Investigating The Stability of The Balance-force Continuum Surface Force Model of Surface Tension In Interfacial Flow

Vinh The Nguyen
University of Massachusetts Dartmouth
Computational Science Training for Undergraduates in the Mathematical Sciences

May 15, 2011

Abstract

Modeling of interfacial flows is very important. However, modeling such flows is difficult. The fluid properties such as density and viscosity can vary sharply across the interface. A popular surface tension model, the balance-force continuum surface force (CSF) model is investigated in this paper. For its stability the model is coupled with different curvature solving methods such as level set (LS), volume of fluid (VOF) and advecting normals (AN).

1 Introduction

Interfacial flow is an important field that is vital in many industrial applications including liquid atomizers, boilers, fuel cells, ink-jet printers, and spray coating and casting processes. Experimental work on such flows is challenging and costly, therefore creating computerized model for these flows is also important. However, modeling of such flows requires solving for the flow field and predicting the shape of fluid interfaces which in term can be very challenging due to characteristics of the fluids vary significantly at the interface. Furthermore, these important characteristics are determined essentially by the evolution of fluid interfaces. As the result, it is important to predict the interfacial geometries accurately. In particular, it is vital to accurately compute interfacial quantities such as curvature and normal vectors because they are used to evaluate the surface tension (Raessi et al. 2010).
The error in computing these quantities then leads to the error in computing surface tension force which is known as the cause of non-physical velocities, commonly known as spurious or parasitic currents. These velocities is observed to grow with time and they can dominate other important physical effects such as buoyancy , and so badly affect the simulation results. The accuracy of the interface curvature is most critical when surface tension is a dominant force (Raessi et al. 2010).
There are two common approaches to modeling the interface kinematics: *interface tracking (Lagrangian) methods and interface capturing (Eulerian) methods (including level set (LS) method and volume of fluid (VOF) method).*

In this paper I focus on using the interface capturing methods to solve for curvature of the interface. In these methods, the location of an interface is implicitly represented by an scalar function on an Eulerian mesh. This make the calculation of surface tension challenging but these methods easily allow the interfaces to merge or rupture (Raessi et al. 2010). The interface is evolved on a fixed numerical mesh by solving the following advection equation:

$$\frac{\delta\chi}{\delta t} + \vec{U}.\nabla\chi = 0 \tag{1}$$

In equation (1) χ is the indicator scalar function (either the LS function or VOF function), and $\vec{U}$ is the velocity. After solving equation (1), the interface unit normal vector $\hat{n}$ is calculated from the spatial derivative of χ as in equation (2), then the curvature is the divergence of $\hat{n}$

$$\hat{n} = \frac{\nabla\chi}{|\nabla\chi|} \tag{2}$$

$$\kappa = -\nabla.\hat{n} \tag{3}$$

As can be seen in equation (2) and (3), calculating unit normal vector $\hat{n}$ and the interface curvature κ from the VOF and LS function are straightforward, although they are not necessarily accurate, approaches. There are multiple approaches that yield better accuracy than LS and VOF and one of them, the advecting normal method, is discussed later in this paper. Those methods are embedded in the continuum surface force (CSF) model and the aim of this paper is to study the stability of this model as different methods of calculating interface curvature are used.
The structure of the paper is as follows. First, the stability of the CSF model is studied when the interface normals and curvatures calculated from the VOF and LS functions. The stability results are then compared to that of using exact curvature (limited case). Then, the AN method is introduced and again, the stability of the model is studied.

2 Mathematical formulations of VOF, LS and AN methods

2.1 VOF method

In the VOF method, the scalar function, f is defined as:

$$f(\vec{x}) = \begin{cases} 1, & \text{if } \vec{x} \in fluid1 \\ 0, & \text{if } \vec{x} \in fluid2 \end{cases} \tag{4}$$

f is the volume fraction and it used to represent fluid 1 in a fluid 1- fluid 2 system. Using this method the interface is tracked via this advection equation:

$$\frac{\partial f}{\partial t} + \vec{u}.\nabla f = 0 \tag{5}$$

The interface unit normal vector is calculated by the gradient of f and its length ∇f as following:

$$\hat{n} = \frac{\nabla f}{|\nabla f|} \tag{6}$$

The curvature, κ is then calculated as:

$$\kappa = -\nabla.(\frac{\nabla f}{|\nabla f|}) \tag{7}$$

This method can be visualized as in figure 1:

