A New Interface Tracking Method: Using Level Set and Particle Methods

Advisors: Jinsun Sohn, Joseph Teran (UCLA)

Trevor Caldwell[2] Noah Duncan[2] Tengyuan Liang[3] Shirley Zheng[1]

[1]Cornell University

[2]Harvey Mudd College

[3]Peking University

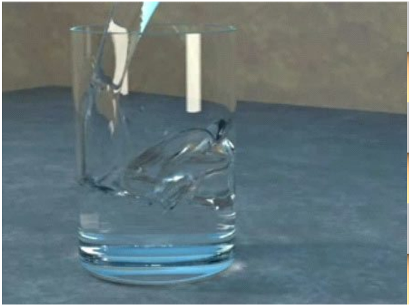
August 3, 2011

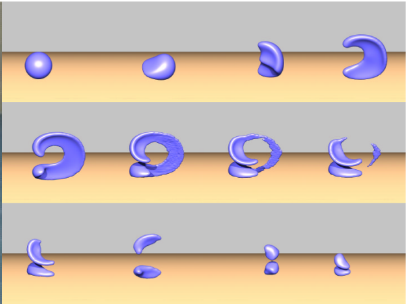
Table of Contents I

Background

Background

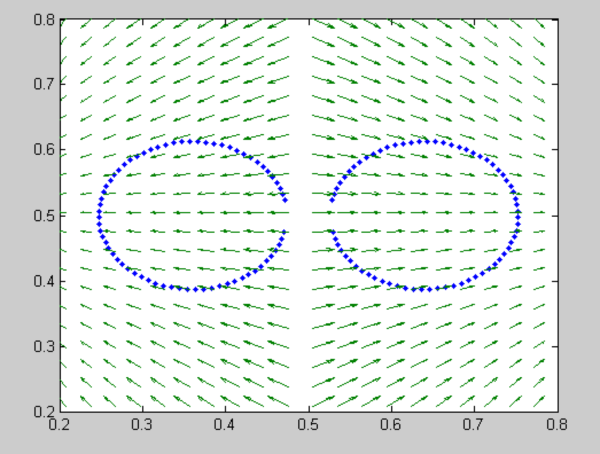
- Typical traditional interface advection is done using level set or particle methods.
- Because of numerical error, objects lose volume when advected by level set methods so that they shrink and eventually disappear as time passes.





Background

- In contrast, particle methods are numerically accurate but do not naturally handle topological change.



Problem Statement

Devise a method that is *numerically accurate* and robust under *topological change*. The interface should be represented implicitly at each iteration, and the particles might be resampled periodically.

Implicit vs Explicit Representation

- We can capture and evolve a hypersurface implicitly using level set methods, or we can track it explicitly using particle methods.

Implicit vs Explicit Representation

- We can capture and evolve a hypersurface implicitly using level set methods, or we can track it explicitly using particle methods.
- In level set methods, we update the implicit function ϕ across the domain at each time step.

Level Set

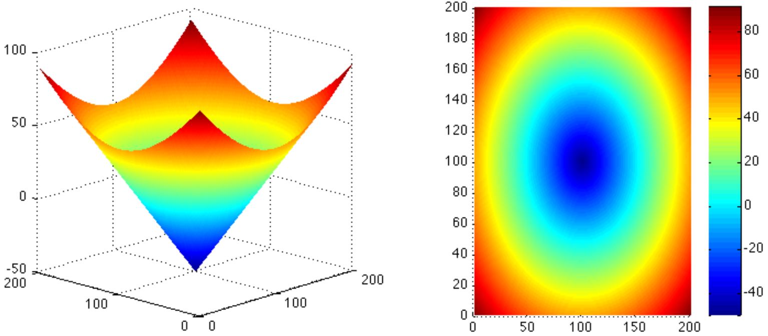


Figure: Signed Distance Representation of Circle in $\mathbb{R}^2$.

Implicit vs Explicit Representation

- *We can capture and evolve a hypersurface implicitly using level set methods, or we can track it explicitly using particle methods.*
- *In level set methods, we update the implicit function ϕ across the domain at each time step.*

Implicit vs Explicit Representation

- *We can capture and evolve a hypersurface implicitly using level set methods, or we can track it explicitly using particle methods.*
- *In level set methods, we update the implicit function* ϕ *across the domain at each time step.*
- In particle methods, we update the location of the zero-isocontour at each time step.

