Propagating interfaces and level set methods

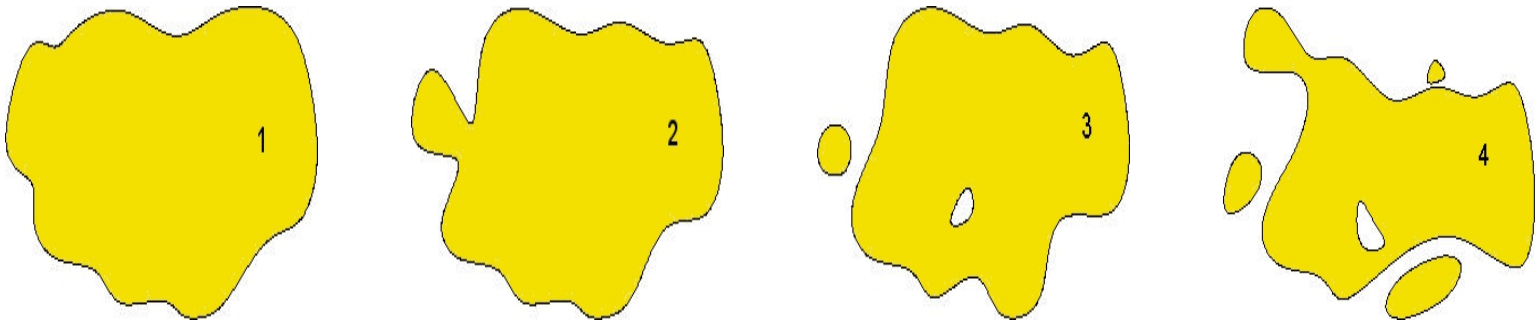
A. Laurain[1]

[1]Humboldt University of Berlin, Germany.

"What is ...?" seminar

Propagating interfaces

- In many problems one needs to model moving geometries, or interfaces. How do we numerically model them?



- Parameterization and functional representation are the simplest ways to model numerically a continuous curve, for instance by using splines. Parameterization may only handle very limited transformations of domains.
- Level Set Methods are much more flexible numerical techniques to model the evolution of interfaces. The interfaces are allowed to develop sharp corners, break apart, and merge together.

Propagating interfaces

Problems with moving interfaces

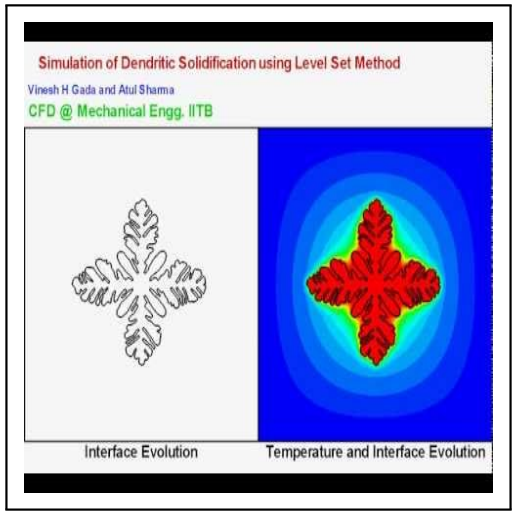
- fluid mechanics
- combustion
- computer animation
- soap bubbles

Looking for an optimal geometry

- inverse problems
- optimization of structure
- image processing
- free boundary problems
- structure of snowflakes

Application: Dendritic Solidification

- Unstable growth of a solidification front.
- The front has a fractal structure.



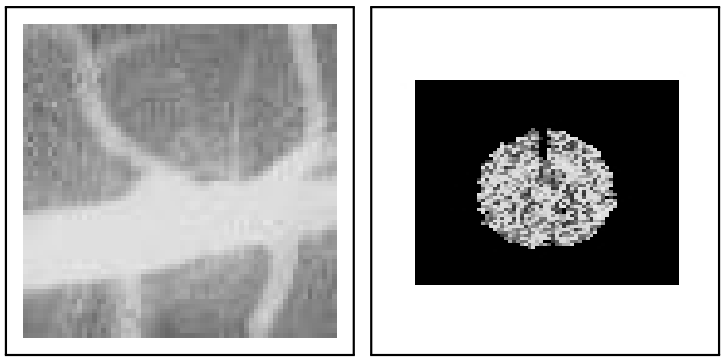
Application: Motion of fluids

- Consider a bubble of a fluid of one density, initially circular, and rising in a fluid of a heavier density.
- As it rises it accelerates in the middle, and the sides are caught up in a pair of swirling vortices.

