

The Level Set Method

MIT 16.920J / 2.097J / 6.339J

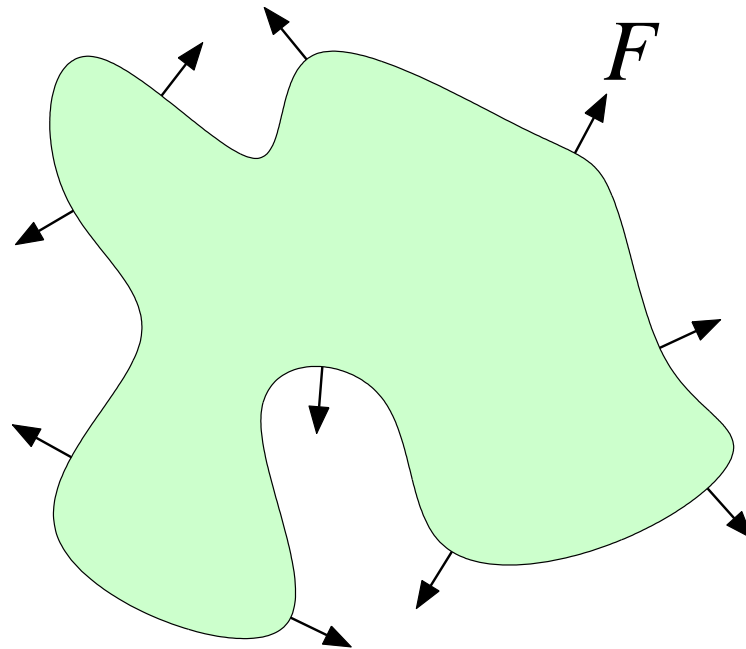
Numerical Methods for Partial Differential Equations

Per-Olof Persson (persson@mit.edu)

March 8, 2005

Evolving Curves and Surfaces

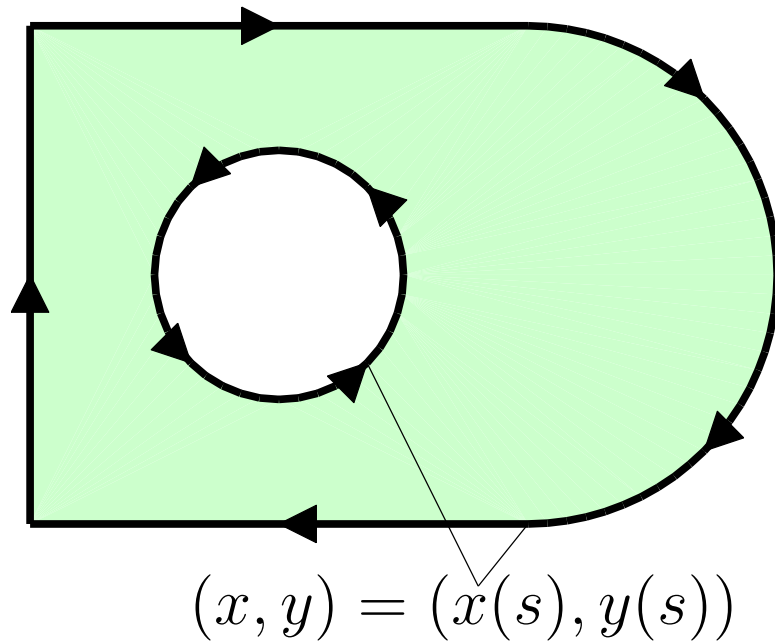
- Propagate curve according to speed function $v = F\mathbf{n}$
- F depends on space, time, and the curve itself
- Surfaces in three dimensions



