

# Volume Conservation in the Level Set Method

Dmitriy Kats

March 13, 2017

## Contents

|                   |                                      |           |
|-------------------|--------------------------------------|-----------|
| <b>1</b>          | <b>Problem Definition</b>            | <b>2</b>  |
| <b>2</b>          | <b>Finite Element Discretization</b> | <b>2</b>  |
| <b>3</b>          | <b>Finite Volume Discretization</b>  | <b>3</b>  |
| <b>4</b>          | <b>Fast Marching Method</b>          | <b>4</b>  |
| <b>5</b>          | <b>Test Case 1</b>                   | <b>5</b>  |
| <b>6</b>          | <b>Test Case 2</b>                   | <b>5</b>  |
| <b>7</b>          | <b>Accuracy of reinitialization</b>  | <b>5</b>  |
| <b>8</b>          | <b>FEM Results for Test Case 1</b>   | <b>7</b>  |
| <b>9</b>          | <b>FEM Results for Test Case 2</b>   | <b>8</b>  |
| <b>10</b>         | <b>FVM Results for Test Case 1</b>   | <b>11</b> |
| <b>11</b>         | <b>FVM Results for Test Case 2</b>   | <b>16</b> |
| <b>12</b>         | <b>Conclusion</b>                    | <b>19</b> |
| <b>References</b> |                                      | <b>19</b> |

# 1 Problem Definition

The level set method is a good choice to capture an interface. In this report, I will investigate how conservative the level set method is for two different test cases where the theoretical solution is known. I evolved the level set in a known velocity field. Test case 1 featured twisting with large topological changes, while test case 2 was a translation problem. For both test cases, the simulation is run until the bubble returns to its original position ideally with no volume change. I will evaluate how close this ideal is reached.

## 2 Finite Element Discretization

The level set evolution equation is:

$$\dot{\phi}(t, \mathbf{x}) + \mathbf{V}(\mathbf{x}) \cdot \nabla \phi(t, \mathbf{x}) = 0 \quad (1)$$

The weak form of this is found by multiplying by a test function,  $w$ , and integrating in the domain:

$$\int_{\Omega} w \dot{\phi}(t, \mathbf{x}) d\Omega + \int_{\Omega} w \mathbf{V}(\mathbf{x}) \cdot \nabla \phi(t, \mathbf{x}) d\Omega = 0 \quad (2)$$

The solution,  $\phi^h(t, \mathbf{x})$ , that satisfies the weak form is approximated as:

$$\phi^h(t, \mathbf{x}) = \sum_I N_I(\mathbf{x}) \phi_I(t) \quad (3)$$

The test function can be approximated in the same way:

$$w^h(\mathbf{x}) = \sum_I N_I(\mathbf{x}) w_I(t) \quad (4)$$

The approximations are substituted into the weak form to produce the following:

$$\int_{\Omega} N_I N_J d\Omega \dot{\phi}_J + \int_{\Omega} N_I u_j \frac{dN_J}{dx_j} d\Omega \phi_J = 0 \quad (5)$$

For convenience, I adopted the following notation:

$$M_{IJ} = \int_{\Omega} N_I N_J d\Omega, \quad C_{IJ} = \int_{\Omega} N_I u_j \frac{dN_J}{dx_j} d\Omega$$

An explicit scheme was used with a lumped mass matrix. The lumped mass matrix,  $M_{IJ}^l$ , is found using a row-summed technique. The values of  $\phi$  are updated then as:

$$\phi^{n+1} = \phi^n - \Delta t (\mathbf{M}^l)^{-1} \mathbf{C} \phi^n \quad (6)$$

The weak form (used in this study for stability purposes) is found by multiplying by a different test function,  $\hat{w} = w + w^*$ , and integrating in the domain:

$$\int_{\Omega} (w + w^*) \dot{\phi}(t, \mathbf{x}) d\Omega + \int_{\Omega} (w + w^*) \mathbf{V}(\mathbf{x}) \cdot \nabla \phi(t, \mathbf{x}) d\Omega = 0 \quad (7)$$

The equation of  $w^*$  is  $w^* = \frac{h\alpha}{2|\mathbf{V}|} \mathbf{V} \cdot \nabla N_I$ . There is no diffusion so  $\alpha \rightarrow 1$ .

After the substitution of the updated test function and shape functions:

$$\left( M_{IJ} + \int_{\Omega} \left( \frac{h\alpha}{2|\mathbf{V}|} u_i \frac{dN_I}{dx_i} N_J \right) d\Omega \right) \dot{\phi}_J + \left( C_{IJ} + \int_{\Omega} \left( \frac{h\alpha}{2|\mathbf{V}|} u_i \frac{dN_I}{dx_i} \right) u_j \frac{dN_J}{dx_j} d\Omega \right) \phi_J = 0 \quad (8)$$

For convenience, I used the following notation:

$$\hat{M}_{IJ} = \left( M_{IJ} + \int_{\Omega} \left( \frac{h\alpha}{2|\mathbf{V}|} u_i \frac{dN_I}{dx_i} N_J \right) d\Omega \right) \text{ and } \hat{C}_{IJ} = \left( C_{IJ} + \int_{\Omega} \left( \frac{h\alpha}{2|\mathbf{V}|} u_i \frac{dN_I}{dx_i} \right) u_j \frac{dN_J}{dx_j} \right)$$

