ILASS Americas 19th Annual Conference on Liquid Atomization and Spray Systems, Toronto, Canada, May 2006

A New Method for Calculating Interface Normal Vectors and Curvatures when Modelling Free-Surface Flows

M. Raessi[*], J. Mostaghimi and M. Bussmann
Department of Mechanical and Industrial Engineering
University of Toronto
5 King's College Road, Toronto, Ontario, Canada M5S 3G8

Abstract

In simulating free-surface flows, the interface normal vectors and curvature are needed for modelling surface tension effects. We present a new method for accurately calculating these quantities. In this method, an entirely different approach is taken, compared to common methods where normals and curvature are obtained from the Volume-of-Fluid (VOF) or Level Set (LS) functions. In this fixed-grid method, the interface normals are advected along with the interface, although the methods of advecting normals and the interface are independent. In this work, the interface is tracked using the VOF method. The advected normals are used to reconstruct the interface at each timestep, and calculate the interface curvature at any point. The mathematical formulation for the advection of normals is derived from the evolution equation of the LS function, but the method is independent of the LS function. The implementation of the new method is straightforward, and its accuracy is demonstrated via two test cases: translation and rotation of a circle. The results show that this method is second-order accurate in calculating interface normals and curvature. Furthermore, the performance of the method has been examined in a flow field which includes shear.

[*] Corresponding Author

Introduction

In simulating free-surface flows, accurate modelling of surface tension effects is a challenging task. Nonphysical velocities (spurious currents) are easily induced in the flow if the surface tension force is not modelled properly. There are two sources of error in modelling surface tension effects: (a) interface curvature calculation and (b) the implementation method of the surface tension force. Recently, Francois et al. [1] proposed a method for implementing surface tension into a Volume-of-Fluid (VOF) model which imposes an exact balance between the surface tension and pressure forces. In this method, no spurious currents are induced in the flow provided that the interface curvature is exact.

This paper, then, focuses on the calculation of accurate interface curvature and normals. There are different methods for calculating interface curvature: A common method is to calculate the first and second derivatives of volume fractions (VOF function) in the VOF method to obtain normals and curvature. It is shown later in this paper that curvatures calculated from the volume fractions are associated with errors which grow linearly when increasing the mesh resolution. Furthermore, the normals obtained from the volume fractions are zero-order accurate in space; i.e. they do not converge with mesh refinement. An increasingly popular alternative is to calculate normals and curvatures by taking the derivatives of a Level Set (LS) function [2]. In general, the LS function yields these quantities more accurately than the VOF function. However, in order to conserve mass (which is very important in modelling multiphase flows) the LS function needs to be reinitialized periodically during the interface evolution. Among different reinitialization methods, only the coupled level set and VOF approach seems to achieve mass conservation exactly. However, it is demonstrated that curvatures calculated from the LS function of a coupled LS and VOF model are zero-order accurate in space.

Taking the derivatives of the VOF or LS function is an straightforward and easy approach for calculating interface curvature. There are other methods for computing curvature which are more accurate and sophisticated. The height-function method [3], for example, yields converging, 2$^{\text{nd}}$-order accurate curvatures. In this method, a "height" function is reconstructed directly from the volume fractions. By taking the second derivative of the height function the interface curvature is obtained. Poo and Ashgriz [4] utilized a second-order polynomial curve fit to calculate the curvature of an interfacial cell in two-dimensional space. In a method known as PROST [5], the interface in each cell is fitted with a paraboloid. The paraboloid with an optimal fitting, which is found iteratively, generates almost the same volume in a 3×3 stencil in 2D or 3×3×3 in 3D. The curvature is then calculated from the quadratic equation of the paraboloid.

Although these methods are able to yield curvature with second-order accuracy, their implementation is not easy, and they become intricate in complicated interface geometries such as liquid breakup.

This paper, then, introduces a new method for accurately calculating interface curvature and normals. The structure of the paper is as follows. First, we evaluate two current methods: taking derivatives of the VOF and LS functions for calculating curvature and normals. Then, the fundamentals and the implementation of the new method are presented and the method is compared with the above two methods. Finally, through various test cases, the accuracy and the performance of the new method is demonstrated.

