The Level Set Method

MIT 16.920J / 2.097J / 6.339J
Numerical Methods for Partial Differential Equations

Per-Olof Persson
March 8, 2005
Evolving Curves and Surfaces

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

Explicit Geometry

- Parameterized boundaries

\[(x, y) = (x(s), y(s))\]

Implicit Geometry

- Boundaries given by zero level set

\[\phi(x, y) = 0\]

\[\phi(x, y) > 0\]

\[\phi(x, y) < 0\]
Explicit Techniques

- Simple approach: Represent curve explicitly by nodes \( x^{(i)} \) and lines

- Propagate curve by solving ODEs

\[
\frac{d x^{(i)}}{d t} = v(x^{(i)}, t), \quad x^{(i)}(0) = x_0^{(i)},
\]

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

\[
\frac{d x^{(i)}}{d s} \approx \frac{x^{(i+1)} - x^{(i-1)}}{2\Delta s}
\]

- 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:

• Two good introductory books:
Implicit Geometries

- Represent curve by zero level set of a function, $\phi(x) = 0$

- Special case: *Signed distance function*:
  - $|\nabla \phi| = 1$
  - $|\phi(x)|$ gives (shortest) distance from $x$ to curve
Discretized Implicit Geometries

- Discretize implicit function $\phi$ on background grid
- Obtain $\phi(x)$ for general $x$ by interpolation
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 \( \phi \):

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

Smooth Heaviside function \( \theta \) 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

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

and

\[ \nabla^-_{ijk} = \left[ \min(D^{-x}\phi^n_{ijk}, 0)^2 + \max(D^{+x}\phi^n_{ijk}, 0)^2 + \min(D^{-y}\phi^n_{ijk}, 0)^2 + \max(D^{+y}\phi^n_{ijk}, 0)^2 + \min(D^{-z}\phi^n_{ijk}, 0)^2 + \max(D^{+z}\phi^n_{ijk}, 0)^2 \right]^{1/2}. \]

- \( D^{-x} \) 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 $\phi$ 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 $\Gamma$
- time $\times$ rate $=$ distance gives the \textit{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 $|∇T|F = 1$ by

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

or

$$\left[ \begin{array}{c} \max(D_{ijk}^{-x}T, -D_{ijk}^{+x}T,0)^2 \\ + \max(D_{ijk}^{-y}T, -D_{ijk}^{+y}T,0)^2 \\ + \max(D_{ijk}^{-z}T, -D_{ijk}^{+z}T,0)^2 \end{array} \right]^{1/2} = \frac{1}{F_{ijk}}$$
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, $O(n \log n)$ performance for $n$ nodes with heap implementation
Applications
First arrivals and shortest geodesic paths

Visibility around obstacles
Structural Vibration Control

- Consider eigenvalue problem

\[ -\Delta u = \lambda \rho(x) u, \quad x \in \Omega \]

\[ u = 0, \quad x \in \partial \Omega. \]

with

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

- 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 $\lambda_i, u_i$
  - Calculate descent direction $\delta \phi = -v(x)|\nabla \phi|$ with $v(x)$ 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
  \[ \frac{\partial \phi}{\partial \tau} + F(x)|\nabla \phi| = 0, \text{ with } F(x) = \varepsilon(x) - \sigma \kappa(x) \]
- 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, $\sigma = 0.05$