Evaluation of Interface Tracking Schemes with Finite Element Discretizations

Jing Xu

23, November 2006

- Problem description
- Interface tracking methods

  - Standard methods

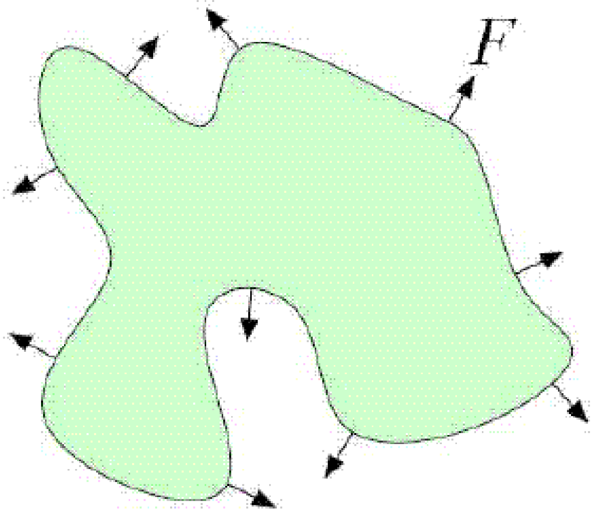
    - Volume of Fluid Method

    - Level Set Methods

  - Particle Level Set Method
- Comparison of interface tracking methods
- Conclusions

Evolving interfaces and surfaces

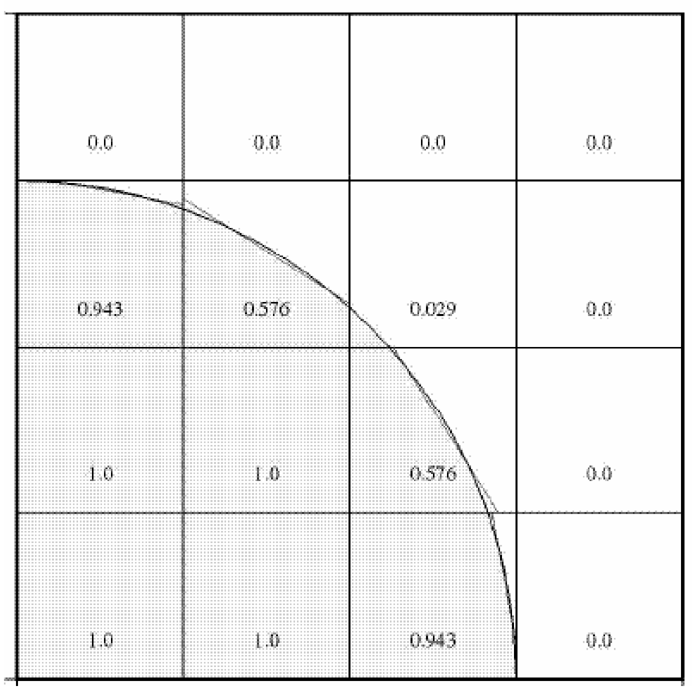
- Propagate interface according to speed function F
- F depends on space, time and the interface itself
- The goal is to model the evolution of the interface under the velocity F



The Volume of Fluid Method

- An algorithm for capturing interfaces discontinuously
- The interface is captured via the volume fraction f
- f is convected with the fluid

$$\frac{\partial f}{\partial t} + \vec{u} \cdot \nabla f = 0$$



Level Set Methods

- Numerical algorithm for tracking dynamic implicit interfaces
- Applications in

  -Computational fluid dynamics

  -Graphics

  -Fluid and combustion simulation

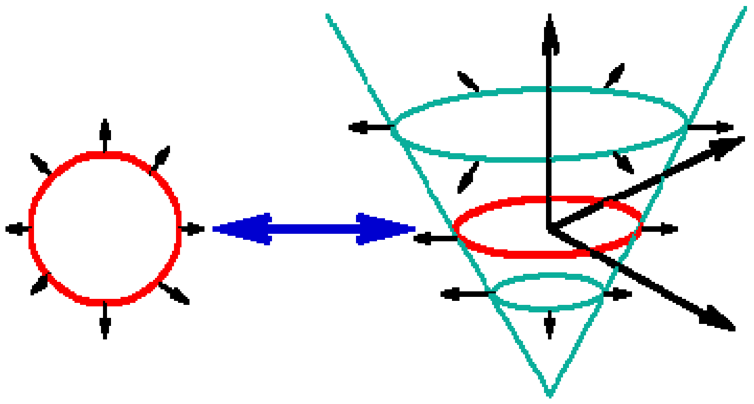
  -Image processing and computer vision

  -Robotics, control and dynamic programming

Level set function I

The underlying idea behind the level set method is to embed an interface Γ in R^m which bounds an open region $\Omega \subset R^m$ as the zero level set of a higher dimensional function $\phi(\vec{x},t)$.

$$\phi(\vec{x}, t=0) = \pm d$$



Level set function II

- The evolution equation for the level set function

$$\phi_t + \vec{u} \cdot \nabla\phi = 0$$

- Re-initialization algorithms

$$\phi_\iota + sgn(\phi_0)(|\nabla\phi| - 1) = 0$$

$sgn(\phi_0)$ is a one-dimensional smeared out signum function approximated numerically as

$$sgn(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + (\Delta x)^2}}$$

TP2D Solver

The level set/VOF solver in the TP2D code is essentially solving the partial differential equation:

$$\frac{\partial \phi}{\partial t} + (\mathbf{w} \cdot \nabla)\phi = 0$$

-For level set calculations

ϕ is initialized as a signed distance function, with $\phi = 0$ describing the interface

-For VOF calculations

ϕ is initialized as a Heaviside function, with $\phi = 0.5$ describing the interface

Time Discretization

Given ϕ^n *and the time step* $\Delta t = t_{n+1} - t_n$ *, then solve for* $\phi = \phi^{n+1}$

$$\frac{\phi - \phi^n}{\Delta t} + \theta[(\mathbf{w} \cdot \nabla)\phi] = (\theta - 1)(\mathbf{w} \cdot \nabla)\phi^n$$

Where $\theta = 1$ will be the simple first-order Backward Euler-scheme (BE), and $\theta = 0.5$ the secon-order Crank-Nicolson-scheme (CN). We used $\theta = 0.5$ in our simulations.