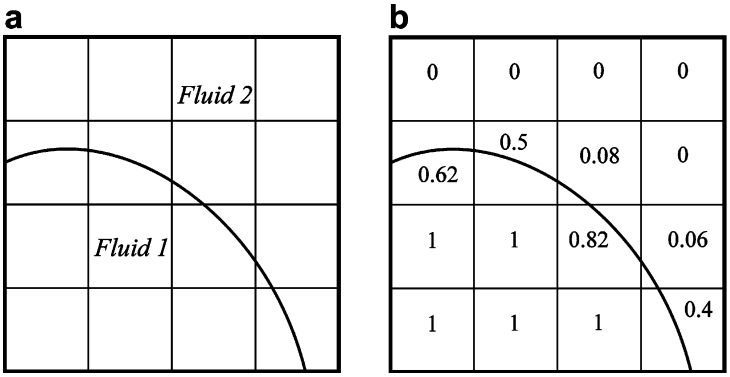

Figure 1: a-VOF 2D interface between fluids 1 and 2, b- the discretize VOF function (Raessi et al. 2006)

For a 2D interface depicted in figure 1a, the discretized VOF function representing fluid 1 and it is observed that the volume fractions vary sharply from zero to one across the interface. This discontinuous behavior makes it difficult to accurately evaluate the first and second derivatives of f, which leads to inaccurate interface normals and curvatures (Raessi et al 2006).

2.2 LS method

In the LS method, the interface is represented by a smooth function ϕ in a domain Ω, which represent fluid 1. This smooth function is defined as the distance to the interface, $\partial\Omega$:

$$|\phi(\vec{x})| = min(|\vec{x} - \vec{x_1}|), \forall x_1 \in \partial\Omega \tag{8}$$

And again this method is still governed by:

$$\frac{\partial \phi}{\partial t} + \vec{u}.\nabla\phi = 0 \tag{9}$$

From equation (8) we can see that $\phi(\vec{x}) = 0$ at when x is at the interface. Then, the smooth function can be represented as:

$$\phi(\vec{x}) = \begin{cases} \geq 0, & \text{if } \vec{x} \in \Omega \\ 0, & \text{if } \vec{x} \in interface\partial\Omega \\ \leq 0, & \text{if } \vec{x} \notin \Omega \end{cases} \tag{10}$$

The unit vector and curvature are calculated the same way as in VOF method:

$$\hat{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{11}$$

$$\kappa = -\nabla.(\frac{\nabla\phi}{|\nabla\phi|}) \tag{12}$$

This LS method can be visualized as in figure 2.

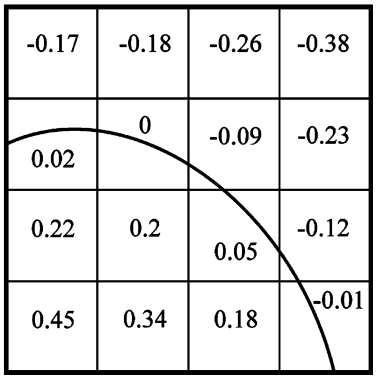

Figure 2: The LS method represent the distance to the interface (Raessi et al 2006)

This method is quite accurate when $\phi = 0$, but when $\phi \neq 0$ this smooth function doesn't necessarily remain the distance function to interface. This error can result in irregular ϕ field that in turn violates conservation of mass.

2.3 Advecting normals

As in previous section we can see that both the VOF and LS is governed by equation (1). The AN method was introduced by Raessi et al. in 2006. In this method, it still considers a smooth functionϕ as in LS method. An extra term, $\vec{N} = \nabla\phi$ is also defined as the normal vector to the contour of ϕ. Equation (9) is then become:

$$\frac{\partial\phi}{\partial t} + \vec{u}.\vec{N} = 0 \tag{13}$$

Taking the gradient of equation (13):

$$\frac{\partial\vec{N}}{\partial t} + \nabla(\vec{u}.\vec{N}) = 0 \tag{14}$$

The difference in this method is instead of taking gradient of the scalar function, the normal vector is advected in the governing equation. After $\vec{N}$ is calculated the curvature is:

$$\kappa = -\nabla.\vec{N} \tag{15}$$

2.4 Errors in computing curvature using VOF,LS and AN:

The error in computing the curvature using the VOF method is shown in table 1. The maximum error l_∞ and the average error l_I grow linearly when the mesh size increases. This is a serious drawback of the VOF method.

The errors associated with curvatures calculated from the VOF function, for a circle of radius 0.15 centered at $(0.5, 0.5)$ in a 1×1 domain, at different mesh resolutions

Δx	l_∞	Order	l_1	Order
1/16	1.09		0.49	
		−0.41		−0.62
1/32	1.44		0.75	
		−1.32		−0.86
1/64	3.59		1.36	
		−1.04		−0.91
1/128	7.38		2.57	
		−1.18		−0.88
1/256	16.75		4.73	
		−1.03		−0.93
1/512	34.14		8.99	
		−1.01		−0.99
1/1024	69.00		17.80	

Table 1: Error in computing curvature using VOF (Raessi et al. 2006)

The LS method otherwise yield much less error than the VOF. The maximum error is about one order less in magnitude comparing to that of VOF. The error is shown in table 2.

