom*) Three dimensions, computed on a 128^3 grid in a unit cube (subset of phases shown).

> those with a small number of phases shrink. Once these phases disappear, the topology of the interconnected network changes, leading to different growth laws for each phase.

We consider a collection of membranes, moving under the combined effects of surface tension, fluid mechanics, and permeability. We assume that membranes are massless and thin, and follow the model used in ref. 18, in which the gas is modeled as an incompressible Newtonian fluid, satisfying the Navier–Stokes equations with density ρ and viscosity μ. Surface tension at the interface induces a fluid flow in addition to any external forcing flow, which in turn moves the interface, providing a feedback mechanism between membrane dynamics and fluid physics. In more detail, interface surface tension induces a pressure jump of $[p] = \sigma\kappa$, where σ is the surface tension coefficient and κ is the mean curvature of the interface. In the absence of permeability, the interface is advected by the velocity $\mathbf{u}$ of the gas, taken as continuous across the interface. With permeability, the diffusion rate of gas per unit length is proportional to the pressure difference (15). As in ref. 18, this permeability is modeled with a slip of the interface in the normal direction relative to $\mathbf{u}$, hence $\mathbf{u} - M\sigma\kappa\mathbf{n}$, where $M \geq 0$ is a constant, denoting the amount of permeability and $\mathbf{n}$ is the normal unit of the interface (with the same orientation as that used to calculate κ). We adopt a continuum surface tension model, whereby surface tension at the interface becomes a body force through the use of a Dirac delta function. The governing equations of motion are therefore

$$\rho[\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u}] = -\nabla p + \mu\Delta\mathbf{u} + \mathbf{st} + \mathbf{F},$$

$$\nabla \cdot \mathbf{u} = 0,$$

$$\phi_t + \mathbf{u} \cdot \nabla\phi = M\sigma\kappa|\nabla\phi|,$$

where $\mathbf{F}$ is any additional body forces (such as gravity) and $\mathbf{st}$ is a surface tension body force appropriate to a multiphase system. For two-phase fluid flow, this term takes the form $\mathbf{st} = -\sigma\kappa\delta(\phi)\nabla\phi$, where ϕ is a signed level set function. In our case of multiple phases which meet at triple junctions (and quad points in 3D), curvature needs to be suitably defined. It is both physically and mathematically natural to take the same definition applied to each phase (gas bubble) separately, sum all of these, and normalize by a factor of two. This formulation is physically consistent, because each bubble is separated from the others by a thin membrane, and it is mathematically natural because the resulting formula effectively enforces Young's law at triple points, i.e., triple junctions instantaneously obtain 120° angles. Using this definition, we have $\mathbf{st} = -\frac{\sigma}{2}\sum_{i=1}^{N}\kappa(\phi_i)\delta(\phi_i)\nabla\phi_i$, where ϕ_i is a signed level set function for phase i and N is the total number of phases. Although the normal is not well-defined at triple junctions, the curvature times the normal, $\kappa(\phi_i)\nabla\phi_i$, is well-defined as a distribution, and is a Dirac delta function with magnitude related to the angle of the corner.