Geometry Representations

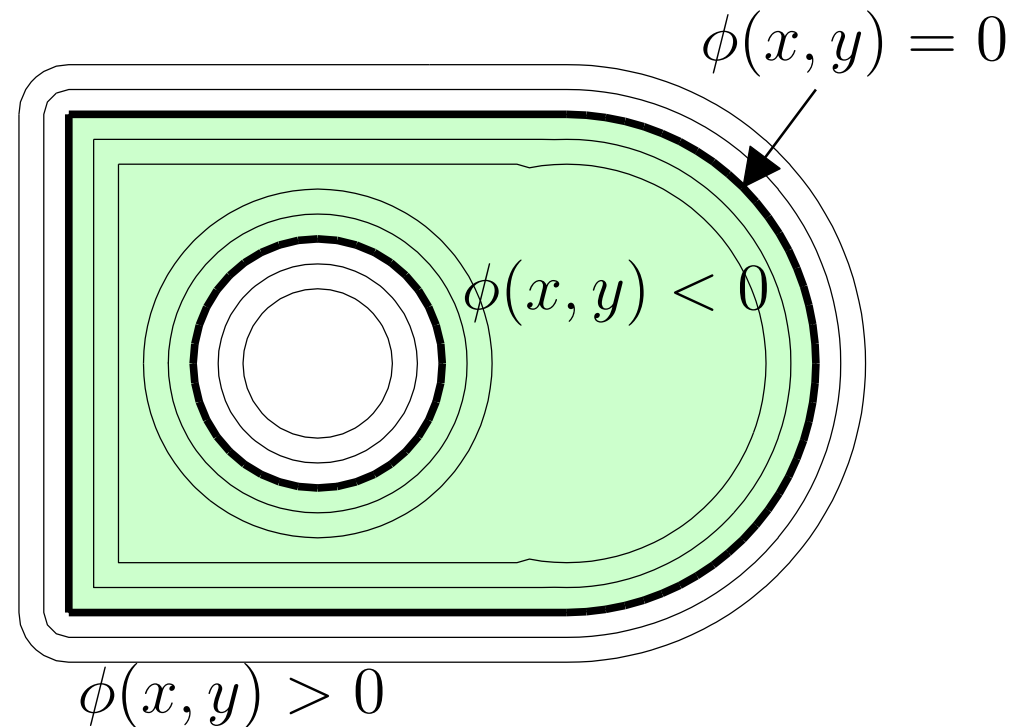
Explicit Geometry

- Parameterized boundaries



Implicit Geometry

- Boundaries given by zero level set



Explicit Techniques

- Simple approach: Represent curve explicitly by nodes $\mathbf{x}^{(i)}$ and lines
- Propagate curve by solving ODEs

$$\frac{d\mathbf{x}^{(i)}}{dt} = \mathbf{v}(\mathbf{x}^{(i)}, t), \quad \mathbf{x}^{(i)}(0) = \mathbf{x}_0^{(i)},$$

- Normal vector, curvature, etc by difference approximations, e.g.:

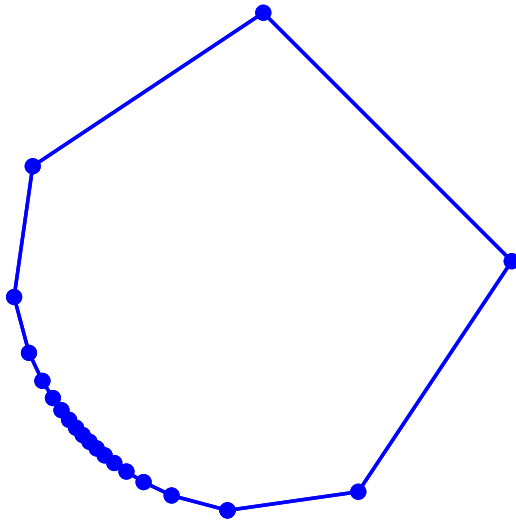
$$\frac{d\mathbf{x}^{(i)}}{ds} \approx \frac{\mathbf{x}^{(i+1)} - \mathbf{x}^{(i-1)}}{2\Delta s}$$

- MATLAB Demo

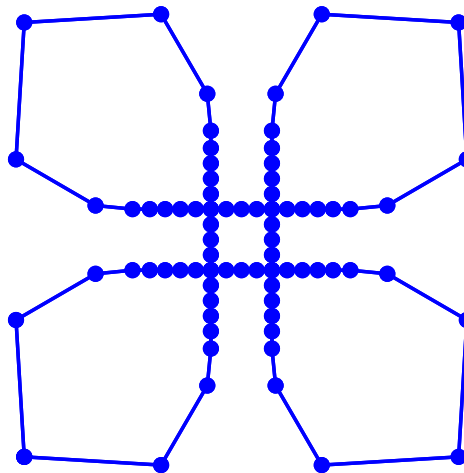
Explicit Techniques - Drawbacks

- Node redistribution required, introduces errors
- No entropy solution, sharp corners handled incorrectly
- Need special treatment for topology changes
- Stability constraints for curvature dependent speed functions