Existing numerical methods for calculating interface unit normal vectors and curvature

In this section, we discuss and assess two methods which have been widely used for calculating interface normals and curvature: using the volume fractions in the VOF method, or employing the level set function in the Coupled Level Set and VOF (CLSVOF) method to calculate normals and curvatures.

Volume-of-Fluid (VOF) method

In the VOF method, a scalar color function, f, defined as

$$f(\vec{x}) = \begin{cases} 1, & \vec{x} \in \text{liquid} \\ 0, & \vec{x} \notin \text{liquid} \end{cases} \tag{1}$$

is used to track the interface via the following advection equation

$$\frac{\partial f}{\partial t} + \vec{u} \cdot \nabla f = 0 \tag{2}$$

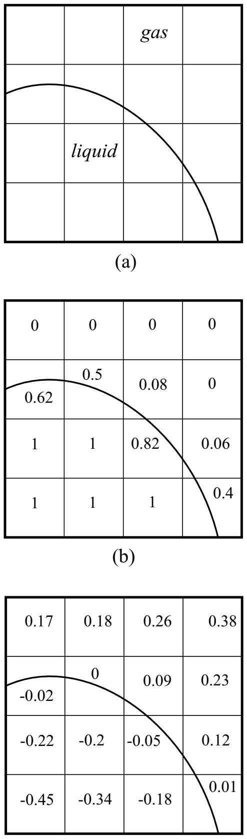
The interface unit normal vector is calculated by taking the gradient of f,

$$\hat{n} = \frac{\nabla f}{|\nabla f|} \tag{3}$$

The curvature of interface is computed by

$$\kappa = -\nabla \cdot \hat{n} \tag{4}$$

The VOF function is discontinuous by definition. The discretized form of the VOF function is



(a)

(b)

(c)

Figure 1. (a) A typical liquid-gas interface, (b) the VOF function for the liquid, and (c) the level set function for the liquid.

the volume fraction of a numerical cell ij occupied by the liquid, and is defined as

$$f_{ij} = \frac{1}{\Omega_{ij}} \int_{\Omega_{ij}} f d\Omega \tag{5}$$

For a typical two-dimensional interface depicted in Figure 1(a), the discretized VOF function, defined at the center of each cell, is shown in Figure 1(b). As can be seen, the volume fractions vary sharply from zero to one across one cell. This discontinuous behavior of f makes it difficult to find first and second derivatives of f and thus, leads to inaccurate interface normals and curvatures.

As a test problem, the normals and curvatures of a 0.1m radius circle, centered at (0.5,0.5) in a 1×1m domain are calculated via equations (3) and (4) using the VOF function. For any quantity q, the l_∞ and l_1 errors are defined as follows:

$$l_\infty = \max_j |(q_{cal.} - q_{exact})_j| \tag{6}$$

$$l_1 = \frac{1}{N} \sum_{j=1}^{N} |(q_{cal.} - q_{exact})_j| \tag{7}$$

The errors associated with unit normal vectors at cell corners calculated by the VOF method are presented in Table 1. The normals are zero-order accurate in space, $\hat{n}_{cal.} = \hat{n}_{exact} + O(\Delta x^0)$, i.e. there is a constant error associated with the normals, which does not vanish as the mesh is refined.

$R/\Delta x$	l_∞	order	l_1	order
10	0.5429		0.0669	
		-0.18		-0.01
20	0.6162		0.0673	
		-0.07		0.04
40	0.6463		0.0656	
		-0.22		-0.02
80	0.7506		0.0666	

Table 1. The errors associated with unit normal vectors at cell corners calculated by the VOF method, for a 0.1m radius circle at different mesh resolutions. The circle is centered at (0.5,0.5) in a 1×1m domain.