Space Discretization

We discretize in space with the finite element method and choose bilinear quadrilaterals ($\mathbb{Q}_1$) as the trial space to search for a solution in.

$$(\mathbb{M}_{\mathbb{L}} + \Delta t \theta \mathbb{K})\mathbf{u}_h^{n+1} = \mathbf{g}$$

$\mathbf{u}_h^{n+1}$: the sought unknown solution vector at time level $n+1$

$\mathbb{M}_{\mathbb{L}}$: the lumped mass matrix of

$$\mathbb{M} = \int_\Omega \psi_i \psi_j \; d\Omega$$

$\mathbb{K}$: the convective iteration matrix

$$\mathbb{K} = \int_\Omega \frac{\partial \psi_i}{\partial x_k} \psi_j \; d\Omega$$

$\mathbf{g}$: the right hand side vector

$$\mathbf{g} = [\mathbb{M}_{\mathbb{L}} + \Delta t(\theta - 1)\mathbb{K}]\mathbf{u}_h^n$$

The Particle Level Set Method

- The method combines the best properties of an Eulerian level set method and a marker particle Lagrangian scheme

- The method utilizes particles to assist the level set in accurately tracking flow characteristics in under-resolved regions and consequently preserve mass

Particle Initialization

Two sets of massless marker particles are placed randomly inside of grid cells which are within three cells of the interface.

- positive particles with $s_p = +1$ in $\phi > 0$
- negative particles with $s_p = -1$ in $\phi < 0$

Each particle has a variable radius r_p

$$r_p = \begin{cases} r_{\max}, & if\ s_p\phi(\vec{x}_p) > r_{\max} \\ s_p\phi(\vec{x}_p), & if\ r_{\min} \le s_p\phi(\vec{x}_p) \le r_{\max} \\ r_{\min}, & if\ s_p\phi(\vec{x}_p) < r_{\min} \end{cases}$$

Time Integration

The particles are advected with

$$\frac{d\vec{x}_p}{dt} = \vec{u}(\vec{x}_p)$$

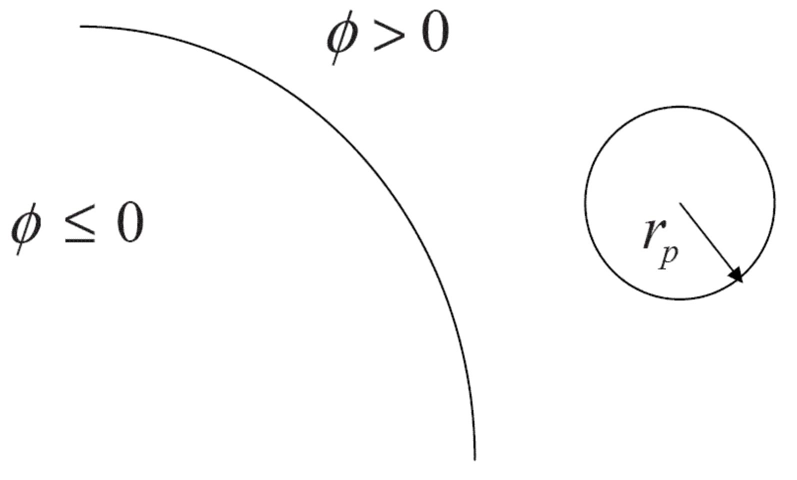
which can be discretized as follows:

$$\vec{x}_p(t) = \vec{x}_p(t-1) + \Delta t \; \vec{u}(\vec{x}_p(t-1))$$

Error Correction I

- Identifying Errors

  An error is detected if a particle is on the wrong side of the interface by more than its radius



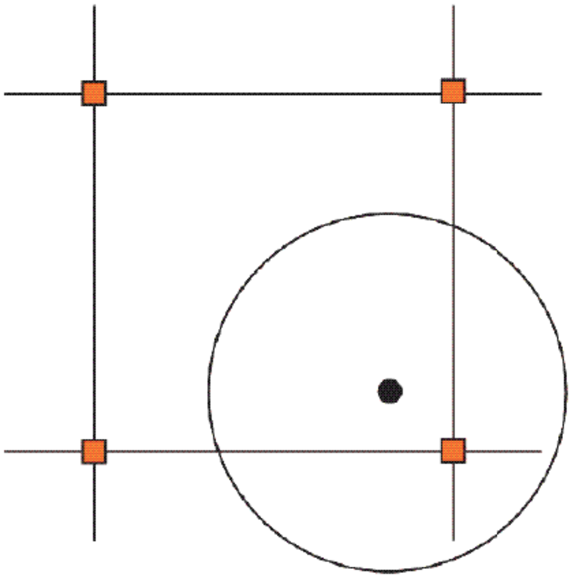
Error Correction II

- Quantifying Errors

A level set value for each particle is defined as follows:

$$\phi_p(\vec{x}) = s_p(r_p - |\vec{x} - \vec{x}_p|)$$

$\phi_p(x)$ is defined only on the nodes of the cell which contains particle p



Error Correction III

- Reducing Errors

Use escaped particles to form reduced-error representation of ϕ

1) build ϕ^+ and ϕ^- as

$$\phi^+ = \max_{\forall p \in E^+} (\phi_p, \phi^+)$$

$$\phi^- = \min_{\forall p \in E^-} (\phi_p, \phi^-)$$

2) Merging of ϕ^+ and ϕ^- to a total ϕ

$$\phi = \begin{cases} \phi^+, & if\ |\phi^+| \leq |\phi^-| \\ \phi^-, & if\ |\phi^+| > |\phi^-| \end{cases}$$

Computational Order

1) Evolution of the particles and the level set function.
2) Error correction in the level set function using particles.
3) Re-initialization.
4) Error correction in the level set function using particles.
5) Radii adjustment.