Particles

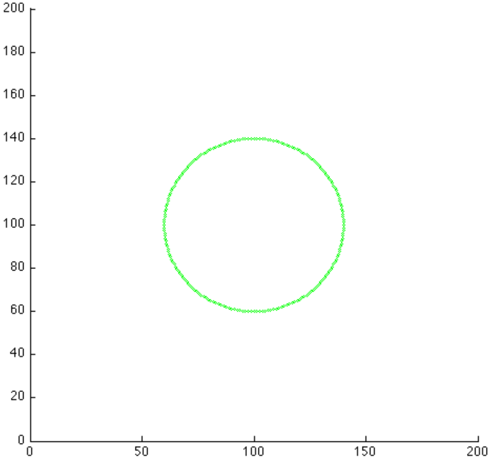


Figure: Explicit Representation of Circle in $\mathbb{R}^2$

Level Set and Particle Method: Tradeoffs

Level Set Method

Strengths

- Maintains distance function
- Handles topological changes

Weaknesses

- Computationally expensive
- Numerically Inaccurate

Particle Method

Strengths

- Computationally efficient and numerically accurate

Weaknesses

- Does not handle topological change (e.g. merging/pinching)

Existing Methods

Existing Methods

Interface Evolution Schemes

- Particle Level Set Method (Enright, Fedkiw, Ferziger, and Mitchell)
- Grid-Based Particle Method (Leung and Zhao)

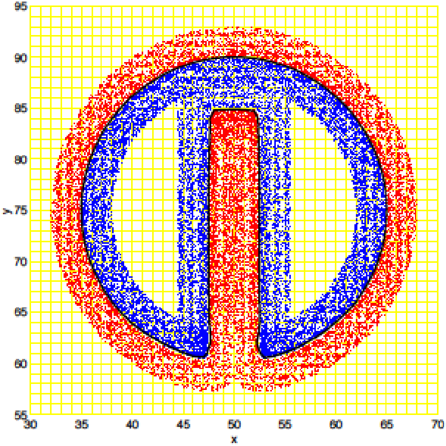
Surface Reconstruction Schemes

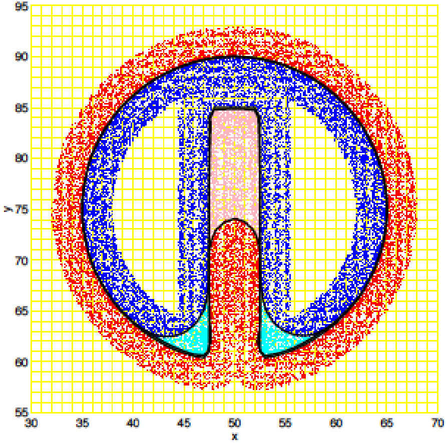
- Fast Variational-Based Surface Reconstruction (Ye, Bresson, Goldstein, and Osher)

Particle Level Set Method

- Seeds particles in a positive and negative band around the level set.
- Detects errors in level set when a particle crosses into a region of opposite sign.
- Overwrites level set signed distance function with particle signed distance function.
- Superior to basic level set method in preserving volume.
  Difficult to implement and reseeding strategy is not robust.

Particle Level Set Method

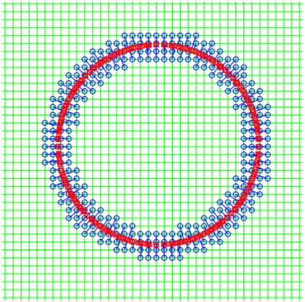




PLS vs Level Set Method. Light red and light blue particles are escaped particles.

Grid-Based Particle Method

- Uses Eulerian information from grid cells to compute curvature and normal vectors for particles.
- Uses a least-squares quadratic fit to approximate the interface locally.
- Lacks a robust method for determing inside / outside information for interface, but accurately computes distance.