The errors associated with curvatures calculated from the LS function ϕ, for a circle of radius 0.15 centered at $(0.5, 0.5)$ in a 1×1 domain, at different mesh resolutions

Δx	l_∞	Order	l_1	Order
1/16	0.5472		0.2963	
		1.55		1.58
1/32	0.1875		0.0991	
		−1.79		−0.51
1/64	0.6481		0.1407	
		0.61		−0.11
1/128	0.4234		0.1518	
		−0.43		−0.02
1/256	0.5689		0.1537	
		−0.29		0.34
1/512	0.6963		0.1215	
		0.11		−0.01
1/1024	0.6453		0.1227	

Table 2: Error in computing curvature using LS (Raessi et al. 2006)

The AN method yields even lower error than the two methods above. It is observed in table 3 that the maximum error and the average error drop dramatically as the the mesh size decreases.

The errors associated with curvatures calculated by the $\vec{N}$ method, for a circle of radius 0.15 centered at (0.5,0.5) in a 1×1 domain, at different mesh resolutions

Δx	l_∞	Order	l_1	Order
1/16	3.87×10^{-1}		2.11×10^{-2}	
		1.72		1.73
1/32	1.18×10^{-1}		6.33×10^{-2}	
		2.11		2.20
1/64	2.73×10^{-2}		1.38×10^{-2}	
		2.07		1.94
1/128	6.51×10^{-3}		3.60×10^{-3}	
		1.95		2.03
1/256	1.68×10^{-3}		8.83×10^{-4}	
		1.97		2.00
1/512	4.29×10^{-4}		2.21×10^{-4}	
		2.02		2.01
1/1024	1.06×10^{-4}		5.50×10^{-5}	

Table 3: Error in computing curvature using AN method (Raessi et al. 2006)

3 Stability Results

The remainder of this paper presents the stability results of tests using the 3 method discussed in previous sections. The results are then compared to that obtained from using exact curvature. However, the case of exact curvature is not practical in reality but it can used as a reference to compared the stability of the methods in computing curvature. The case study in this paper is the 2D static circle test. This case can be visualized as a floating fluid 1 (water) of density $\rho = 1000$ in fluid 2 of the same density. The viscosities of the two fluids are also the same, $\mu = 0.05$, surface tension coefficient, $\sigma = 0.1$

3.1 The Courant-Friedrichs-Lewy condition

The problem discussed in this project is a highly non-linear partial differential equation. To solve this problem also involves solving the Navier-Stokes' Equation. The stability of the solutions depends on the Courant-Friedrichs-Lewy (CFL) condition when the problem is solved numerically. The CFL number is calculated as:

$$CFL = \frac{u.\Delta t}{\Delta x} \tag{16}$$

u is the maximum velocity
Δt is the timestep
Δx is the interval length
For a stable solution, the CFL number has to be less than or equal to one. In this problem we use CFL number at every timestep as a reference point of the appearing of spurious currents. Theoretically computing the CFL number is straight forward but to modify a already exist model to compute the CFL number at very timestep is time consuming.

3.2 Timestep restriction

Another key issue associated with modeling interfacial flow using this model is the timestep restriction. This timestep restriction often becomes more severe than others when surface tension is the dominant force; by requiring a small timestep, it dramatically increases the overall computational time. For numerical stability in the CSF method , the timestep size Δt must satisfy the following condition (Raessi 2008):

$$\Delta t \leq \Delta t_{ST} = \sqrt{\frac{\overline{\rho}(\Delta x)^3}{2\pi\sigma}} \tag{17}$$

$\overline{\rho}$ is the average density of the two fluids.
In my case study the grid size is $\Delta x = \frac{1}{128} = \Delta y$, $\sigma = 0.1$, $\rho = 1000$. The timestep restriction is calculated to be $\Delta t = 0.03$. The aiming of this project is to study how far beyond Δt_{ST} the we can get for timestep size so that the solution remains stable (spurious currents are minimal).

3.3 Stability using exact curvature

As discuss above the exact curvature results are just used as references to emphasize that eliminating error in computing curvature could eliminate the spurious current and the solutions would stay stable at larger timestep restriction.
Since this project is a 2D static drop (static circle test) the exact curvature is:

$$\kappa = \frac{1}{R} \tag{18}$$

The timestep sizes used in for exact curvature were: 0.5, 2, 4, 8 Δt_{ST}. The timestep size is increased by a factor of two. As in the result shown in figure 3 and 4, the solution for exact curvature becomes unstable at 36s when the timestep size is $8\Delta t_{ST}$.