Particle Reseeding

- Particle Addition

  Particles will periodically be added to the cell which is near the interface and has fewer particles than a previously defined maximum

- Particle Deletion

  Particles which are too far from the interface to provide any useful information will periodically be deleted

Test Cases

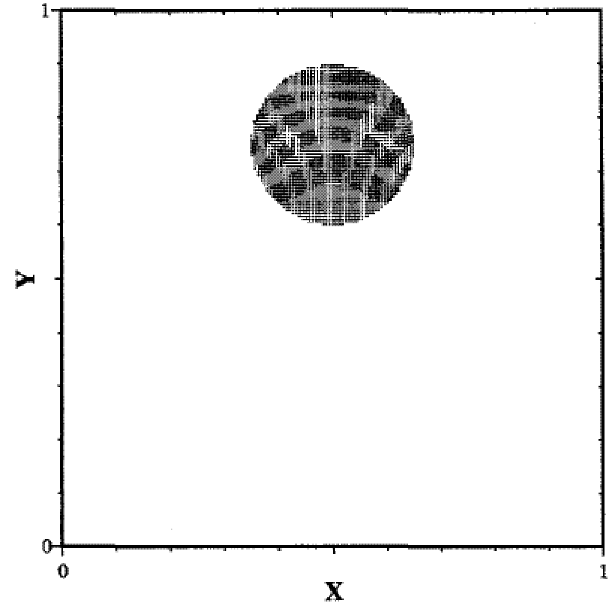
- Initial conditions

  - for VOF method

  the volume fraction is 1 inside the circle and 0 outside

  - for LS and PLS methods

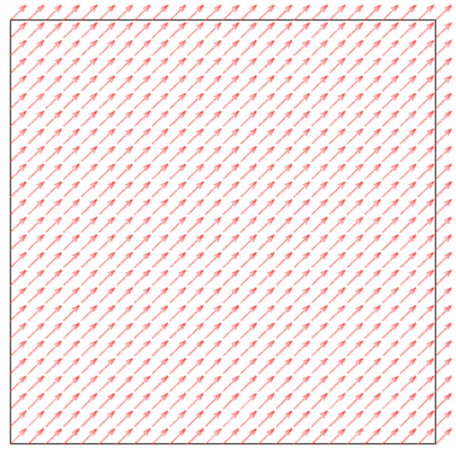
$$\phi = \sqrt{(x-0.5)^2 + (y-0.75)^2} - 0.15$$



Test Cases

- Case 1: simple translation

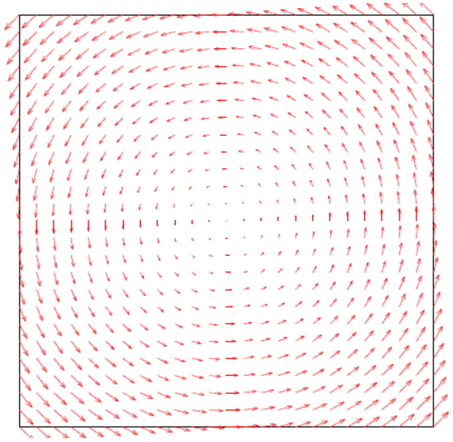
$$u = 1, \quad v = 1$$



A constant velocity field for case 1

- Case 2: solid body rotation

$$u = 2\pi(0.5 - y), \quad v = 2\pi(x - 0.5)$$



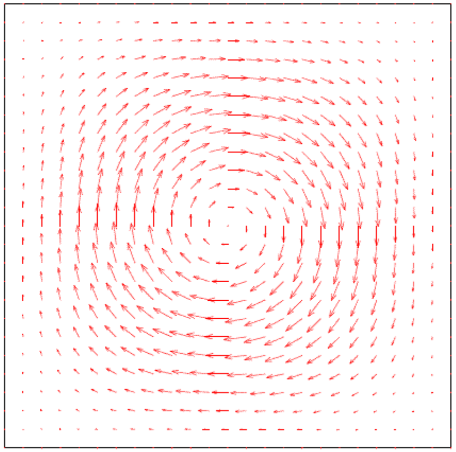
A constant-vorticity velocity field for case 2

Test Cases

- Case 3 : single vortex

$$u = -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y),$$

$$v = 2\sin(\pi x)\sin^2(\pi y)\cos(\pi x)$$



Single vortex flow

Case 3 : Single vortex

$1/h$	Density error	Order	Relative area change	Order	Circularity
40	0.06603	–	1.0000	–	inf
80	0.02500	1.4012	0.31689	1.6579	0.97008
160	0.00899	1.4755	0.05443	2.5415	0.99240

Table 1: Relative error norms and convergence rates for the single vortex problem with VOF when $t_{end} = 4$.

$1/h$	Density error	Order	Relative area change	Order	Circularity
40	0.06603	–	1.0000	–	inf
80	0.01997	1.7256	0.21011	2.2508	0.97073
160	0.01042	0.9387	0.01698	3.6292	0.99125

Table 2: Relative error norms and convergence rates for the single vortex problem with the standard level set method when $t_{end} = 4$

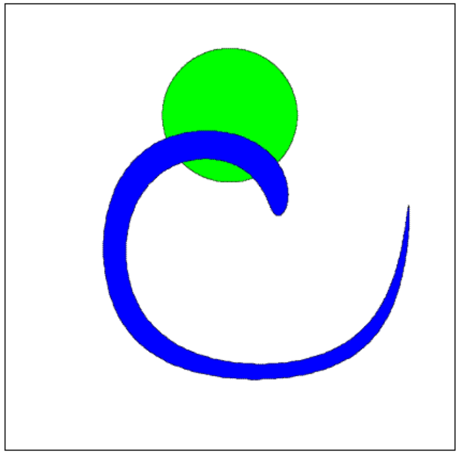
$1/h$	Density error	Order	Relative area change	Order	Circularity
40	0.00357	–	0.06099	–	0.99795
80	0.00015	4.5729	0.02235	1.4483	0.99751
160	0.00023	-0.6167	0.00986	1.1806	0.99752

Table 3.10: Relative error norms and convergence rates for the single vortex problem with the particle level set method with 8 particles in each cell when $t_{end} = 4$.