$R/\Delta x$	l_∞	order	l_1	order
10	30.8		4.4	
		-1.12		-0.54
20	66.8		6.4	
		-1.02		-0.84
40	135.4		11.4	
		-1.00		-0.93
80	270.7		21.8	

Table 2. The errors associated with curvatures calculated by the VOF method, for a 0.1m radius circle at different mesh resolutions. The circle is centered at (0.5,0.5) in a 1×1m domain.

The errors of the curvature values calculated by the VOF method are presented in Table 2. It can be seen that both l_∞ and l_1 grow linearly when increasing the mesh resolution, $\kappa_{cal.} = \kappa_{exact} + O(1/\Delta x)$. This is a serious disadvantage of the VOF method. Contrary to what one would hope, increasing the mesh resolution deteriorates the accuracy of the curvature values.

To rectify this problem the level set method, which was believed to yield more accurate normals and curvatures, was implemented. The fundamentals and implementation of the level set method,

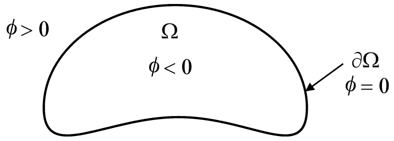


Figure 2. A representation of the level set function ϕ for a typical domain Ω

along with its performance in calculating normals and curvatures, will be presented next.

Coupled Level Set and VOF (CLSVOF) method

In the level set (LS) method [2, 6], the interface is tracked by a LS function ϕ which for a domain Ω (Figure 2) is defined as a signed distance to the boundary $\partial\Omega$

$$|\phi(\vec{x})| = \min(|\vec{x} - \vec{x_I}|) \quad \text{for all} \quad \vec{x_I} \in \partial\Omega \qquad (8)$$

implying that $\phi(\vec{x}) = 0$ on the boundary where $\vec{x} \in \partial\Omega$. Thus,

$$\phi(\vec{x}) = \begin{cases} < 0, & \vec{x} \in \Omega \\ 0, & \vec{x} \in \partial\Omega \\ > 0, & \vec{x} \notin \Omega \end{cases} \qquad (9)$$

For a typical two-dimensional interface depicted in Figure 1(a), the discretized level set function ϕ, which is defined at the center of each cell, is shown in Figure 1(c).

The unit normal vector and curvature at any point on the liquid interface are calculated from the LS function by

$$\hat{n} = \frac{\nabla\phi}{|\nabla\phi|} \qquad (10)$$

and

$$\kappa = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right) \qquad (11)$$

Since the LS function ϕ is smooth and continuous across the interface (see Figure 1(c)), spatial derivatives can be easily discretized to calculate interface curvature and normal vectors.

In the LS method, the motion of the interface is defined by the following advection equation

$$\frac{\partial\phi}{\partial t} + \vec{u} \cdot \nabla\phi = 0 \qquad (12)$$

There are a number of schemes for discretizing equation (12). In this study, the spatial derivatives in (12) were discretized using a second-order accurate, essentially nonoscillatory (ENO) scheme. The forward Euler scheme was used to discretize the temporal derivative.

When ϕ is advected, the $\phi = 0$ contour moves at the correct velocity and properly represents the interface; however, contours of $\phi \neq 0$ do not necessarily remain distance functions. This can result in an irregular ϕ field and lead to violation of mass conservation. To rectify this problem reinitialization methods have been proposed, which adjust ϕ back to a signed distance function without changing the zero level set.

There are mathematical and geometrical approaches for reinitializing the LS function. The mathematical approach reinitializes $\phi(\vec{x})$ by solving a partial differential equation to steady state. The method is detailed in [7, 8] and is not presented here. Although the mathematical approach improves conservation of mass considerably, it cannot achieve it exactly.

In this study, a geometrical approach was implemented for reinitializing the level set function. In this method, the VOF function is advected along with the LS function and is used to reinitialize the LS function by the following procedure. Given the LS and VOF functions, ϕ^{n+1} and f^{n+1}, the interface, approximated as piecewise linear, is reconstructed from f^{n+1} using the interface normal vectors calculated from ϕ^{n+1}. The LS function is then reinitialized by calculating the distance between any cell center (where the LS function is defined) and the interface. This method is called "coupled level set and volume-of-fluid"(CLSVOF) and it guarantees exact mass conservation because it is based on the volume fractions.