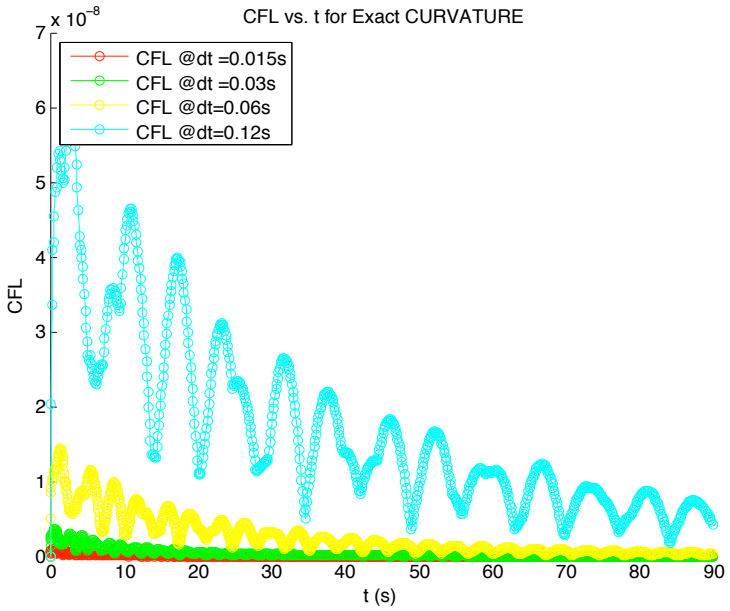

Figure 3: CFL vs time for exact curvature with dt=0.5, 2, 4 Δt_{ST}

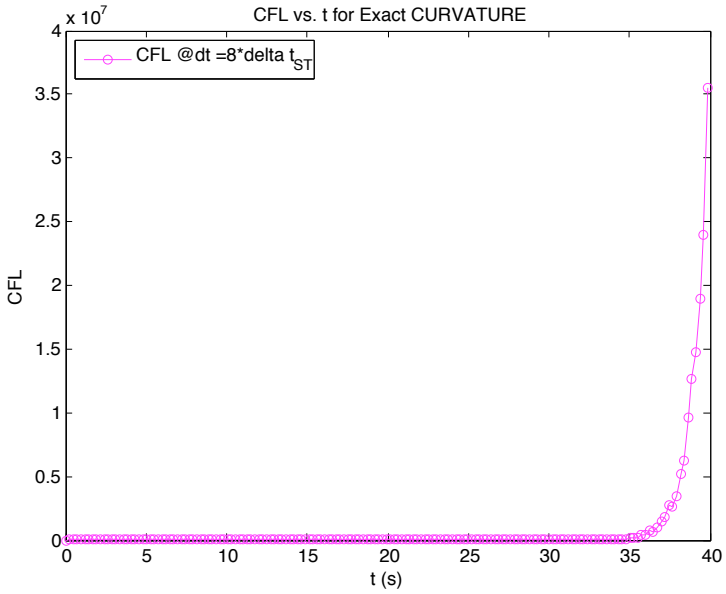

Figure 4: CFL vs. time for exact curvature with dt=8Δt_{ST}

It is possible that the solution might get unstable even of timestep size smaller $8\Delta t_{ST}$. Therefore, refining the timstep size restriction becomes another important task in this project. After multiple trial and error I have found that using timestep size of $6.67\Delta t_{ST}$ the solution is stable, however, at timestep size of $7\Delta t_{ST}$ the solution get unstable at 36s. Again in this project the simulation time is just 90s, so, in order to check if at timestep of $6.67\Delta t_{ST}$ the

solution won't get unstable anytime after 90, I increased the simulation time to 270s and still got the CFL number less than one.