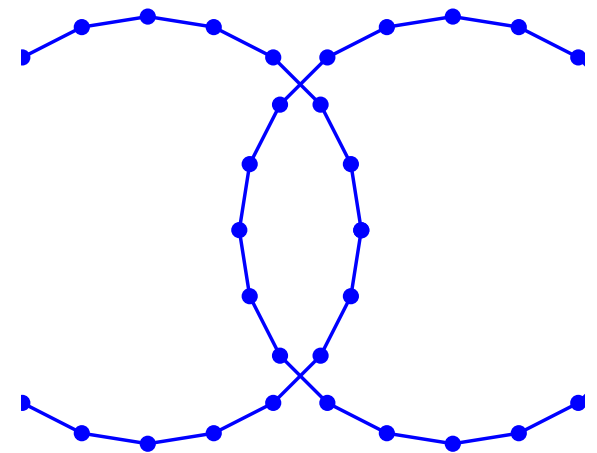
Node distribution



Sharp corners



Topology changes

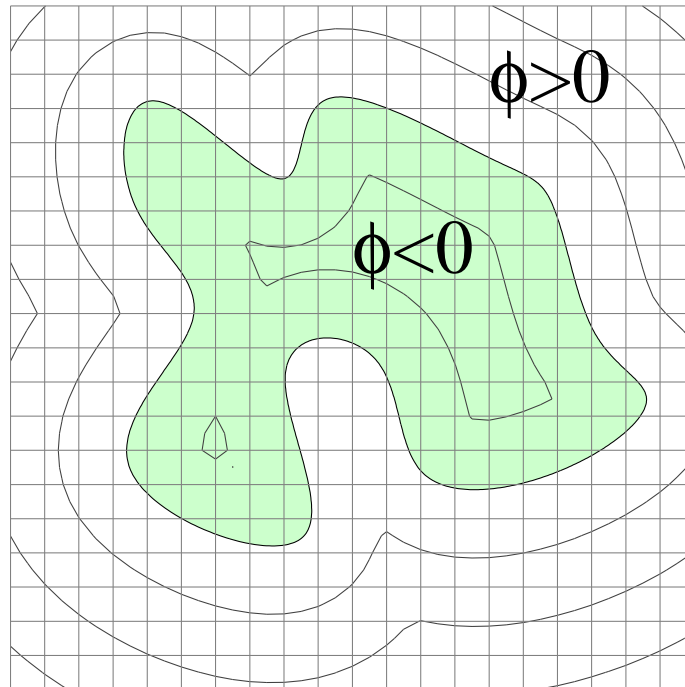


The Level Set Method

- Implicit geometries, evolve interface by solving PDEs
- Invented in 1988 by Osher and Sethian:
 - Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- Two good introductory books:
 - James A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, Cambridge, second edition, 1999.
 - Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer-Verlag, New York, 2003.

Implicit Geometries

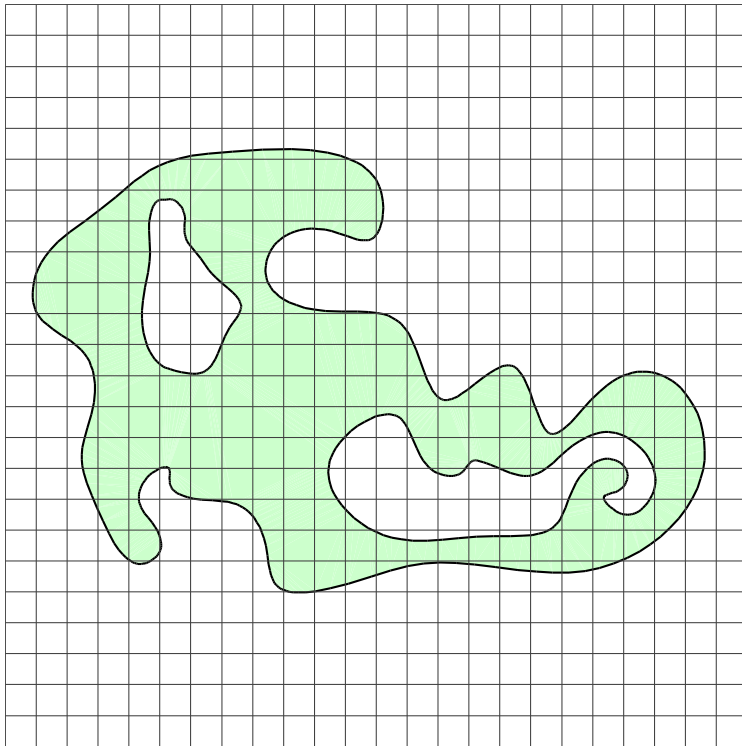
- Represent curve by zero level set of a function, $\phi(\mathbf{x}) = 0$
- Special case: *Signed distance function*:
 - $|\nabla \phi| = 1$
 - $|\phi(\mathbf{x})|$ gives (shortest) distance from \mathbf{x} to curve