Application: Segmentation in Medical Imaging

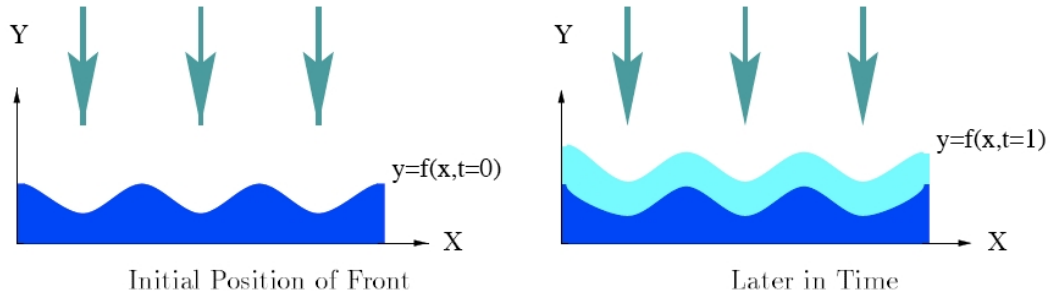
Segmentation consists in extracting the important feature in an image

- When the curve passes over places where the image gradient (that is, the change in value from one pixel to the next) is small, we let the curve expand quickly.
- When the curve passes over places where the image gradient is large, we suspect we are near the boundary, and slow the curve down.
- In addition, we include a curvature term to the speed to add a little surface tension to the expanding contour.

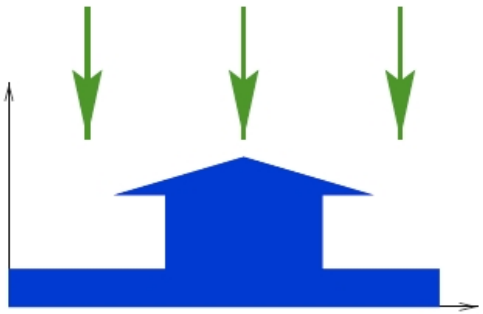


Functional representation

- Imagine snow falling on a hilly terrain $y = f(x, t = 0)$. As the snow accumulates, the height changes in time above each point.

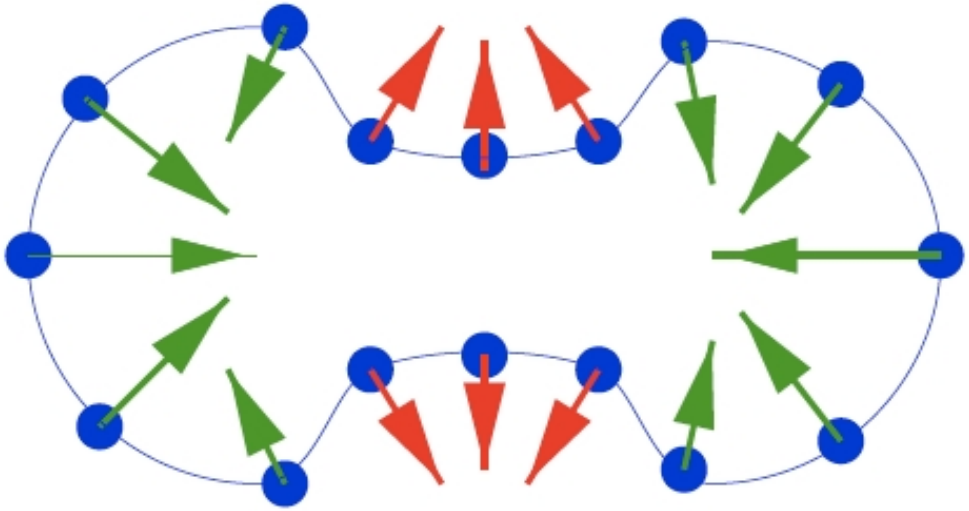


- The initial position of the front can not always be written as the graph of a function.
- Abandon the functional representation for a parameterization.