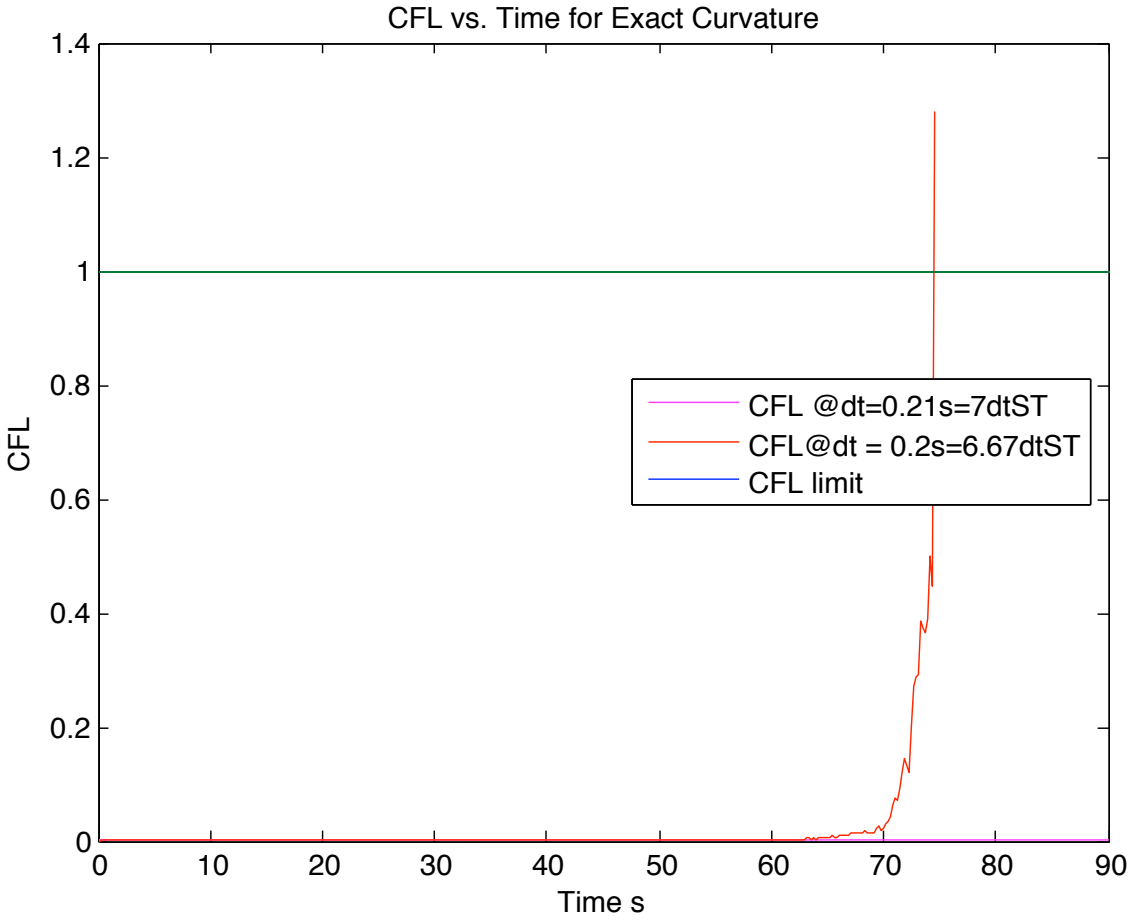

Figure 5: CFL vs time for exact curvature with dt=6.67, 7 Δt_{ST}

3.4 Stability using LS to compute curvature

The simulation time for this problem is initially 90s. In this section the LS method is used to compute the curvature of the drop. The timestep sizes are chosen to be $0.5\Delta t_{ST}$, Δt_{ST}, $2\Delta t_{ST}$, $4\Delta t_{ST}$. The simulation is top as soon as the CFL number exported to screen is greater than one.
It is observed in figures 6 and 7 that using LS method to compute curvature, the CSF solution is still stable when the timestep size is $0.5\Delta t_{ST}$, Δt_{ST}, $2\Delta t_{ST}$,$4\Delta t_{ST}$.
When the timestep size is increased to $4\Delta t_{ST}$ grows dramatically after 1s. The next task is to find exactly what the maximum timestep size that can be used above which the solution is unstable.

Using the same approach as for exact curvature, I have also found that using timestep size