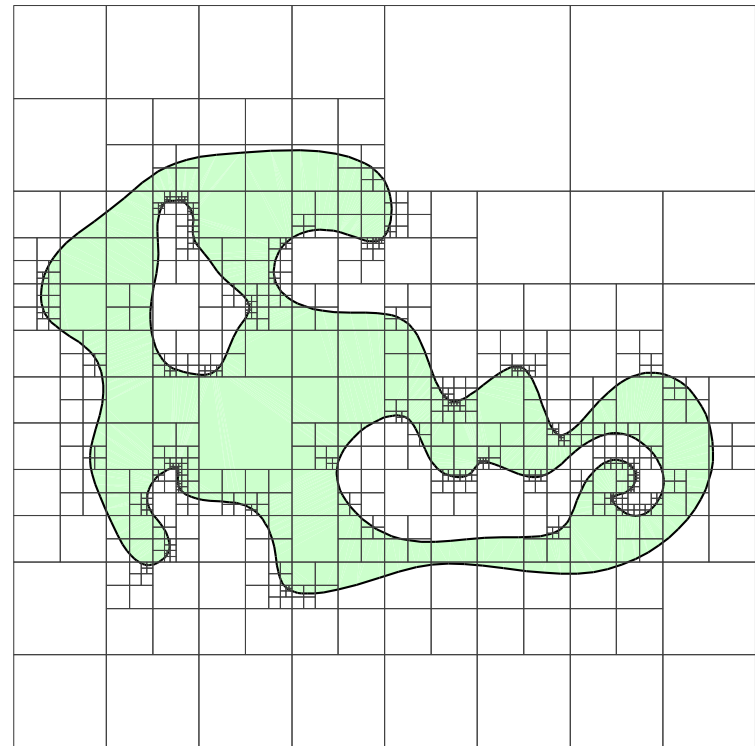
Discretized Implicit Geometries

- Discretize implicit function ϕ on *background grid*
- Obtain $\phi(\mathbf{x})$ for general \mathbf{x} by interpolation

Cartesian



Quadtree/Octree



Geometric Variables

- Normal vector \mathbf{n} (without assuming distance function):

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$$

- Curvature (in two dimensions):

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_y\phi_x\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}.$$

- Write material parameters, etc, in terms of ϕ :

$$\rho(\mathbf{x}) = \rho_1 + (\rho_2 - \rho_1)\theta(\phi(\mathbf{x}))$$

Smooth Heaviside function θ over a few grid cells.

The Level Set Equation

- Solve convection equation to propagate $\phi = 0$ by velocities \mathbf{v}

$$\phi_t + \mathbf{v} \cdot \nabla \phi = 0.$$

- For $\mathbf{v} = F\mathbf{n}$, use $\mathbf{n} = \nabla \phi / |\nabla \phi|$ and $\nabla \phi \cdot \nabla \phi = |\nabla \phi|^2$ to obtain the *Level Set Equation*

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

- Nonlinear, hyperbolic equation (Hamilton-Jacobi).

Discretization

- Use upwinded finite difference approximations for convection
- For the level set equation $\phi_t + F|\nabla\phi| = 0$:

$$\phi_{ijk}^{n+1} = \phi_{ijk}^n + \Delta t_1 \left(\max(F, 0) \nabla_{ijk}^+ + \min(F, 0) \nabla_{ijk}^- \right),$$

where

$$\begin{aligned} \nabla_{ijk}^+ = & \left[\max(D^{-x}\phi_{ijk}^n, 0)^2 + \min(D^{+x}\phi_{ijk}^n, 0)^2 + \right. \\ & \max(D^{-y}\phi_{ijk}^n, 0)^2 + \min(D^{+y}\phi_{ijk}^n, 0)^2 + \\ & \left. \max(D^{-z}\phi_{ijk}^n, 0)^2 + \min(D^{+z}\phi_{ijk}^n, 0)^2 \right]^{1/2}, \end{aligned}$$

Discretization

and

$$\nabla_{ijk}^- = \left[\min(D^{-x}\phi_{ijk}^n, 0)^2 + \max(D^{+x}\phi_{ijk}^n, 0)^2 + \right. \\ \min(D^{-y}\phi_{ijk}^n, 0)^2 + \max(D^{+y}\phi_{ijk}^n, 0)^2 + \\ \left. \min(D^{-z}\phi_{ijk}^n, 0)^2 + \max(D^{+z}\phi_{ijk}^n, 0)^2 \right]^{1/2}.$$

- D^{-x} backward difference operator in the x -direction, etc
- For curvature dependent part of F , use central differences
- Higher order schemes available
- MATLAB Demo

Reinitialization

- Large variations in $\nabla\phi$ for general speed functions F
- Poor accuracy and performance, need smaller timesteps for stability
- *Reinitialize* by finding new ϕ with same zero level set but $|\nabla\phi| = 1$
- Different approaches:

1. Integrate the *reinitialization equation* for a few time steps

$$\phi_t + \text{sign}(\phi)(|\nabla\phi| - 1) = 0$$

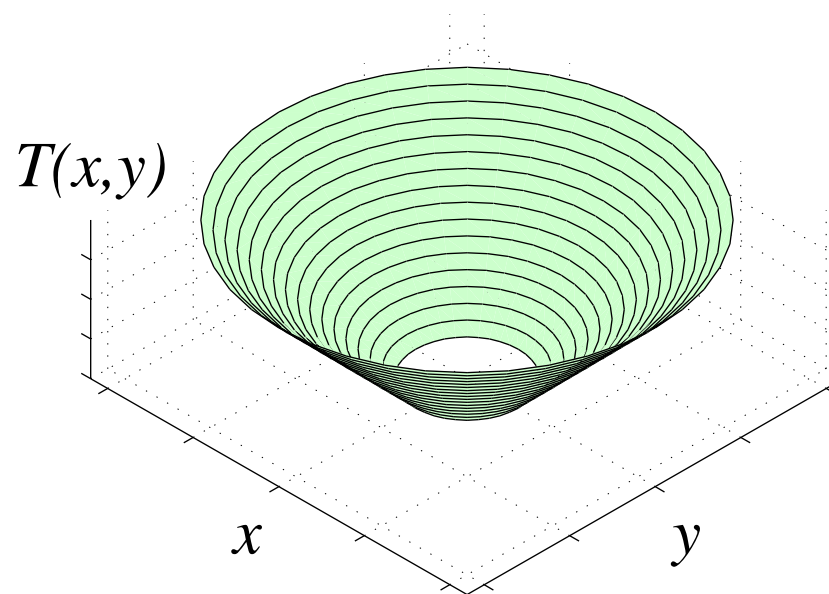
2. Compute distances from $\phi = 0$ explicitly for nodes close to boundary, use Fast Marching Method for remaining nodes