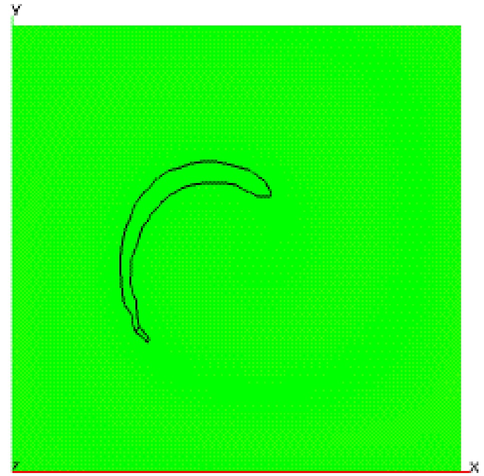
$1/h$	Density error	Order	Relative area change	Order	Circularity
40	0.00178	–	0.04322	–	0.99771
80	0.00015	3.5496	0.01876	1.204	0.99806
160	0.00019	-0.3398	0.00968	0.9546	0.99922

Table 3.11: Relative error norms and convergence rates for the single vortex problem with the particle level set method with 16 particles in each cell when $t_{end} = 4$.

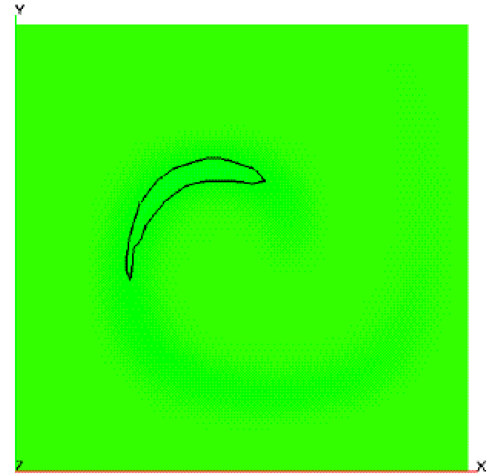
Comparison of methods on a 40×40 cell computational grid for the single vortex problem at $t = 2$ when $t_{end} = 4$



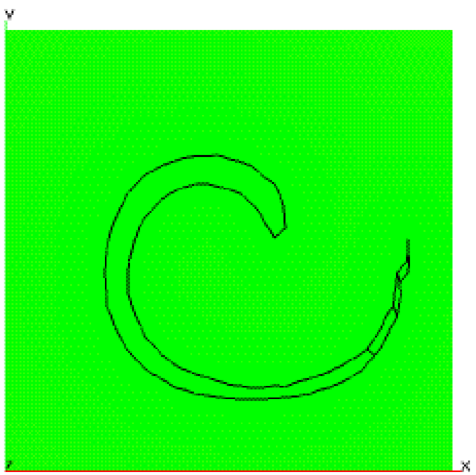
Solution for single vortex at t=2



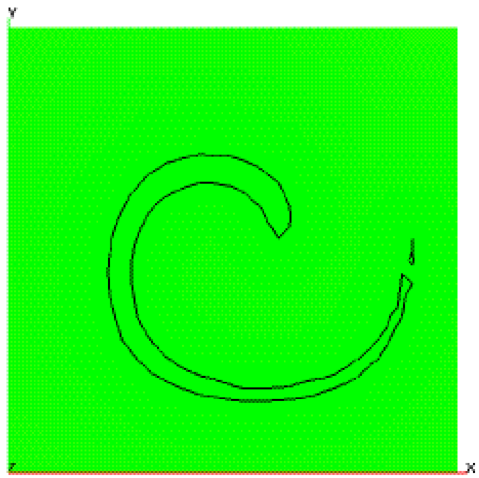
LS



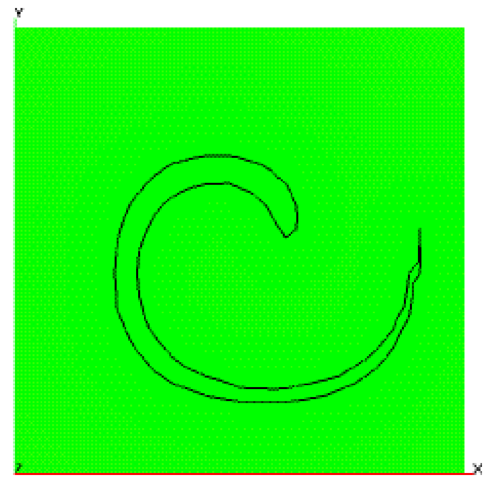
VOF



PLS with 8 particles in each cell

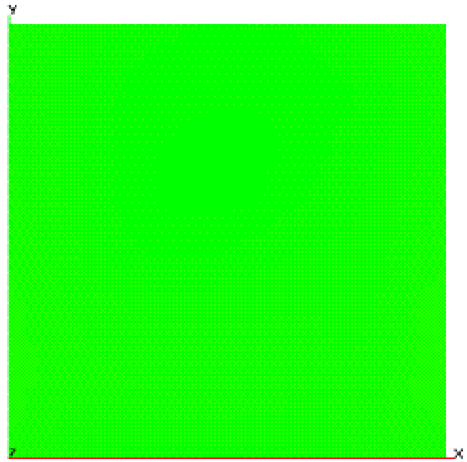


PLS with 16 particles in each cell

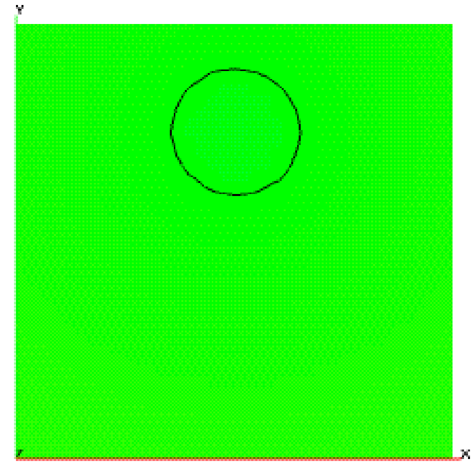


PLS with 32 particles in each cell

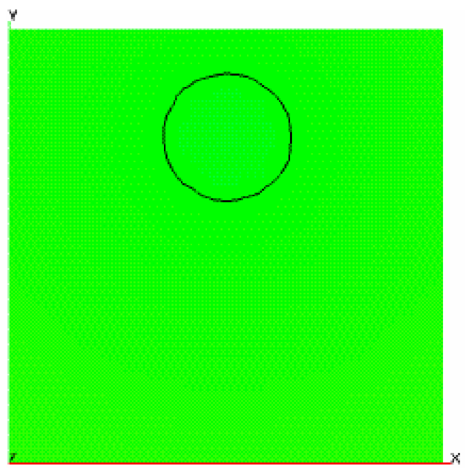
Comparison of methods on a 40×40 cell computational grid for the single vortex problem at $t = 4$ when $t_{end} = 4$