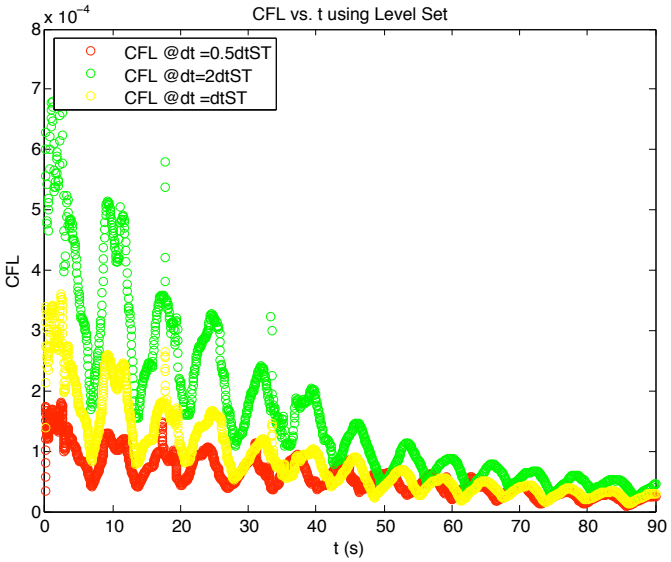

Figure 6: CFL vs. time for dt = 0.5,1,2 Δt_{ST}

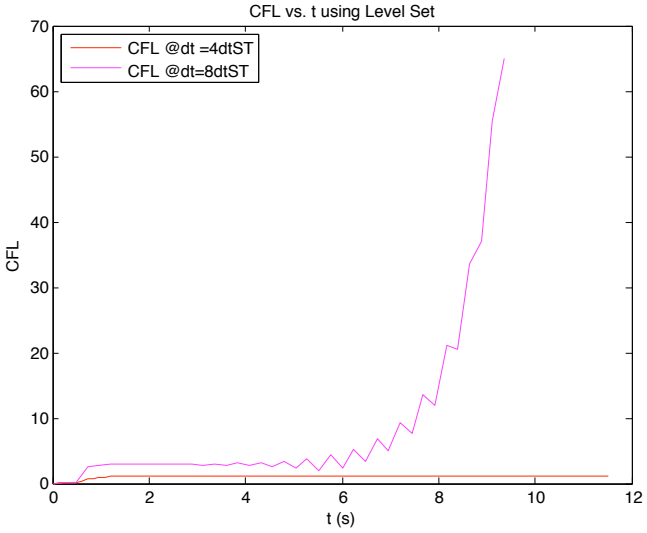

Figure 7: CFL vs. time for dt=4,8Δt_{ST}

of $3.167\Delta t_{ST}$ the solution is stable, however, at timestep size of $3.2\Delta t_{ST}$ the solution get unstable at 36s. Again, when using timestep size of $3.167\Delta t_{ST}$ I also increased the simulation time to 270s to make sure the solution still stable and I still got the CFL number less than one throughout the whole simulation.

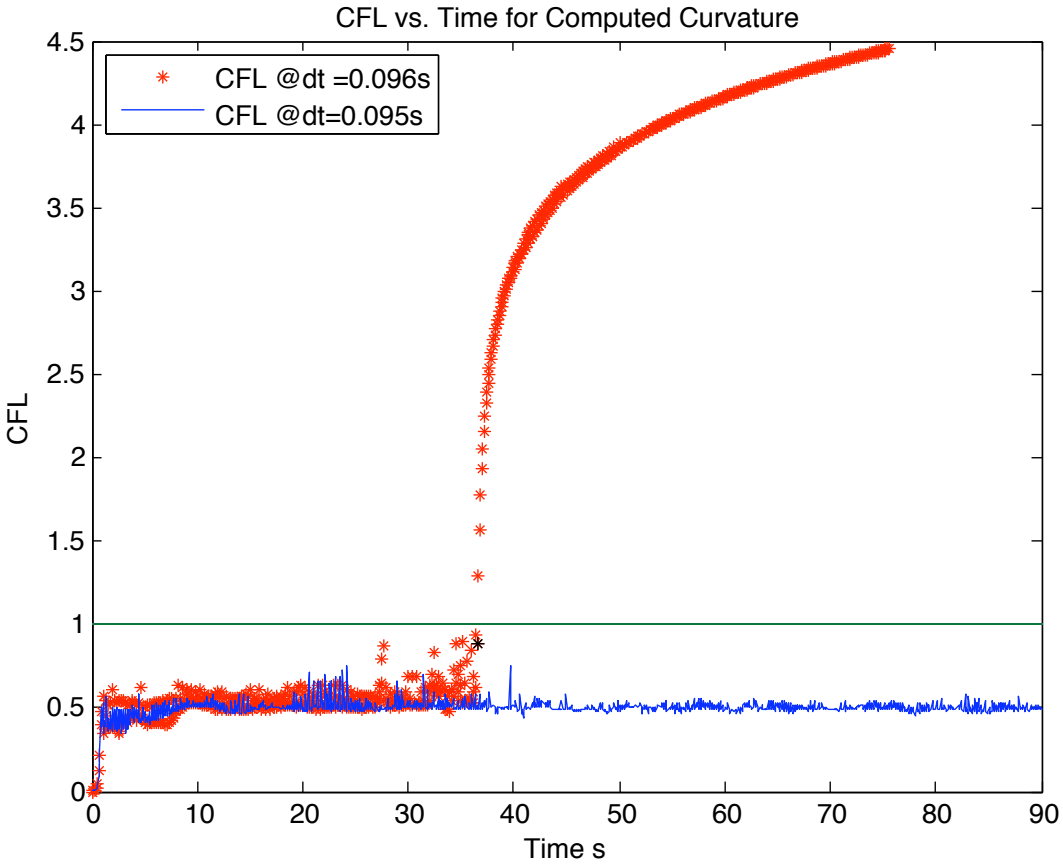

Figure 8: CFL vs. time for dt = 3.167, and 3.2 Δt_{ST}

3.5 Stability using VOF to compute curvature

As discussed before, VOF is not a very accurate technique to calculate curvature. The error in computing curvature is higher than that of LS method and AN method, therefore, one might thought that the solution might get unstable at smaller timestep size than other more robust method. However the result I obtained is totally contradicted to that. The plot in figure 9 shows that the CFL magnitude still stays below one when I applied the same timestep size at which the LS method gets unstable. The final timestep size that the VOF method gets unstable is found to exactly $4\Delta t_{ST}$. From these results, I could conclude in my case study the VOF method stays stable at larger timestep restriction than the LS method. This interesting result could be explained using the definitions of the LS method and the VOF method. Although the LS method is more accurate in computing curvature than the VOF method, each layer of the mesh heavily depends on the one another of which small error could be carried out and magnified throughout the whole simulation. In contrast, the VOF method is just a volume fraction function and it mainly focus on the interface, so if there is some error, this error wouldn't be magnified or or greatly affect the whole solution.