The Boundary Value Formulation

- For $F > 0$, formulate evolution by an arrival function T
- $T(x)$ gives time to reach x from initial Γ
- time * rate = distance gives the *Eikonal equation*:

$$|\nabla T|F = 1, \quad T = 0 \text{ on } \Gamma.$$

- Special case: $F = 1$ gives distance functions



The Fast Marching Method

- Discretize the Eikonal equation $|\nabla T|F = 1$ by

$$\left[\begin{aligned} &\max(D_{ijk}^{-x}T, 0)^2 + \min(D_{ijk}^{+x}T, 0)^2 \\ &+ \max(D_{ijk}^{-y}T, 0)^2 + \min(D_{ijk}^{+y}T, 0)^2 \\ &+ \max(D_{ijk}^{-z}T, 0)^2 + \min(D_{ijk}^{+z}T, 0)^2 \end{aligned} \right]^{1/2} = \frac{1}{F_{ijk}}$$

or

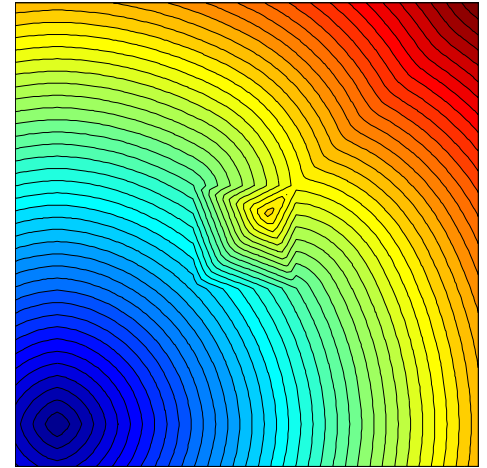
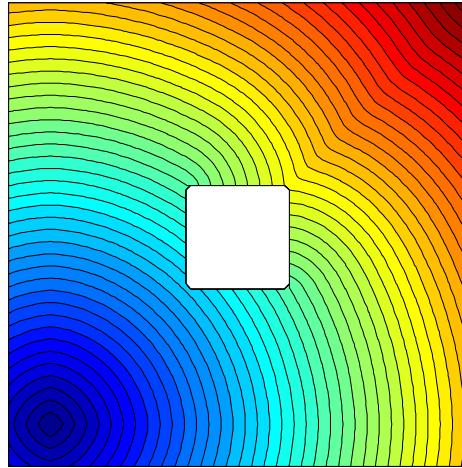
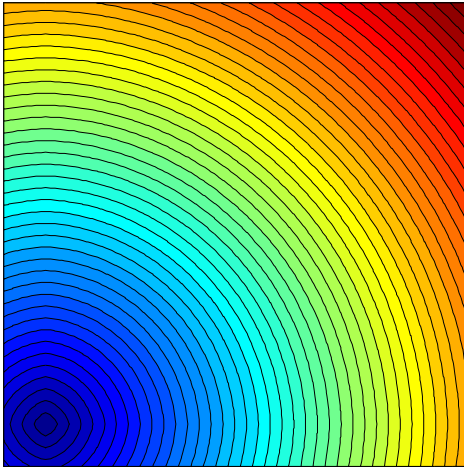
$$\left[\begin{aligned} &\max(D_{ijk}^{-x}T, -D_{ijk}^{+x}T, 0)^2 \\ &+ \max(D_{ijk}^{-y}T, -D_{ijk}^{+y}T, 0)^2 \\ &+ \max(D_{ijk}^{-z}T, -D_{ijk}^{+z}T, 0)^2 \end{aligned} \right]^{1/2} = \frac{1}{F_{ijk}}$$