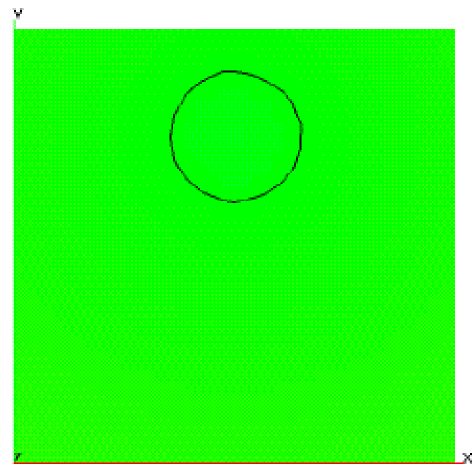
LS/VOF



PLS with 8 particles in each cell



PLS with 8 particles in each cell



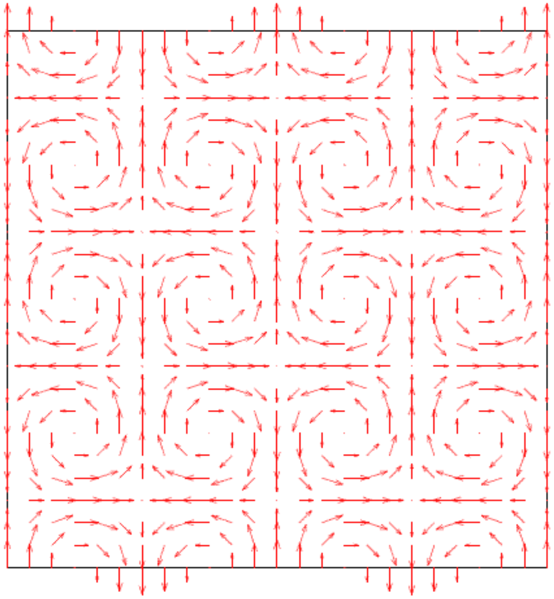
PLS with 32 particles in each cell

Test Cases

- Case 4 : deformation problem

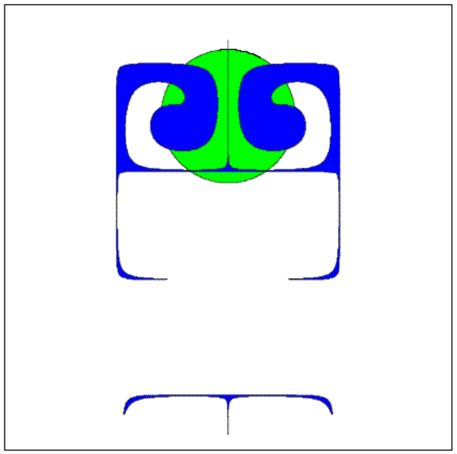
$$u = \sin(4\pi(x + \tfrac{1}{2}))\sin(4\pi(y + \tfrac{1}{2})),$$

$$v = \cos(4\pi(x + \tfrac{1}{2}))\cos(4\pi(y + \tfrac{1}{2}))$$

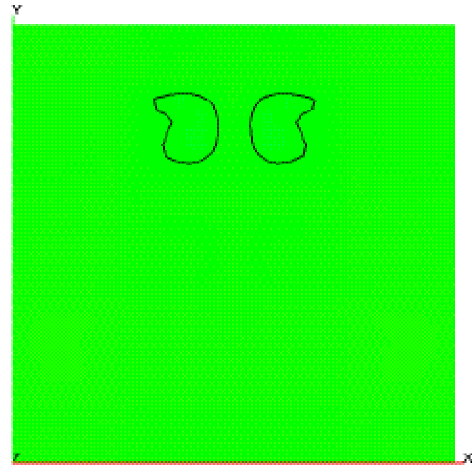


Deformation velocity field

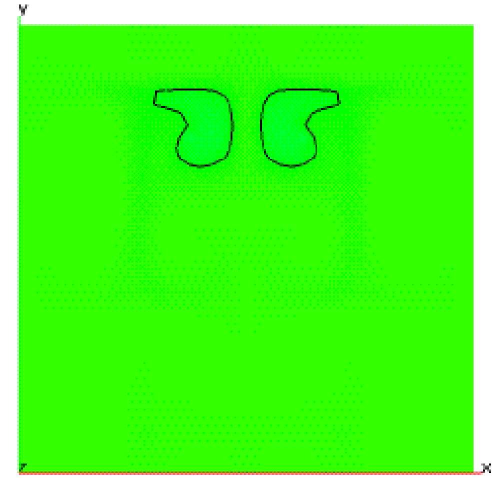
Comparison of methods on a 40×40 cell computational grid for the deformation problem at $t = 1$ when $t_{end} = 2$



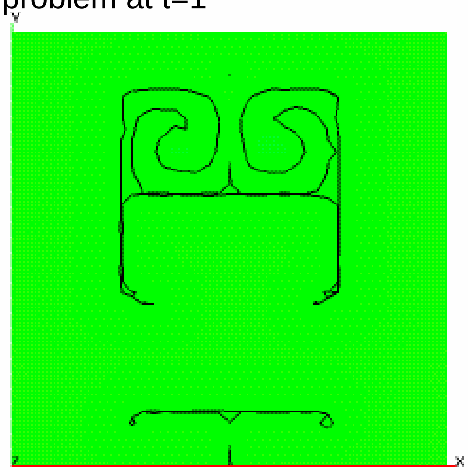
Solution for deformation field problem at t=1