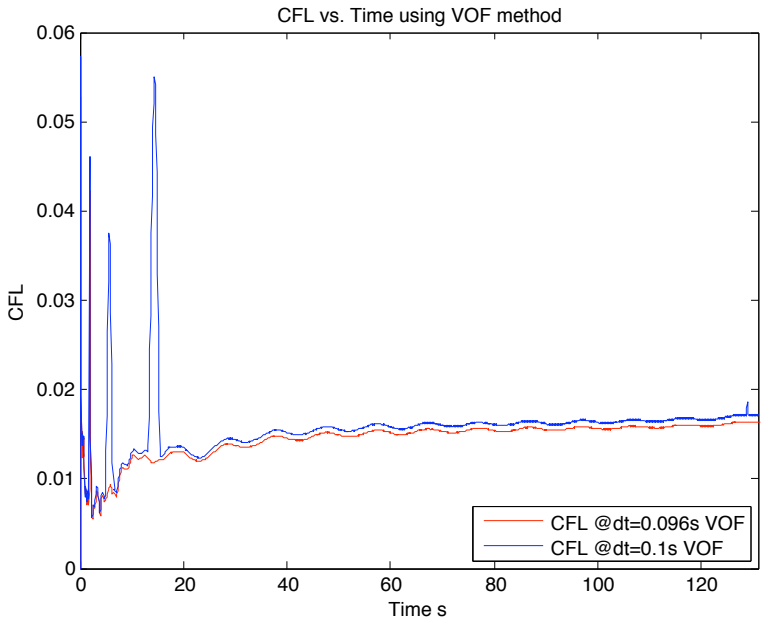

Figure 9: CFL vs time using VOF method to calculate curvature at timestep size $3.2\Delta t_{ST}$ and $3.33\Delta t_{ST}$

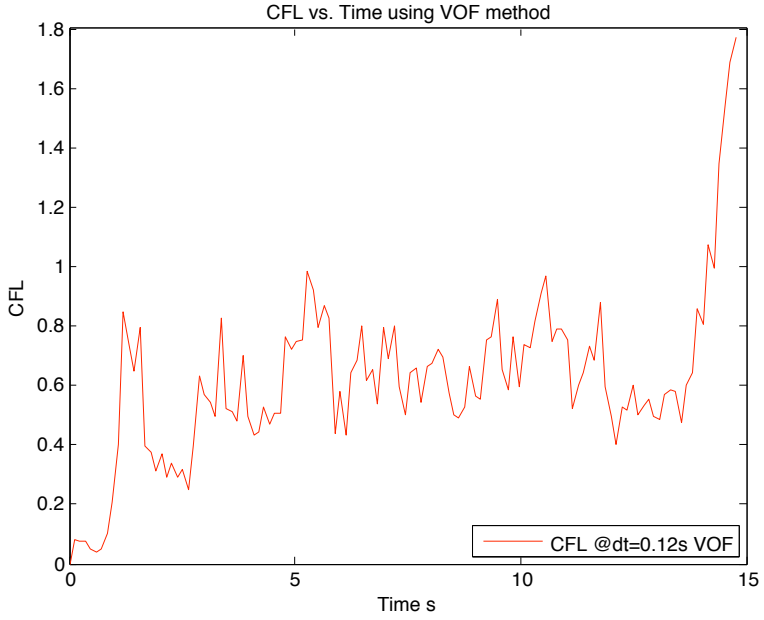

Figure 10: CFL vs time using VOF method to calculate curvature at timestep size $4\Delta t_{ST}$

3.6 Stability using AN to compute curvature

Following every step as in the others method I have found that using AN to computing curvature the solution got unstable at the timestep size of $3.1.67\Delta t_{ST}$ which is the smaller comparing to that of LS method. Also, using AN to compute curvature the solution get