Variational Method

- Estimates a surface from a set of unorganized scattered points using variational methods
- Solves an inverse edge detection problem to obtain an initial surface estimate
- We adapt this step of the method to align a level set with particles

(a) $p = 1$

(b) $p = \frac{1}{2}$

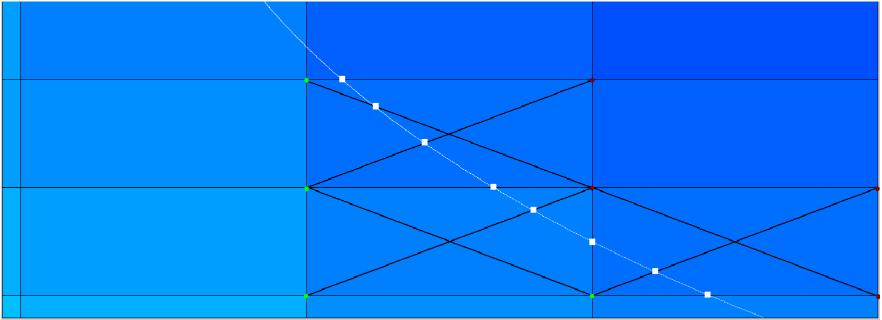
Our Method: Dynamic Reconstruction Method

Dynamic Reconstruction Method

Algorithm 5.1: DYNAMICRECONSTRUCTION(T, k, ϕ)

for $t = 1 : T$
if $(t \mod k) == 1$
  (re)seed particles on zero-isocontour of ϕ;
end
$\begin{cases} \text{advect particles;} \\ \text{calculate distance function } \phi \text{ using the particles;} \\ \text{compute sign (inside/outside) information using an edge detector;} \end{cases}$
end

1. (Re)seed particles on the zero-isocontour



GREEN dots mark exterior grid points, RED dots mark interior grid points. Unless a grid cell has unanimous sign on all grid points, linearly interpolate the zero-crossing on each of the 6 edges.

2. Advect Particles

Advect particles using 2[nd] order Runge-Kutta scheme: **for** t = 1:T

$\forall p \in \{Particles\}$
do:
$x_0 = x(p); y_0 = y(p);$
$x_1 = x_0 + v_x \cdot dt;$
$y_1 = y_0 + v_y \cdot dt;$
$x_2 = x_1 + v_x \cdot dt;$
$y_2 = y_1 + v_y \cdot dt;$
$x(p) = \frac{x_2 + x_0}{2}; y(p) = \frac{y_2 + y_0}{2};$
end

3. Compute Distance Using Particles

Fast-Sweeping Algorithm (Zhao, 2004)

We want to reconstruct the zero-isocontour using the particles.

1. **Initialization.** On cut grid cells: calculate distance to the interface using the particles. On the rest of the domain: initialize to a large constant, c.

3. Compute Distance Using Particles

Fast-Sweeping Algorithm (Zhao, 2004)

We want to reconstruct the zero-isocontour using the particles.

1. **Initialization.** On cut grid cells: calculate distance to the interface using the particles. On the rest of the domain: initialize to a large constant, c.
2. **Discretize, and iteratively sweep the domain in alternating directions.** At each interior grid point, solve the eikonal equation

$$|\nabla d|^2 \approx \frac{[(d^h_{i,j} - d^h_{x\,\min})^+]^2 + [(d^h_{i,j} - d^h_{y\,\min})^+]^2}{h^2} = f^2_{i,j}$$

and update $\bar{d}_{i,j}$ to be the minimum between c and the computed solution. Use one-sided difference on the boundary.

3. Compute Distance Using Particles

We solve $|\nabla d| = 1$ using the fast sweeping algorithm:

Ready

4. Calculate Sign Information Using an Edge Detector

Surface Reconstruction Using the Eikonal Equation and the Chan-Vese Model (Ye, Bresson, Goldstein, Osher, 2010)

- We want to approximate a two-valued function f whose edges are located along the set of particles, that is

$$f(x) \begin{cases} < C & \text{for } x \in \Omega^+, \\ \geq C & \text{for } x \in \Omega^-. \end{cases}$$

  where $0.5 < C < 1$ is a critical value that segments the domain.
- Let $d(\cdot)$ be the unsigned distance function. Observe that $-\nabla d$ is equivalent to a vector flow pointing toward the interface computed from particles. $d(\cdot)$ is an *edge detector function*.
- Using $\epsilon = \mathrm{d}x^p$ for stability, we can approximate the image f to this edge detector by solving the eikonal equation:

$$|\nabla f| = \frac{1}{d^p + \epsilon} \quad \text{for } p = 3$$

4. Calculate Sign Information Using an Edge Detector

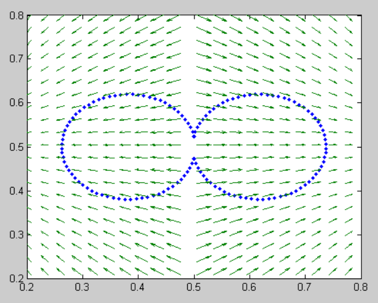
Surface Reconstruction Using the Eikonal Equation and the Chan-Vese Model (Ye, Bresson, Goldstein, Osher, 2010)