Parametric representation

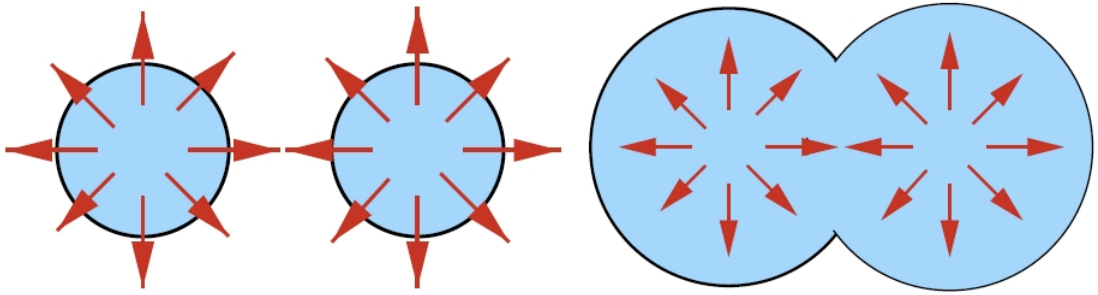
- Use a parameterization of the curve. Plant a blue buoy at regular intervals. A motion by curvature would look like this.



- The strategy is to advance the positions of the buoys according to the arrows, recalculate new arrows and then advance the buoys again.
- Several flaws in this approach!

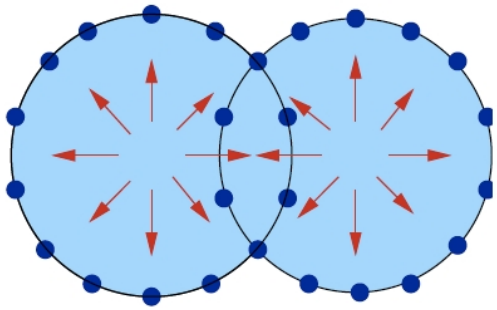
Parametric representation

- The buoys may cross over themselves. A remedy is to reinitialize the placement of the buoys periodically.



Initial Flame Later in Time

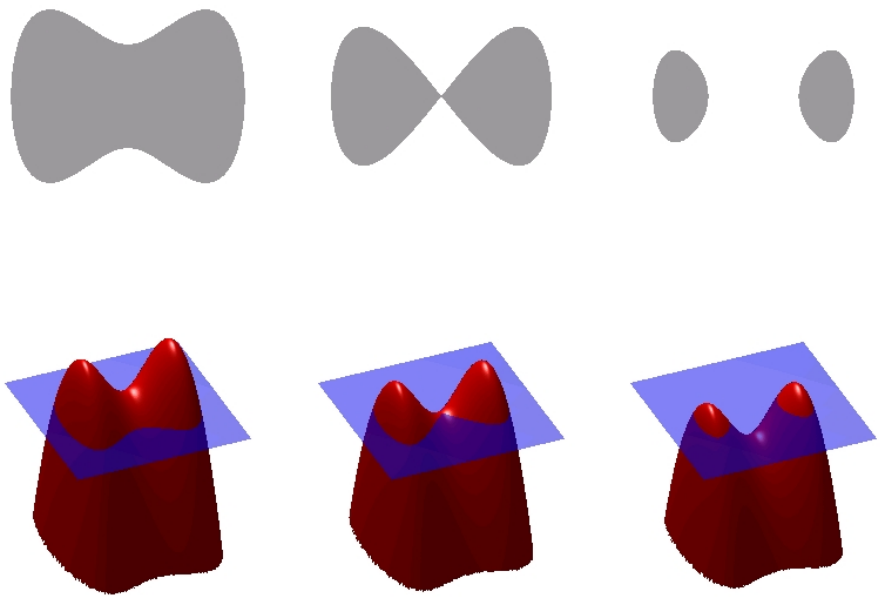
- When the topology changes, the algorithm runs into trouble. Determine which buoy to remove is a confusing task and overwhelming in 3D.



Only "Edge" Bouys Correspond to Propagating Interface

Level set representation

- Functional and parametric representations allow to model only limited situations and have several flaws.
- The idea of a level set representation is to use a coordinate system in one higher dimension.
- We introduce a height $z = \phi(t, x, y)$ where ϕ is the level set function. The curve is the set of points $\{z = 0\}$.