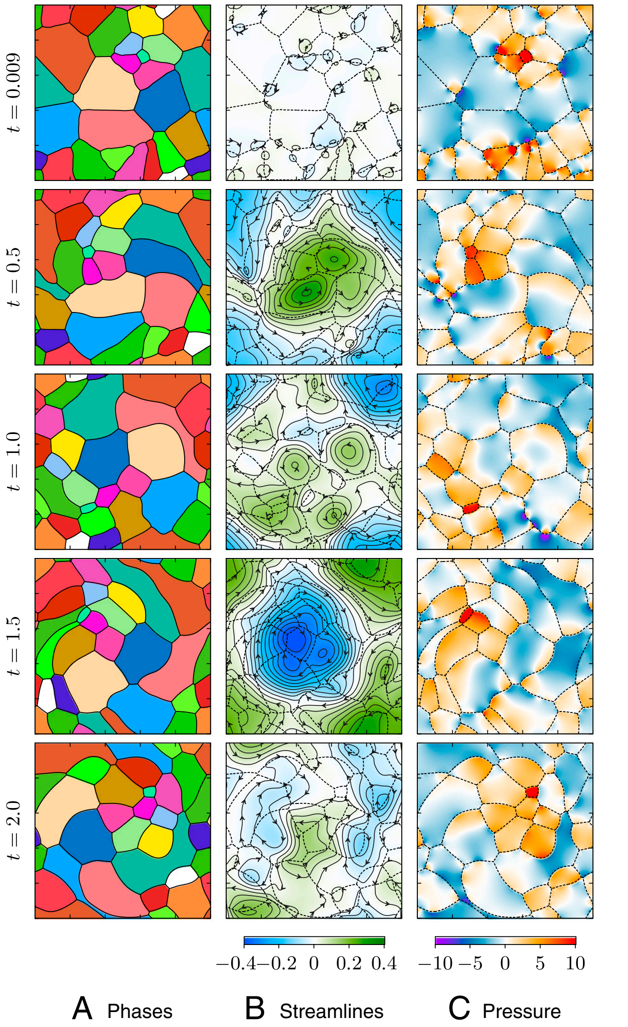


Fig. 7. Results of a fluid flow simulation with an external agitator force and no permeability. Computed on a 256×256 grid with periodic boundary conditions.

A careful analysis and detailed computation of the equations of motion were first performed in ref. 18, in which an explicit 2D Lagrangian-based front tracking method together with an immersed boundary method was used to couple gas fluid mechanics to membrane dynamics under the effects of large shear forces, showing a variety of phenomena, including the balance between evolution toward structures satisfying Young's law and the evolving fluid mechanics. Here, the VIIM allows topological changes in an implicit fashion, and can track 3D flow. Numerical details will be presented elsewhere. Briefly, we use a fixed Cartesian mesh and a second-order projection method, using upwinding for the advection term and Crank–Nicholson for the diffusion term, with the interface surface tension term mollified to the background grid.

Two-Dimensional Flow With and Without Permeability. Consider a foam being agitated by a strong external forcing which first spins counterclockwise, then settles, and then reverses. We take a unit

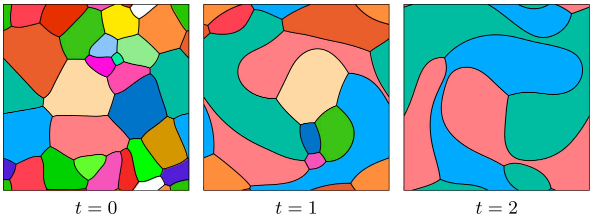
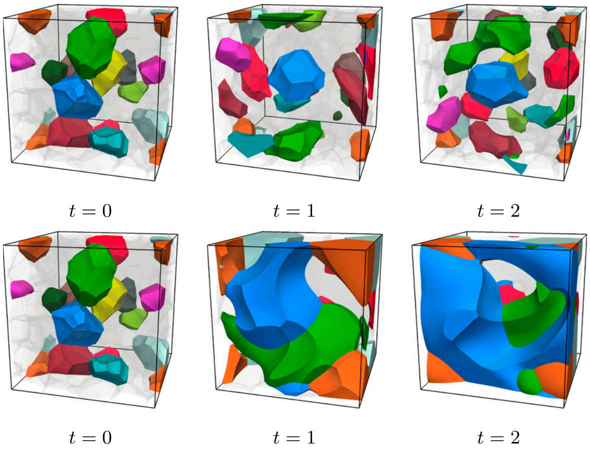

$t = 0$ $t = 1$ $t = 2$

Fig. 8. Two-dimensional multiphase agitator motion with permeability.

square $[0,1]^2$ with periodic boundary conditions, with $\rho = 1$, $\sigma = 1$, and $\mu = 0.005$, chosen so that effects of inertia, viscosity, and surface tension are of similar importance. The agitator is an external force $\mathbf{F}$ in the Navier–Stokes momentum equations: $\mathbf{F}(x,y,t) = 15(\sin \pi x \sin 2\pi y, -\sin 2\pi x \sin \pi y) \sin \pi t$, corresponding to a force which first spins in the counterclockwise direction and then reverses. The factor of 15 in $\mathbf{F}$ was chosen to give a relatively strong shearing/spinning force that dominates the stabilization effects of surface tension. Starting with 25 random phases, we evolve the Navier–Stokes equations with the agitator forcing. First, consider $M = 0$, i.e., no permeability (Fig. 7). Flow undergoes significant shearing, causing considerable rearrangement of the phases and topological change. Stream function plots computed from the velocity field $\mathbf{u}$ show flow strongly affected by agitator forcing, but localized in nature due to effects of surface tension.

Fig. 8 shows the effect of permeability ($M = 0.05$). Diffusion of gas across the membranes cause phases to collapse, others to grow, all of which occurs through large shearing forces.