The reinitialization method used in this paper is that of Son and Hur [9] and is not presented here.

The order of accuracy when reinitializing the LS function and calculating unit normal vectors and curvatures from the LS function

Similar to the evaluation of the VOF function presented earlier, the test problem is a 0.1m radius circle, centered at (0.5,0.5) in a 1×1m domain. The errors associated with the reinitialized ϕ, and the order of accuracy of the calculations are presented in Table 3.

The errors associated with the unit normal vectors and curvatures calculated from the LS function via equations (10) and (11) are presented in Tables 4 and 5, respectively. As the results in Tables 3 to 5 show, the level set function and normal vectors are second-order and first-order accurate in space, respectively; but the curvature values are unfortunately zero-order accurate, which means a constant

$R/\Delta x$	l_∞	order	l_1	order
10	1.83e-4		4.36e-5	
		1.97		2.20
20	4.68e-5		9.52e-6	
		1.83		2.11
40	1.32e-5		2.20e-6	
		2.04		1.90
80	3.22e-6		5.91e-7	

Table 3. The errors associated with the reinitialized level set function for a 0.1m radius circle, centered at (0.5,0.5) in a 1×1m domain, at different mesh resolutions.

$R/\Delta x$	l_∞	order	l_1	order
10	0.0186		0.00401	
		0.96		1.17
20	0.0096		0.00178	
		1.01		1.08
40	0.0047		0.00084	
		0.87		0.97
80	0.0026		0.00043	

Table 4. The errors associated with the unit normal vectors at cell corners calculated by the level set method, for a 0.1m radius circle at different mesh resolutions. The circle is centered at (0.5,0.5) in a 1×1m domain.

error, ranging from 10 to 20%, is always associated with the curvature values regardless of the mesh resolution.

The accuracy of the CLSVOF method used in this research project is consistent [10] with the accuracy of the CLSVOF model developed by Sussman and Puckett [11]. Although, using the LS function significantly improves the accuracy of normal vectors and curvature values compared to using the VOF method, it does not provide converging curvature values.

A new method for accurate calculation of interface normals and curvature is presented next. In this method, interface normals are advected along with the interface. The interface curvature is then calculated directly from the unit normals.

A new method for calculating interface normals and curvature: advecting the normals

Mathematical fundamentals

As reviewed earlier, the evolution of the level set function is defined by the following advection equation:

$R/\Delta x$	l_∞	order	l_1	order
10	1.88		0.460	
		0.01		0.29
20	1.87		0.375	
		-0.01		0.16
40	1.88		0.335	
		0.02		0.02
80	1.86		0.339	

Table 5. The errors associated with the curvature values calculated by the level set method, for a 0.1m radius circle at different mesh resolutions. The circle is centered at (0.5,0.5) in a 1×1m domain.

$$\frac{\partial \phi}{\partial t} + \overrightarrow{u} \cdot \nabla \phi = 0$$

The above equation can be written in the following form, too

$$\frac{\partial \phi}{\partial t} + u_i N_i = 0 \tag{13}$$

where

$$N_i = \frac{\partial \phi}{\partial x_i}$$

and N_i denotes the component of the interface normal vector in the i-direction.

By taking the gradient of equation (13), we obtain

$$\frac{\partial}{\partial t}(\frac{\partial \phi}{\partial x_j})\hat{e}_j + \frac{\partial}{\partial x_j}(u_i N_i)\hat{e}_j = 0 \tag{14}$$

which can be written as

$$\frac{\partial}{\partial t}(N_j)\hat{e}_j + \frac{\partial}{\partial x_j}(u_i N_i)\hat{e}_j = 0 \tag{15}$$

In 2D cartesian coordinates, equation (15) results in the following equations:

$$\frac{\partial N_x}{\partial t} + \frac{\partial}{\partial x}(uN_x + vN_y) = 0 \tag{16}$$

and

$$\frac{\partial N_y}{\partial t} + \frac{\partial}{\partial y}(uN_x + vN_y) = 0 \tag{17}$$

Similar to the VOF and LS methods, the interface curvature at any point is obtained from:

$$\kappa = \nabla \cdot \hat{n}$$

where $\hat{n}$ is the unit normal vector calculated from $\vec{N}$.