The Fast Marching Method

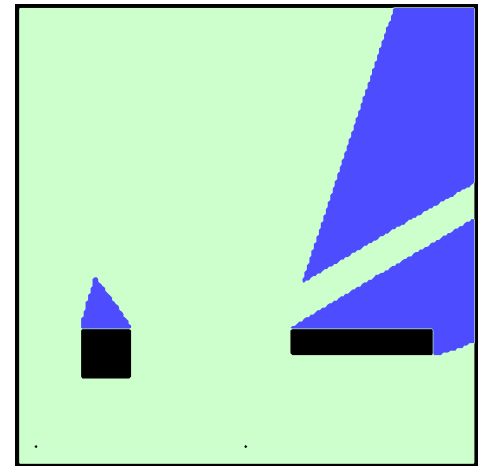
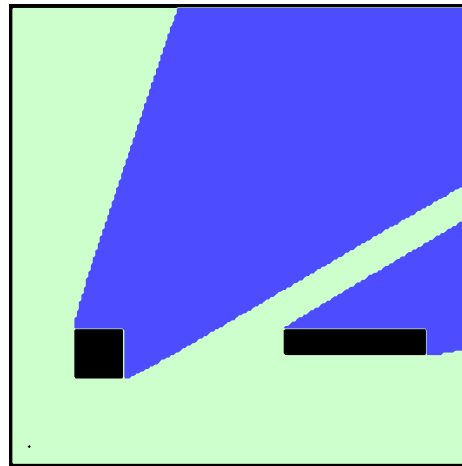
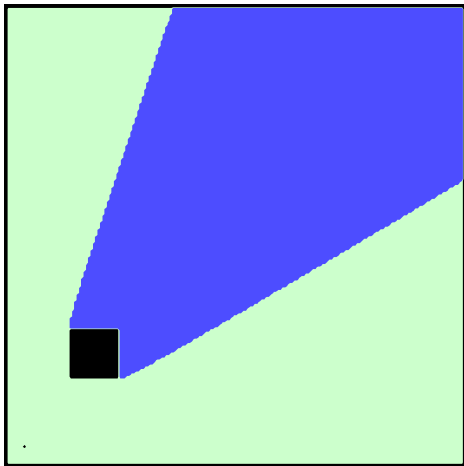
- Use the fact that the front propagates outward
- Tag known values and update neighboring T values (using the difference approximation)
- Pick unknown with smallest T (will not be affected by other unknowns)
- Update new neighbors and repeat until all nodes are known
- Store unknowns in priority queue, $\mathcal{O}(n \log n)$ performance for n nodes with heap implementation

Applications

First arrivals and shortest geodesic paths



Visibility around obstacles



Structural Vibration Control

- Consider eigenvalue problem

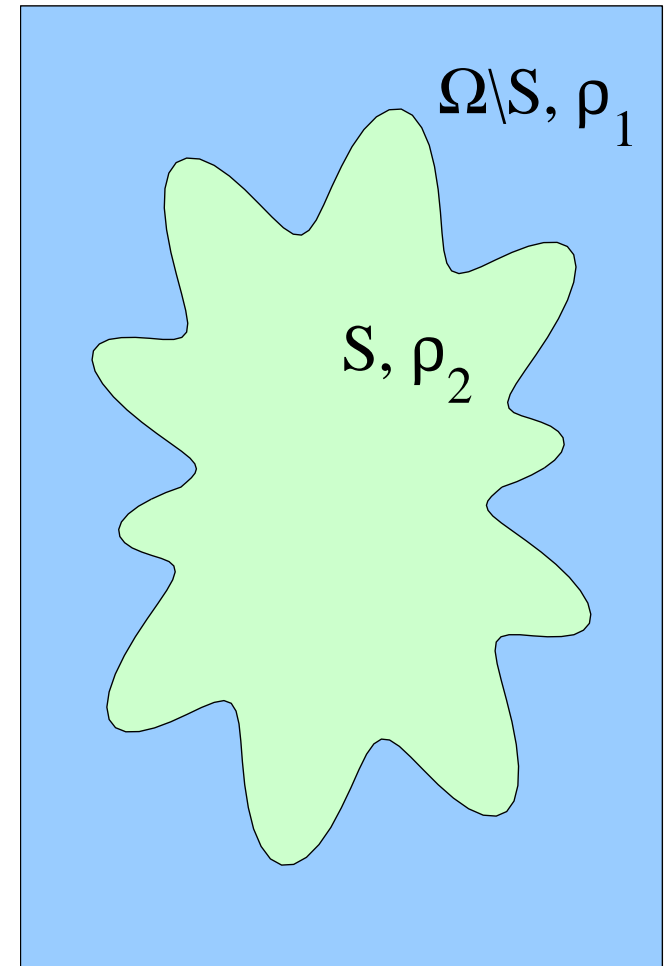
$$\begin{aligned} -\Delta u &= \lambda \rho(\boldsymbol{x}) u, & x &\in \Omega \\ u &= 0, & x &\in \partial\Omega. \end{aligned}$$

with

$$\rho(\boldsymbol{x}) = \begin{cases} \rho_1 & \text{for } x \notin S \\ \rho_2 & \text{for } x \in S. \end{cases}$$