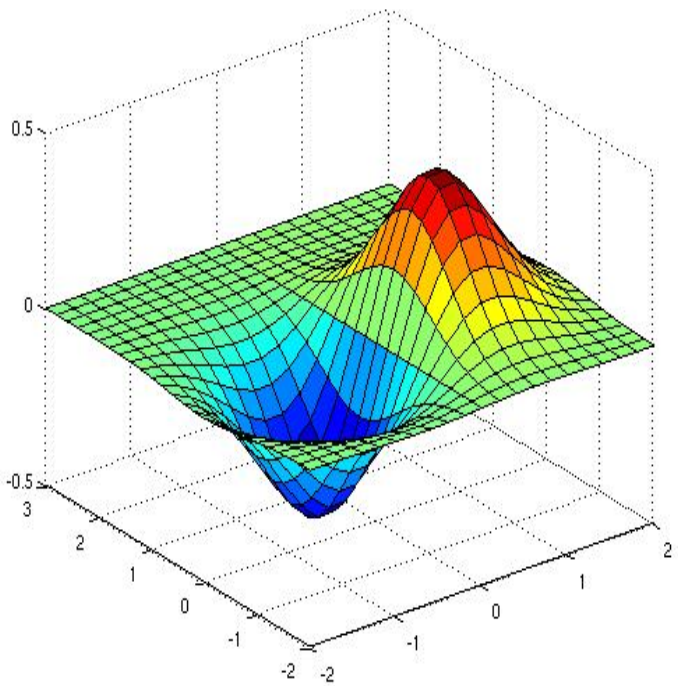
Why a level set method?

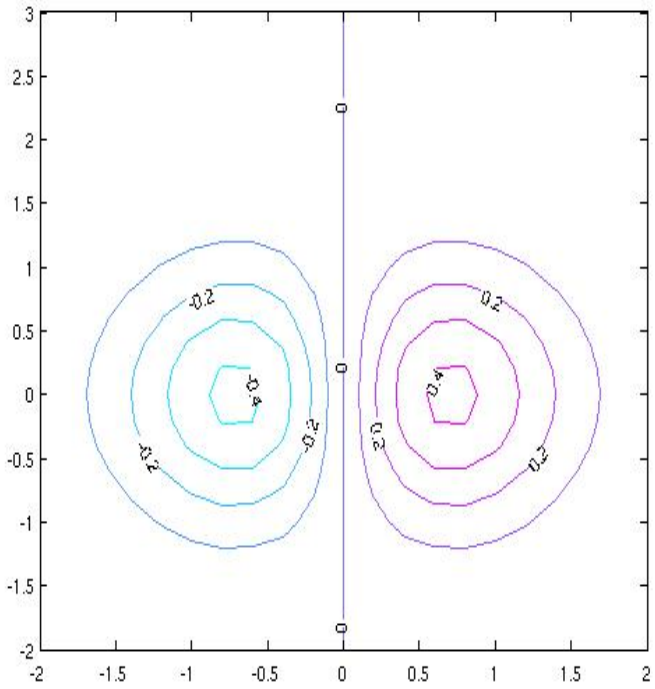
- We trade a moving curve for a moving surface - it seems to be a bad idea in terms of complexity but ...
- The curve may be wild, whereas the surface is always ”well-behaved” (it is always a function).
- Topological changes (breaking and merging) are easy to handle.
- We can considerably reduce the cost of level set methods using narrow bands .
- The method can be directly extended to 3D problems.
- Building accurate numerical schemes to approximate the equations of motion is easy.
- We are using a functional representation in one higher dimension.

What is a level set?

Consider the function $\phi(x, y) = xe^{-(x^2+y^2)}$. The c-level set of ϕ is

$$L(c) = \{(x, y) \in \Omega = [-2, 2]^2 \mid \phi(x, y) = c\}$$





The zero level set is $L(0) = \{(x, y) \in \Omega \mid x = 0\}$.

What is a level set?