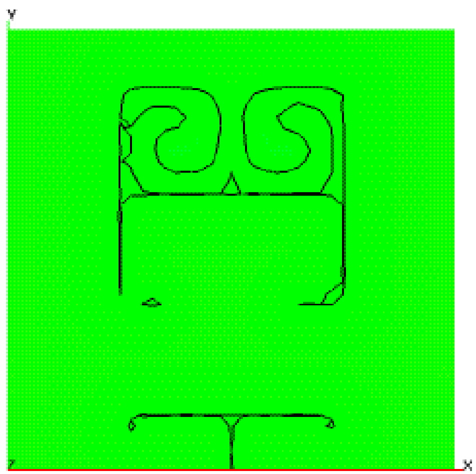
LS



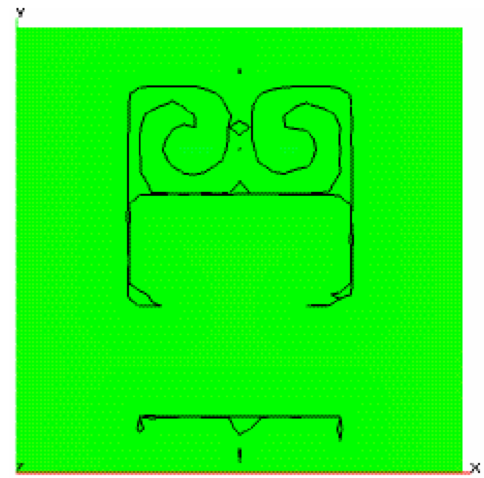
VOF



PLS with 8 particles in each cell

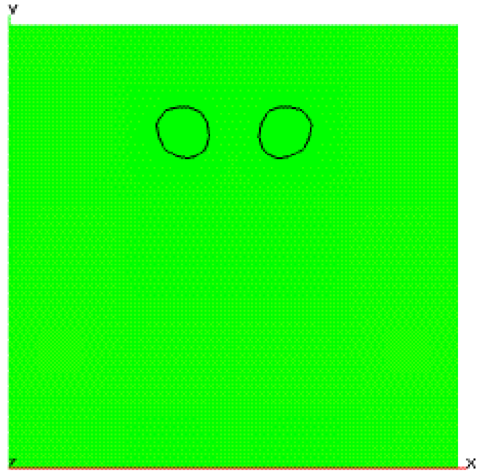


PLS with 16 particles in each cell

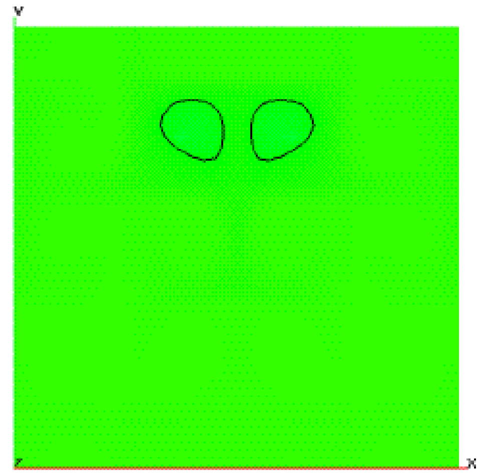


PLS with 32 particles in each cell

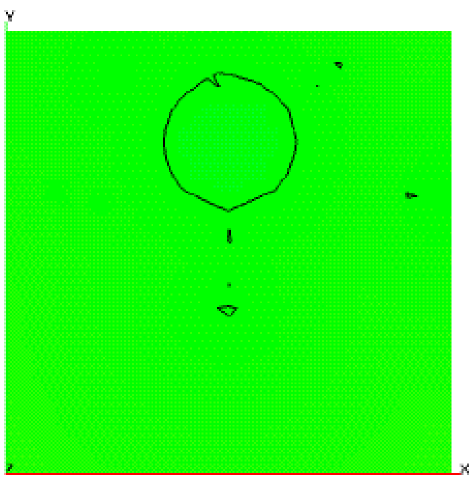
Comparison of methods on a 40×40 cell computational grid for the deformation problem at $t = 2$ when $t_{end} = 2$