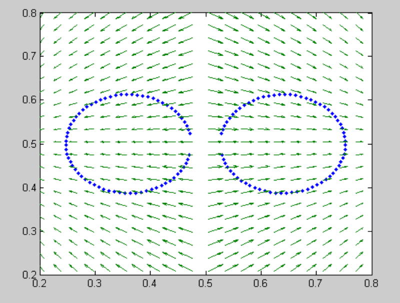
- We solve $|\nabla f| = \frac{1}{d^p+\epsilon}$ using the fast sweeping algorithm.

Ready

Regarding Reseeding

We lose detail when particles become too sparse.





To avoid saturating the interface with particles, we need to reseed to redistribute particles on the interface.

Regarding Reseeding

Ready

Reseeding allows us to use fewer particles.

Level Set Method vs. Dynamic Reconstruction

Ready

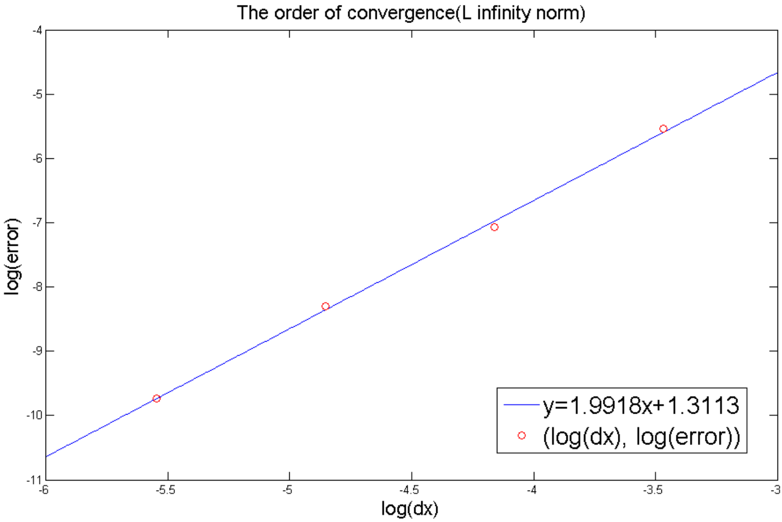
Level Set Method

3^{rd} order ENO advection and reinitialization

Dynamic Reconstruction Method

Reseed every 10 times

Dynamic Reconstruction Method: Accuracy



Dynamic Reconstruction Method: Merge Test

Ready

Conclusion

- We devised a method that utilizes particle advection to evolve the interface and surface reconstruction methods to reconstruct an implicit representation of the interface.
- Our scheme periodically reseeds to stabilize particle density and population.
- Our scheme gives second-order accuracy, and can be easily implemented.

Future Improvements

Future goals for this method include:

- Devise a consistent reseeding scheme
- Implement the method in 3D
- Test the method on a fluid velocity field
- Achieve more accurate sign computation

Thank You

We are grateful to our advisors, Profs. Jinsun Sohn and Joey Teran for teaching us the background and guiding us with our research!

Thank you, Prof. Andrea Bertozzi, for giving us this great opportunity!

References

Osher, S. and Fedkiw, R.P (2003)
Level set methods and dynamic implicit surfaces
Springer Verlag 153

Leung, S. and Zhao, H. (2009)
A grid based particle method for moving interface problems
Journal of Computational Physics 228

Enright, D. and Fedkiw, R. and Ferziger, J. and Mitchell, I. (2002)
A hybrid particle level set method for improved interface capturing
Journal of Computational Physics 153

Ye, J. and Bresson, X. and Goldstein, T. and Osher, S. (2010)
A Fast Variational Method for Surface Reconstruction from Sets of Scattered Points
UCLA CAM Report