Now take a time-dependent function:

$$\phi(t, x, y) = e^{-(x^2+y^2)} - t$$

The 0-level set of $\phi(t, \cdot)$ is also time-dependent

$$\begin{aligned} L(t, 0) &= \{(x, y) \in \Omega \mid \phi(t, x, y) = 0\} \\ &= \{(x, y) \in \Omega \mid e^{-(x^2+y^2)} = t\} \\ &= \{(x, y) \in \Omega \mid x^2 + y^2 = -\log(t)\} \end{aligned}$$

We observe that $L(t, 0)$ is the equation of a curve, precisely a disk of radius $\sqrt{-\log(t)}$ for $0 < t < 1$, and the empty set for $t > 1$.

Modelling the curve evolution

Knowing $\phi(t, x, y)$, we can find its 0-level set (explicitely or an approximation) and reconstruct the moving curve $L(t, 0)$.

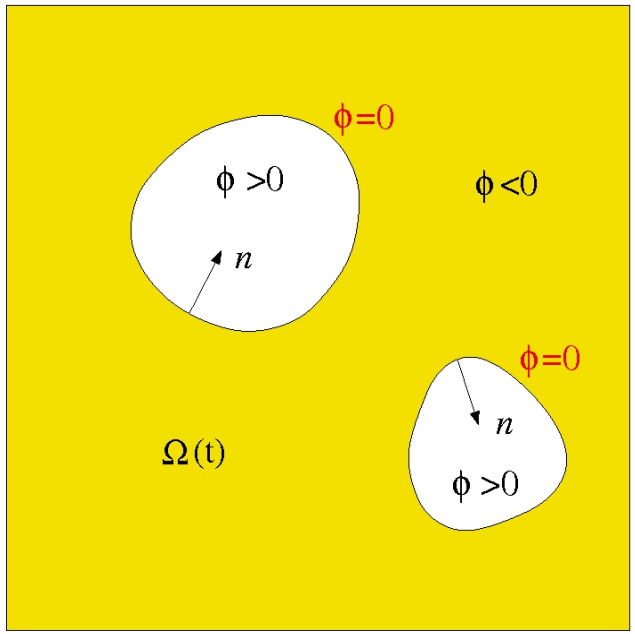
Level set method: definition

Let $D \subset \mathbb{R}^N$ be a bounded domain. Take a level set function

$$\phi : \begin{array}{ccc} \mathbb{R}^+ \times D & \to & \mathbb{R} \\ (t,x) & \mapsto & \phi(t,x) \end{array}$$

and define the sets

$$\begin{array}{rcl} \Omega_t & := & \{x \in D \mid \phi(t,x) < 0\} \\ \Omega_t^c & := & \{x \in D \mid \phi(t,x) > 0\} \\ \partial\Omega_t & := & \{x \in D \mid \phi(t,x) = 0\} \end{array}$$



Level set method: initialization

$\phi(0, x)$ can be initialized as the signed distance function to $\partial\Omega_0$

$$\phi(0, x) = \text{dist}(x, \partial\Omega_0) \text{ if } x \in \Omega^c$$
$$\phi(0, x) = -\text{dist}(x, \partial\Omega_0) \text{ if } x \in \Omega$$

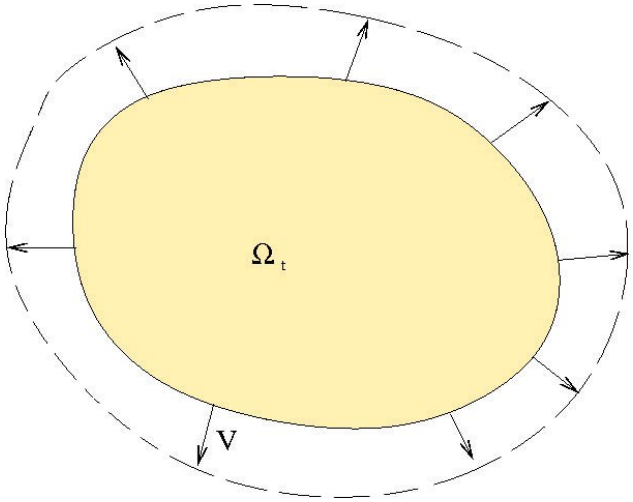
The derivatives of the signed distance function provide useful geometric characteristics of the curve $\partial\Omega_t$

- $|\nabla_x \phi| = 1$ where ϕ is differentiable
- $n(t, x) = \nabla_x \phi(t, x)$ is the normal vector on $\partial\Omega_t$.
- $\kappa(t, x) = \Delta_x \phi(t, x)$ is the curvature on $\partial\Omega_t$.