- Solve the optimization

$$\min_S \lambda_1 \text{ or } \lambda_2 \text{ subject to } \|S\| = K.$$



Structural Vibration Control

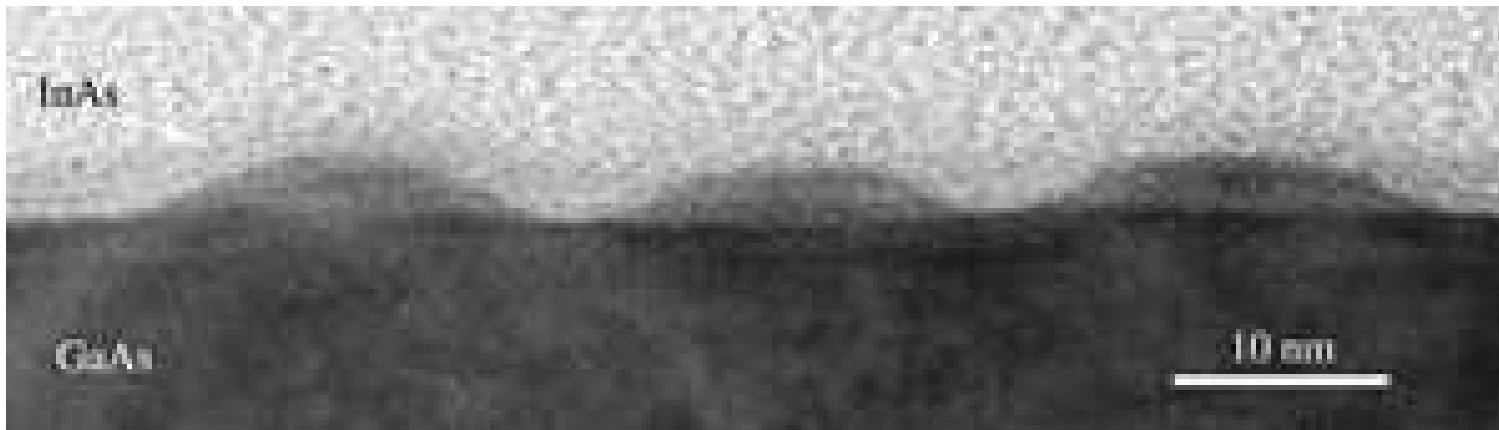
- Level set formulation by Osher and Santosa:
 - Finite difference approximations for Laplacian
 - Sparse eigenvalue solver for solutions λ_i, u_i
 - Calculate descent direction $\delta\phi = -v(\mathbf{x})|\nabla\phi|$ with $v(\mathbf{x})$ from shape sensitivity analysis
 - Find Lagrange multiplier for area constraint using Newton's method
 - Represent interface implicitly, propagate using level set method

Stress Driven Rearrangement Instabilities

- Epitaxial growth of InAs on a GaAs substrate, stress from misfit in lattices
- Quasi-static interface evolution, descent direction for elastic energy and surface energy

$$\frac{\partial \phi}{\partial \tau} + F(\mathbf{x}) |\nabla \phi| = 0, \text{ with } F(\mathbf{x}) = \varepsilon(\mathbf{x}) - \sigma \kappa(\mathbf{x})$$

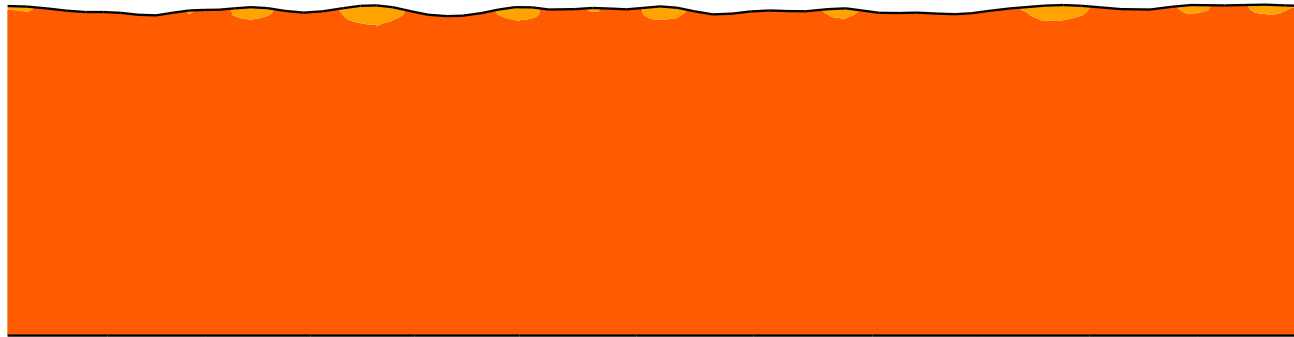
- Level set formulation by Persson, finite elements for the elasticity