Now, consider the following lemma [12]:

Lemma 1 *Let $u_n = \vec{u} \cdot \nabla\phi$ be the normal velocity of each level set, and set $\phi(\vec{x}, 0)$ to be the signed distance function. Then ϕ remains as a signed distance function if and only if $\nabla u_n \cdot \nabla\phi = 0$*

The condition $\nabla u_n \cdot \nabla\phi = 0$ can be also expressed as:

$$\nabla(\vec{u} \cdot \vec{N}) \cdot \vec{N} = 0 \tag{18}$$

where $\vec{N} = \nabla\phi$.

Now, from the advection equation of $\vec{N}$:

$$\frac{\partial \vec{N}}{\partial t} + \nabla(\vec{u} \cdot \vec{N}) = 0 \tag{19}$$

we can obtain:

$$\frac{\partial \vec{N}}{\partial t} \cdot \vec{N} + \nabla(\vec{u} \cdot \vec{N}) \cdot \vec{N} = 0 \tag{20}$$

or

$$\frac{1}{2}\frac{\partial}{\partial t}(|\vec{N}|^2) + \nabla(\vec{u} \cdot \vec{N}) \cdot \vec{N} = 0 \tag{21}$$

So, if the condition in the above lemma is met (equation (18) is satisfied), then we have:

$$\frac{\partial}{\partial t}(|\vec{N}|^2) = 0 \tag{22}$$

and since initially $|\nabla\phi| = |\vec{N}| = 1$, then $\vec{N}$ remains a unit vector (this can be also deduced from the above lemma directly). In other words, if we begin with unit normal vector $\vec{N}$ and solve equation (19) to advect $\vec{N}$ while satisfying equation (18), then $\vec{N}$ remains a unit vector.

Equation (15) is solved to advect the interface normal vector $\vec{N}$. The corresponding numerical methods are detailed next.

Numerical method for solving the equations of the new method

Equation (15) represents an initial value problem. The initial value of $\vec{N}$ defined at each cell corner is specified easily from the initial geometry of an interface.

The scheme which was used to discretize the spatial derivative in equation (15) is a weighted essentially non-oscillatory (WENO) scheme [6] that guarantees third-order accuracy but is up to fifth-order accurate in smooth regions of the flow. However, before taking the spatial derivatives, equation (18) is satisfied.

The temporal derivative of equation (15) is discretized using the third-order, total variation diminishing (TVD) Runge-Kutta method [13].

Results of the new method

To compare the new method with the VOF and level set methods, the same test problem (a 0.1m radius circle, centered at (0.5,0.5) in a 1×1m domain) is considered again. $\vec{N}$ is exactly specified initially. A 2$^{\text{nd}}$-order accurate operator is used to calculate curvatures from $\vec{N}$. The errors associated with the curvatures are presented in Table 6. As one would expect, and unlike the VOF and LS methods, the new method is able to calculate the interface curvature by 2$^{\text{nd}}$-order accuracy.

$R/\Delta x$	l_∞	order	l_1	order
10	0.0427		0.0196	
		2.08		2.03
20	0.0101		0.0048	
		2.07		2.00
40	0.0024		0.0012	
		2.00		2.00
80	0.0006		0.0003	

Table 6. The errors associated with curvatures calculated by the new method, for a 0.1m radius circle at different mesh resolutions. The circle is centered at (0.5,0.5) in a 1×1m domain.