Three-Dimensional Flow With and Without Permeability. We repeat the above in 3D, further demonstrating the capabilities of VIIM. The domain is a unit cube $[0,1]^3$ and the external force is the same but with no forcing in the z-direction. We start with 125 random phases and evolve the Navier–Stokes equations with the agitator forcing. Once again, the flow undergoes significant shearing, causing considerable rearrangement of the phases and many topological changes. In the case of no permeability (Fig. 9, *Top*), all phases conserve their volume, whereas in the case of permeability (Fig. 9, *Bottom*), diffusion causes some phases to collapse and others to grow under the large shear. In 3D, the rate of volume change matches with a recently found generalization of von Neumann's law (19, 20).


$t = 0$ $t = 1$ $t = 2$

$t = 0$ $t = 1$ $t = 2$

Fig. 9. Three-dimensional multiphase agitator motion, without permeability (*Top*) and with permeability (*Bottom*).

Conclusion

The Voronoi Implicit Interface Method is a robust, accurate, and efficient numerical method to track a large number of evolving interfaces moving under coupled complex interactions of geometry, physics, constraints, and internal boundary conditions. The accuracy of the method has been demonstrated under both convergence studies and verification of von Neumann–Mullins' law, and applications have been presented for fluid flow in the presence of permeability and large shearing forces.

ACKNOWLEDGMENTS. This research was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, US Department of Energy, under contract number DE-AC02-05CH11231, by the Division of Mathematical Sciences of the National Science Foundation, and by National Cancer Institute U54CA143833. J.A.S. was also supported by the Miller Foundation at University of California, Berkeley, and as an Einstein Visiting Fellow of the Einstein Foundation, Berlin. R.I.S. was also supported by an American Australian Association Sir Keith Murdoch Fellowship.

1. Sethian JA (1999) *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences* (Cambridge University Press, Cambridge, U.K.).
2. Merriman B, Bence JK, Osher SJ (1994) Motion of multiple junctions: A level set approach. *J Comput Phys* 112:334–363.
3. Smith KA, Solis FJ, Chopp DL (2002) A projection method for motion of triple junctions by level sets. *Interface Free Bound* 4:263–276.
4. Elsey M, Esedoglu S, Smereka P (2009) Diffusion generated motion for grain growth in two and three dimensions. *J Comput Phys* 228:8015–8033.
5. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J Comput Phys* 79:12–49.
6. Sethian JA (1982) *An analysis of flame propagation* (University of California, Berkeley).
7. Sethian JA (1985) Curvature and the evolution of fronts. *Commun Math Phys* 101:487–499.
8. Sethian JA (1987) Numerical methods for propagating fronts. *Variational Methods for Free Surface Interfaces, Proceedings of the September, 1985 Vallambrosa Conference*, eds P Concus and R Finn (Springer, New York), pp 155–164.
9. Adalsteinsson D, Sethian JA (1995) A fast level set method for propagating interfaces. *J Comput Phys* 118:269–277.
10. Adalsteinsson D, Sethian JA (1999) The fast construction of extension velocities in level set methods. *J Comput Phys* 148:2–22.
11. Chopp DL (2001) Some improvements of the fast marching method. *SIAM J Sci Comput* 23:230–244.
12. Sethian JA (1996) A fast marching level set method for monotonically advancing fronts. *Proc Natl Acad Sci USA* 93:1591–1595.
13. Tsitsiklis JN (1995) Efficient algorithms for globally optimal trajectories. *IEEE Trans Autom Control* 40:1528–1538.
14. Sethian JA, Vladimirsky AV (2003) Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms. *SIAM J Numer Anal* 41:325–363.
15. Weaire D, Hutzler S (1999) *The Physics of Foams* (Oxford University Press, Oxford).
16. von Neumann J (1952) *Metal Interfaces* (American Society for Metals, Cleveland), pp 108–110.
17. Mullins WW (1956) Two-dimensional motion of idealized grain boundaries. *J Appl Phys* 27:900–904.
18. Ming-Chih L, Peskin CS, Yongsam K (2010) Numerical simulations of two-dimensional foam by the immersed boundary method. *J Comput Phys* 229:5194–5207.
19. Hilgenfeldt S, Kraynik AM, Koehler SA, Stone HA (2001) An accurate von Neumann's law for three-dimensional foams. *Phys Rev Lett* 86:2685–2688.
20. MacPherson RD, Srolovitz DJ (2007) The von Neumann relation generalized to coarsening of three-dimensional microstructures. *Nature* 446:1053–1055.