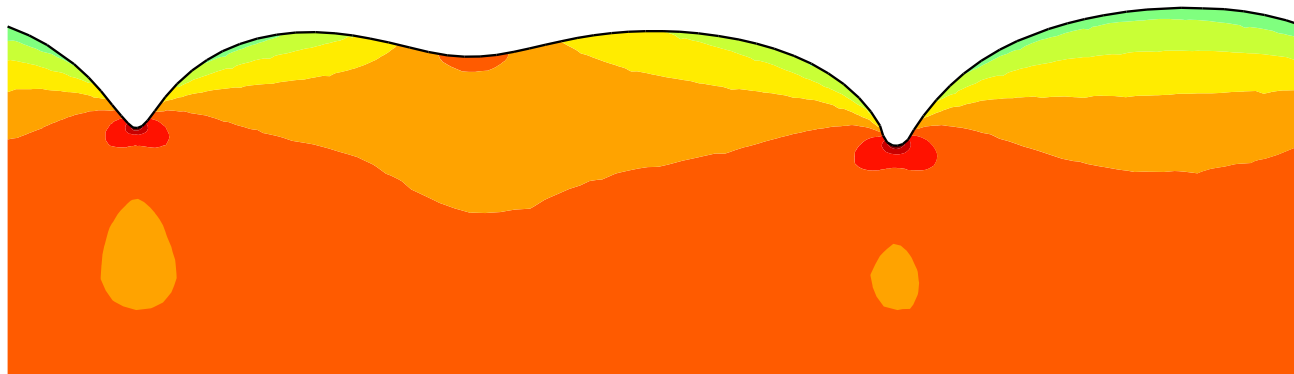
Electron micrograph of defect-free InAs quantum dots

Stress Driven Rearrangement Instabilities

Initial Configuration

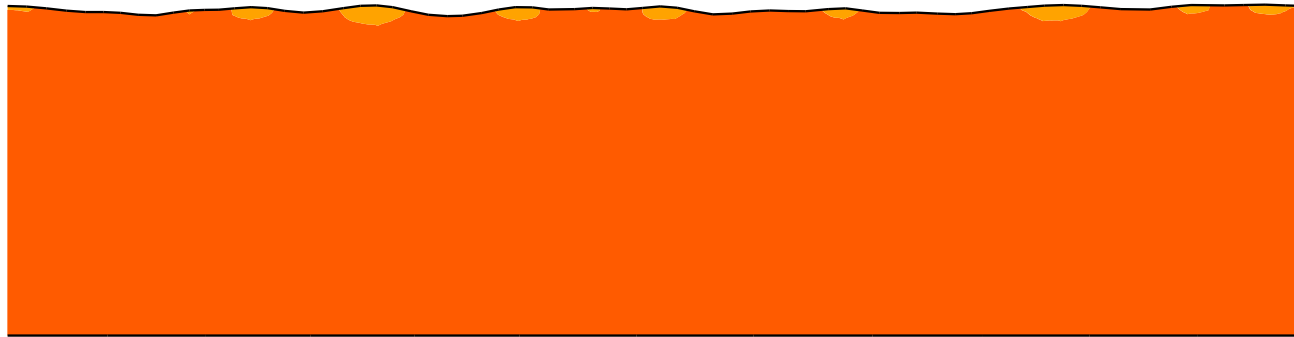


Final Configuration, $\sigma = 0.20$

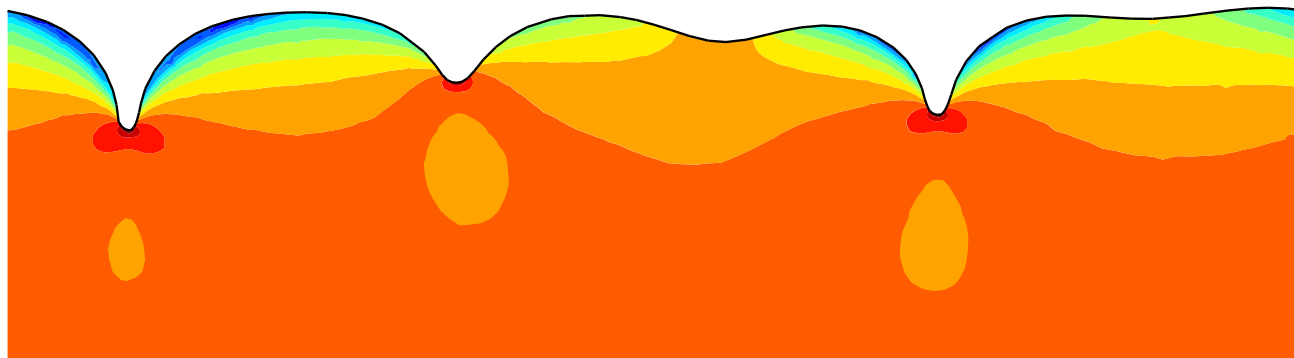


Stress Driven Rearrangement Instabilities

Initial Configuration



Final Configuration, $\sigma = 0.10$



Stress Driven Rearrangement Instabilities

Initial Configuration



Final Configuration, $\sigma = 0.05$