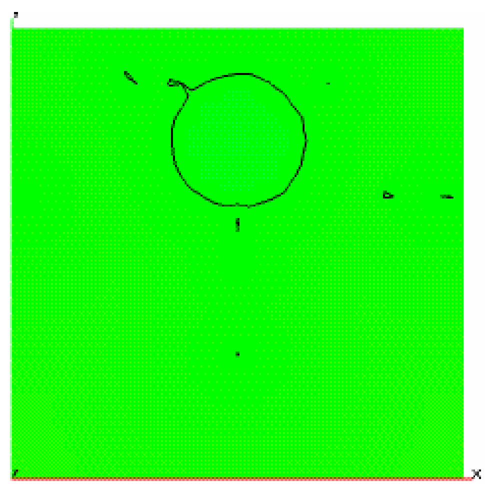
LS



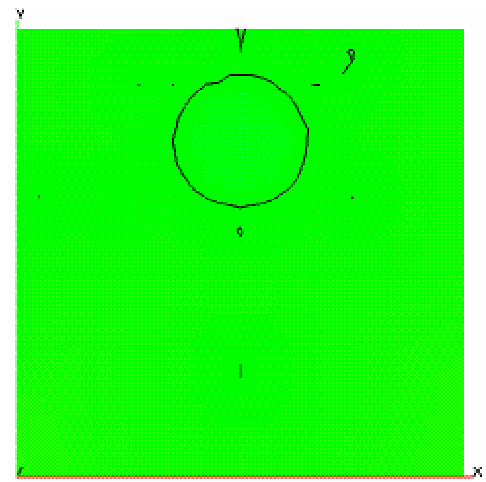
VOF



PLS with 8 particles in each cell

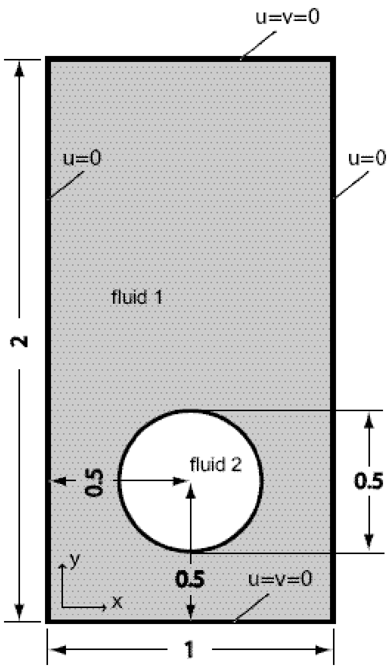


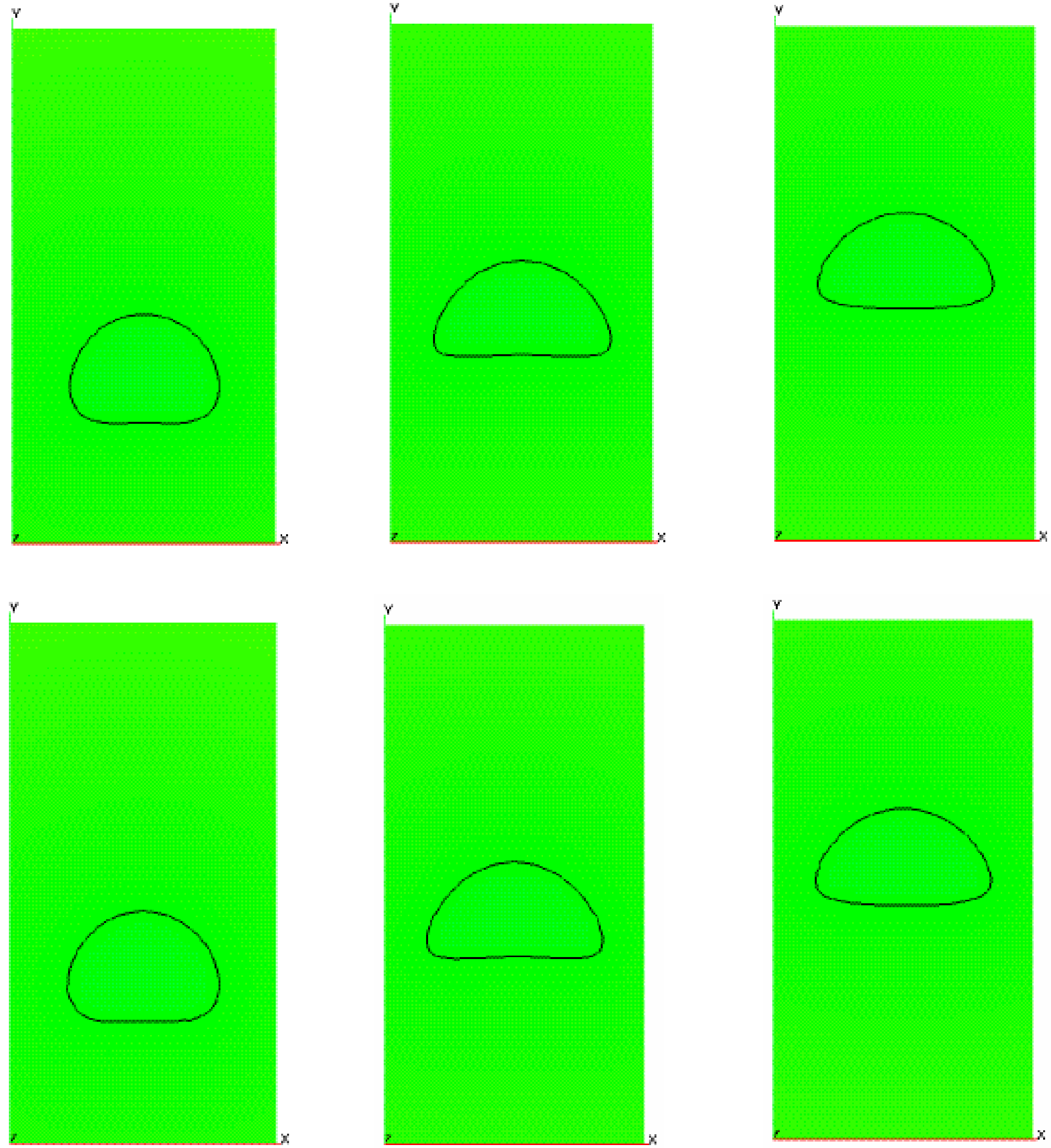
PLS with 16 particles in each cell



PLS with 32 particles in each cell

Rising Bubble Problem

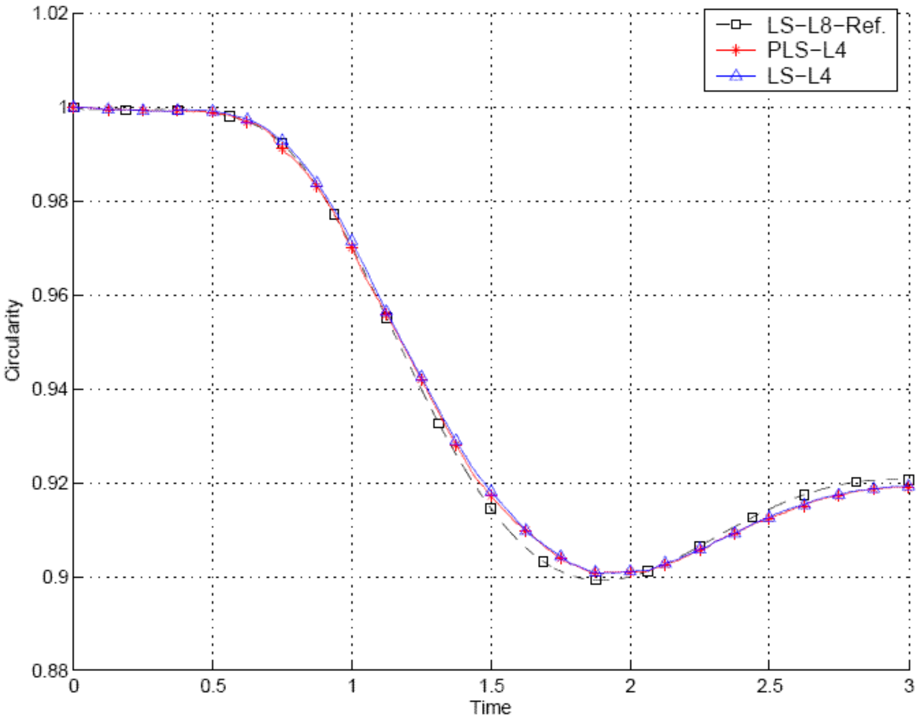




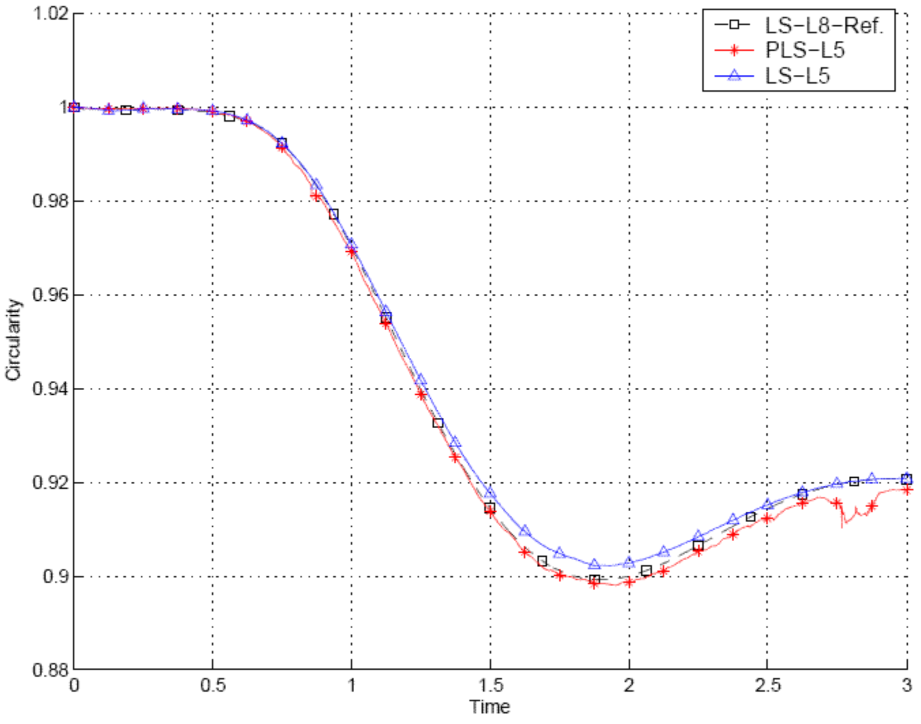
LS

PLS

Circularity for the rising bubble problem

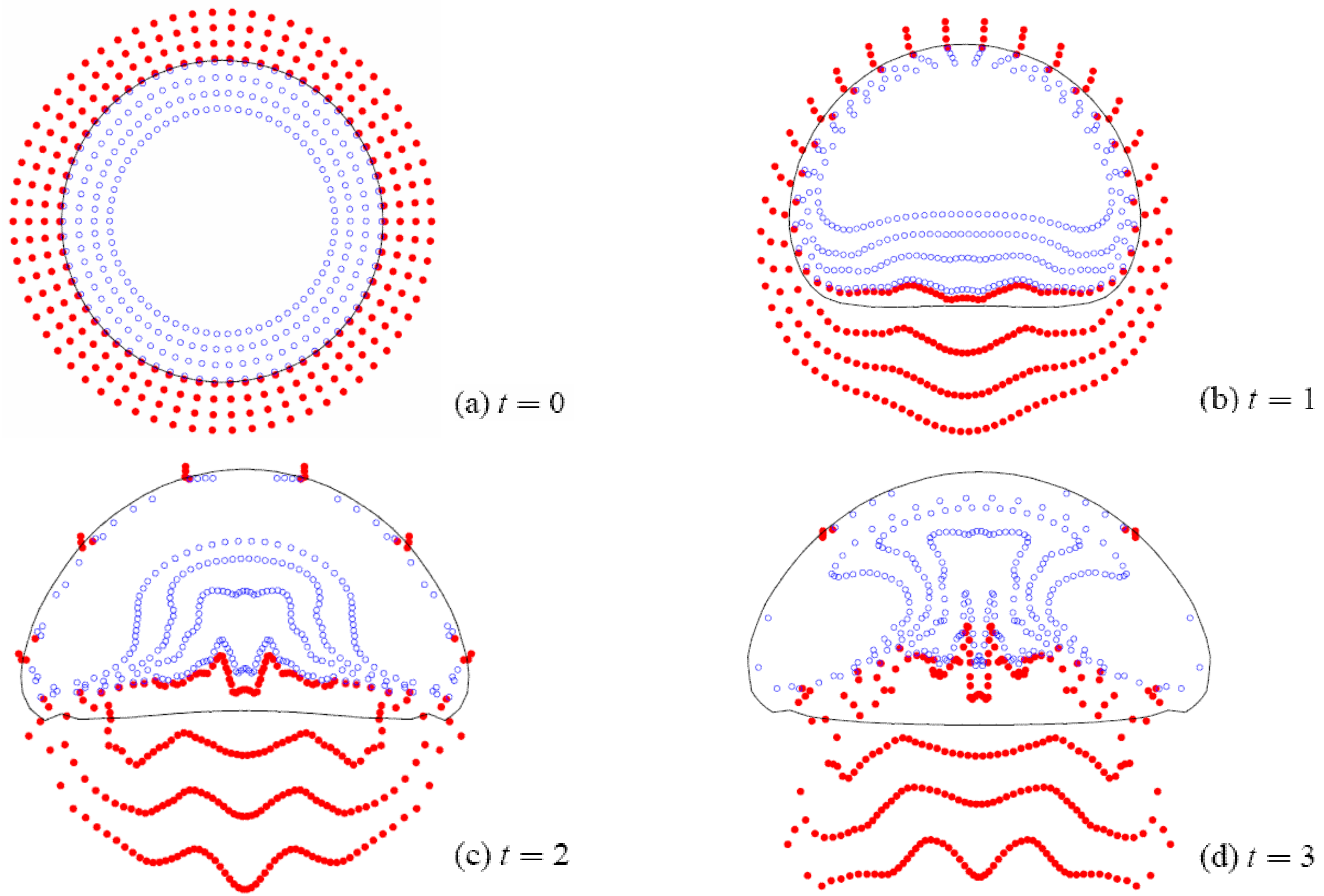


40×40



80×80

Convection of the particles for the rising bubble problem



Conclusions I

- VOF
  - Advantages : robust, may preserve mass inherently
  - Disadvantages : not so accurate, difficult to implement
- LS
  - Advantages:
    - Easy to represent a variety of shapes
    - Unified framework for many types of motion
    - Geometrical quantities easily and accurately evaluated
    - Topological changes are automatic
    - Easy to implement
  - Disadvantages:
    - Mass loss

Conclusions II

- PLS

  - Advantages : accurate, robust, easy implementation, preserve mass

  - Disadvantages: computation time-consuming

  - Open issues :

    the number of particles necessary

    how to make particles follow the interface correctly

    ……