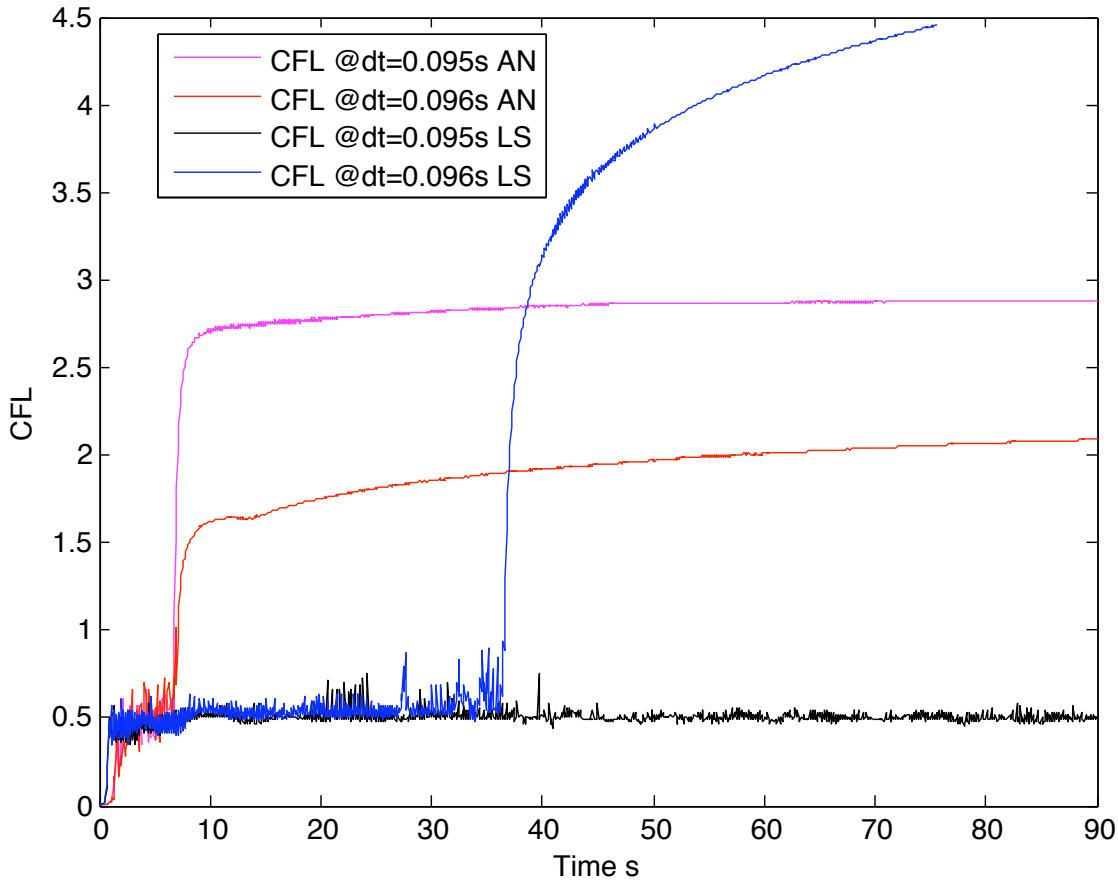

Figure 11: Comparing LS vs AN's CFL vs time

unstable quicker than using LS at the same timestep size of $3.2\Delta t_{ST}$. The result can be seen as below in figure 12. This phenomenon is very interesting because as proposed in Raessi et al. 2006, AN method is more accurate than LS and yet its less stable.
The explaination for this phenomenon is that because the normal vector is advected inside the PDE, even small error in computing unit normal vector would create large error in surface tension resulting in large pressure drop arcross the interface (see Apendix). Also in Raessi et al. 2006, this AN technique is more accurate in a flow problem where the fluid velocity dominates.

4 Conclusion

Overall, using different methods of calculating curvature yield different stability of the solution. Based on these results I found in this project, the curvature play an important role in the stability, therefore, the more accurate the curvature is computed, the more stable the solution is. The more stable the solution in turn increases the timestep restriction which will then shorten the computational cost.

5 Acknowledgement

I would like to thank you Dr. Mehdi Raessi whose encouragement, guidance and support from the initial to the final level enabled us to develop an understanding of the subject. Lastly, I offer my best regards Dr. Gottlieb, Dr. Kim and other CSUMS staff members who supported me in any mean during the completion of the project.

Vinh The Nguyen
University of Massachusetts-Dartmouth
May 15th, 2011

References

[1] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams. *" A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework."* **Journal of Computational Physics, 213:141–173, 2006**

[2] Mehdi Raessi, Javad Mostaghimi, Markus Bussmann *"A volume-of-fluid interfacial flow solver with advected normals"* **2010**

[3] Mehdi Raessi *"On modeling surface tension-dominan, large density ratio, two-phase flows"* **2008**,

[4] M. Raessi, J. Mostaghimi , M. Bussmann *"Advecting normal vectors: A new method for calculating interface normals and curvatures when modeling two-phase ows"* **2007**