Translation test

The next test problem is the translation of a 0.15m radius circle centered initially at point (0.25,0.5) in a 1×1m domain, where a constant velocity field (1,0) m/s is specified. The circle is advected for 0.5m with CFL number of 0.125. At the end of translation, the errors associated with curvatures and each component of $\vec{N}$ are studied at different mesh resolutions. Tables 7(a) and (b) show the errors in the x and y-components of $\vec{N}$, respectively. $\vec{N}$ is second-order accurate. The errors associated with the curvatures are presented in Table 7(c). The curvature values are converging with almost second-order accuracy.

Rotation test

In another test, a 0.15m radius circle is centered at (0.8,0.5) in a 1×1m domain. An angular velocity ω = 1rad/s is specified about the center of the domain. The circle is advected 2π radians at different mesh resolutions, with a constant CFL = $\pi/50$. Tables 8(a) and (b) show the errors associated with N_x and N_y, respectively, at the end. The errors associated with the curvatures are presented in Table 8(c).

	N_x			
Δx	l_∞	order	l_1	order
1/32	4.34e-2		1.12e-2	
		1.75		1.72
1/64	1.29e-2		3.40e-3	
		1.68		1.50
1/128	4.02e-3		1.20e-3	

(a)

	N_y			
Δx	l_∞	order	l_1	order
1/32	1.04e-1		1.83e-2	
		1.70		1.70
1/64	3.19e-2		5.6e-3	
		1.89		1.71
1/128	8.63e-3		1.71e-3	

(b)

	κ			
Δx	l_∞	order	l_1	order
1/32	3.34		0.527	
		1.83		1.44
1/64	0.94		0.194	
		1.85		1.35
1/128	0.26		0.076	

(c)

Table 7. The errors associated with (a) N_x (the x-component of $\vec{N}$), (b) N_y and (c) curvatures, for a 0.15m radius circle after being translated for 0.5m. The circle is initially centered at (0.25,0.5) in a 1×1m domain and moves at 1m/s in the x-direction. CFL = 0.125.

	N_x			
Δx	l_∞	order	l_1	order
1/32	8.62e-2		2.30e-2	
		2.83		2.84
1/64	1.21e-2		3.21e-3	
		1.96		2.27
1/128	3.11e-3		6.66e-4	

(a)

	N_y			
Δx	l_∞	order	l_1	order
1/32	6.62e-2		2.07e-2	
		2.63		2.79
1/64	1.07e-2		2.98e-3	
		2.38		2.56
1/128	2.05e-3		5.06e-4	

(b)

	κ			
Δx	l_∞	order	l_1	order
1/32	1.867		0.408	
		2.86		2.53
1/64	0.256		0.071	
		0.85		1.24
1/128	0.142		0.030	

(c)

Table 8. The errors associated with (a) N_x, (b) N_y and (c) curvatures, for a 0.15m radius circle after being rotated for 2π radians. The circle is initially centered at (0.8,0.5) in a 1×1m domain and rotates at $\omega = 1$rad/s around the centre of the domain. CFL = $\pi/50$.

Vortex test

This section concludes with a problem that includes shear: the vortex test. In this problem, the following velocity field is specified in a 1×1m domain:

$$u = \sin^2(\pi x)\sin(2\pi y) \tag{23}$$

$$v = -\sin^2(\pi y)\sin(2\pi x) \tag{24}$$

A circle of 0.15m radius is initially placed at (0.5,0.75); see Figure 3. The circle is advected for 1 second and then the sign of the velocity field is reversed and the circle is advected for another 1 second. Ideally, at the end, we would obtain the initial configuration, i.e. the circle. The test was performed at two mesh resolutions: 15 cells per radius (cpr) and 30 cpr, with CFL = 0.1. The results are presented in Figure 3. At 15 cpr, since the circle is sheared until very thin, the tail region is not well resolved. As the interfaces on the two sides come very close to each other, the normal vectors in that region vary sharply across one cell. When the sheared circle moves backward, the tail (with poorly resolved normals) expands. The inaccurate information in this region then leads to the deformed circle at the end. When the resolution is increased to 30 cpr, the recovery of the circle is improved significantly. This implies that for satisfactory performance the new method needs a minimum resolution in areas where the normals vary sharply.