Evolution of the level set function

- If the level set function $\phi(t, x, y)$ is given, we can reconstruct the corresponding domain Ω_t.
- Usually, the only data at our disposal is an initial domain Ω_0 and a vector field (a "speed") $V(t)$ which may itself depend on Ω_t.
- How can we find the corresponding level set function $\phi(t, x, y)$ without knowing Ω_t?
- We need to solve a partial differential equation (PDE) for $\phi(t, x, y)$.

Evolution of the level set function



moving domain Ω_t

X: Lagrangian coordinate
$x(t, X)$: Eulerian coordinate

$$\begin{aligned} \frac{d}{dt}x(t,X) &= V(t, x(t,X)) \\ x(0,X) &= X \end{aligned}$$

$T_t(V)(X) = x(t, X)$
$\Omega_t = T_t(V)(\Omega)$: moving domain

- Consider a point $x(t)$ on the moving boundary $\Gamma_t := \partial\Omega_t$.
- For this point we have $\phi(t, x(t)) = 0$. Differentiating w.r.t. t we get

$$\partial_t\phi(t,x) + V(t,x)\cdot\nabla\phi(t,x) = 0.$$

Hamilton-Jacobi equation

Since $\nabla\phi(t,x) = |\nabla\phi(t,x)|n(t,x)$ we get with $v_n = \langle V, n\rangle$

$$\partial_t\phi(t,x) + v_n(t,x)|\nabla\phi(t,x)| = 0,$$

and $\phi(0,x)$ is the signed distance function to $\partial\Omega_0$.

Examples of perturbation fields v_n

- $v_n \equiv 1$: the domain becomes bigger
- $v_n \equiv -\kappa$: (κ is the mean curvature) the domain becomes smoother and shrinks to a point.
- $\mathrm{div}_x(v_n) \equiv 0$: the volume of the domain Ω_t is constant.
- In general v_n can be any function with enough regularity.

Numerical scheme

- The level set equation should not be too flat or too steep.
- Take ϕ to be a distance function i.e. $|\nabla\phi| = 1$.
- The solution ϕ of the Hamilton-Jacobi equation does not remain close to a distance function in general.
- Reinitialize ϕ at time t by solving

$$\begin{aligned} \partial_\tau \varphi + S(\phi)(|\nabla\varphi| - 1) &= 0 \text{ in } \mathbb{R}^+ \times D, \\ \varphi(0, x) &= \phi(t, x),\ x \in D, \end{aligned}$$

up to the stationary state, with the approximate sign function

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2\varepsilon^2}}$$

with $\varepsilon = \min(\Delta x, \Delta y)$ and Δx and Δy are the space steps discretization.

Extend the normal velocity

- If $v_n = \langle V, n \rangle$ is defined only on Γ_t it is necessary to extend it on the entire domain D.
- At the same time we may enforce ϕ to remain (close to) a distance function.
- Compute an extended normal velocity V_{ext} constant along the normal, i.e. V_{ext} should satisfy

$$\nabla V_{\text{ext}} \cdot \nabla \phi = 0 \text{ in } \mathbb{R}^+ \times D,$$

  it can be shown that ϕ then satisfies $|\nabla \phi| = 1$.
- $V_{\text{ext}}(t, x) = \lim_{\tau \to \infty} q(\tau, x)$ with q solution of

$$\begin{aligned} \partial_\tau q + S(\phi) \frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla q &= 0 \text{ in } \mathbb{R}^+ \times D \\ q(0, x) &= v_n(t, x) \mathbb{1}_{\Gamma_t}, \ x \in D \end{aligned}$$

The fast marching method

- Unlike level set methods, Fast Marching Methods are designed for problems in which the speed function never changes sign.
- This allows us to convert the problem to a stationary formulation, because the front crosses each point on the grid only once.
- Can be seen as a special case of domain evolution, for which very fast algorithms exists.
- This amounts to solving an Eikonal equation for which we can use a Dijkstra-like method.

Thanks for your attention!