Summary

It has been shown that the accuracy of the curvature values calculated from the Volume-of-Fluid function deteriorates with mesh refinement. As well, there is a constant error associated with the curvatures calculated from the level set function, which does not vanish by decreasing the mesh size. A new method for calculating the interface curvature and normal vectors is proposed in which the interface normal vectors are advected along with the interface. The method is easy to implement, and is second-order accurate in calculating interface normals and curvature.

References

[1] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams. *J. Comput. Phys.*, 213(1):141–173, 2006.

[2] S. Osher and J. A. Sethian. *J. Comput. Phys.*, 79:12–49, 1988.

[3] J. Helmsen, P. Colella, and E. G. Puckett. Non-convex profile evolution in two dimensions using volume of fluids. Technical Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.

[4] J. Y. Poo and N. Ashgriz. *J. Comput. Phys.*, 84(2):483–491, 1989.

[5] Y. Renardy and M. Renardy. *J. Comput. Phys.*, 183(2), 2002.

[6] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Method*. Springer-Verlag, New York, 2003.

[7] M. Sussman, P. Smereka, and S. Osher. *J. Comput. Phys.*, 114:146159, 1994.

[8] M. Sussman, E. Fatemi, P. Smereka, , and S. Osher. *Comput. Fluids.*, 27:663–680, 1998.

[9] G. Son and N. Hur. *Numer. Heat Transfer, Part B*, 42:523–542, 2002.

[10] M. Sussman, Personal communication. 2005.

[11] M. Sussman and E. G. Puckett. *J. Comput. Phys.*, 162:301–337, 2000.

[12] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. *J. Comput. Phys.*, 127(1):179–195, 1996.

[13] S. Gottlieb and C.-W. Shu. *Math. Comput.*, 67(221):73–85, 1998.

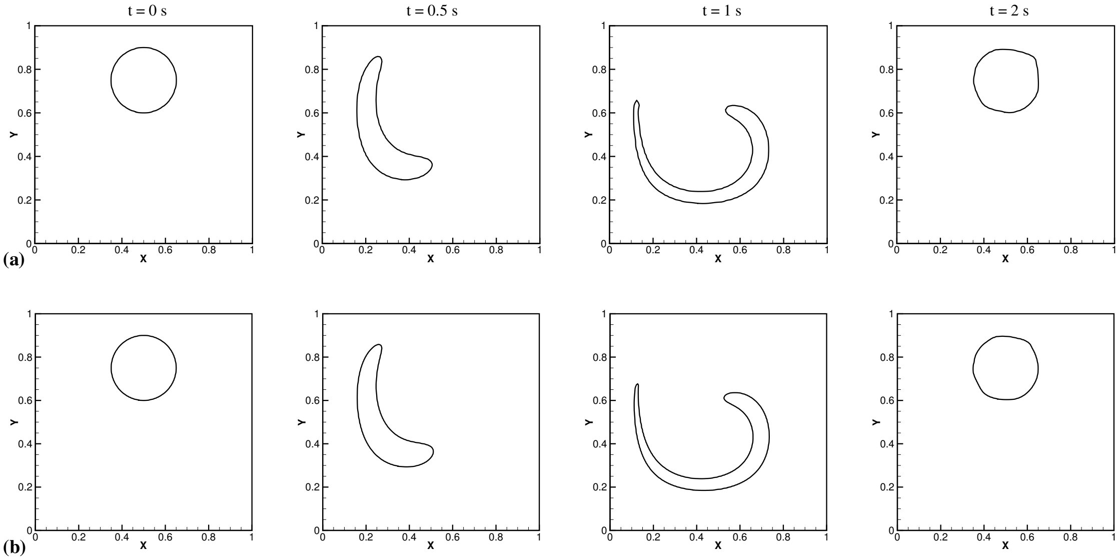


Figure 3. Numerical results of the vortex test for (a) 15cpr resolution and, (b) 30cpr; CFL = 0.1.