The explicit scheme (using a lumped mass) with damping is then:

$$\phi^{n+1} = \phi^n - \Delta t (\hat{\mathbf{M}}^l)^{-1} \hat{\mathbf{C}} \phi^n \quad (9)$$

### 3 Finite Volume Discretization

I took this discretization from [1]. The level set evolution equation is:

$$\partial_t \phi(t, \mathbf{x}) + \mathbf{V}(\mathbf{x}) \cdot \nabla \phi(t, \mathbf{x}) = \mathbf{0} \quad (10)$$

It can be rewritten (for the 2D case) as:

$$\partial_t \phi(t, x, y) + \nabla \cdot (\phi(t, x, y) \mathbf{V}) = \phi(t, \mathbf{x}, \mathbf{y}) \nabla \cdot \mathbf{V} \quad (11)$$

Using the finite volume technique this equation is integrated over a control volume  $\Omega_{ij}$ :

The integrated equation is:

$$\begin{aligned} \phi_{i,j}^{n+1} &= \phi_{i,j}^n - \frac{\Delta t}{h} \left( u_{i+1/2,j} (\phi_{i+1/2,j}^{n+1/2} - \phi_{i,j}^{n+1/2}) + v_{i,j+1/2} (\phi_{i,j+1/2}^{n+1/2} - \phi_{i,j}^{n+1/2}) \right. \\ &\quad \left. + \frac{\Delta t}{h} \left( u_{i-1/2,j} (\phi_{i-1/2,j}^{n+1/2} - \phi_{i,j}^{n+1/2}) + v_{i,j-1/2} (\phi_{i,j-1/2}^{n+1/2} - \phi_{i,j}^{n+1/2}) \right) \right) \end{aligned} \quad (12)$$

$$\phi_{i+1/2,j}^{n+1/2} = \begin{cases} \phi_{(i)+1/2,j}^{n+1/2} & u_{i+1/2,j} > 0 \\ \phi_{(i+1)-1/2,j}^{n+1/2} & u_{i+1/2,j} < 0 \end{cases}$$

$$\phi_{i-1/2,j}^{n+1/2} = \begin{cases} \phi_{(i-1)+1/2,j}^{n+1/2} & u_{i-1/2,j} > 0 \\ \phi_{(i)-1/2,j}^{n+1/2} & u_{i-1/2,j} < 0 \end{cases}$$

$$\phi_{i,j+1/2}^{n+1/2} = \begin{cases} \phi_{i,(j)+1/2}^{n+1/2} & v_{i,j+1/2} > 0 \\ \phi_{i,(j+1)-1/2}^{n+1/2} & v_{i,j+1/2} < 0 \end{cases}$$

$$\phi_{i,j-1/2}^{n+1/2} = \begin{cases} \phi_{i,(j-1)+1/2}^{n+1/2} & v_{i,j-1/2} > 0 \\ \phi_{i,(j)-1/2}^{n+1/2} & v_{i,j-1/2} < 0 \end{cases}$$

Taking Taylor expansions:

$$\begin{aligned}\phi_{i,j}^{n+1/2} &= \phi_{i,j}^n - \frac{\Delta t}{2} \mathbf{V}_{i,j} \cdot \nabla \phi_{i,j}^n \\ \phi_{(i)+1/2,j}^{n+1/2} &= \phi_{i,j}^n + \frac{h}{2} \partial_x \phi_{i,j}^n - \frac{\Delta t}{2} \mathbf{V}_{i,j} \cdot \nabla \phi_{i,j}^n \\ \phi_{(i)-1/2,j}^{n+1/2} &= \phi_{i,j}^n - \frac{h}{2} \partial_x \phi_{i,j}^n - \frac{\Delta t}{2} \mathbf{V}_{i,j} \cdot \nabla \phi_{i,j}^n \\ \phi_{i,(j)+1/2}^{n+1/2} &= \phi_{i,j}^n + \frac{h}{2} \partial_y \phi_{i,j}^n - \frac{\Delta t}{2} \mathbf{V}_{i,j} \cdot \nabla \phi_{i,j}^n \\ \phi_{i,(j)-1/2}^{n+1/2} &= \phi_{i,j}^n - \frac{h}{2} \partial_y \phi_{i,j}^n - \frac{\Delta t}{2} \mathbf{V}_{i,j} \cdot \nabla \phi_{i,j}^n\end{aligned}$$

Variations of the above are:

$$\begin{aligned}\phi_{(i+1)-1/2,j}^{n+1/2} &= \phi_{i+1,j}^n - \frac{h}{2} \partial_x \phi_{i+1,j}^n - \frac{\Delta t}{2} \mathbf{V}_{i+1,j} \cdot \nabla \phi_{i+1,j}^n \\ \phi_{(i-1)+1/2,j}^{n+1/2} &= \phi_{i-1,j}^n + \frac{h}{2} \partial_x \phi_{i-1,j}^n - \frac{\Delta t}{2} \mathbf{V}_{i-1,j} \cdot \nabla \phi_{i-1,j}^n \\ \phi_{i,(j+1)-1/2}^{n+1/2} &= \phi_{i,j+1}^n - \frac{h}{2} \partial_y \phi_{i,j+1}^n - \frac{\Delta t}{2} \mathbf{V}_{i,j+1} \cdot \nabla \phi_{i,j+1}^n \\ \phi_{i,(j-1)+1/2}^{n+1/2} &= \phi_{i,j-1}^n + \frac{h}{2} \partial_y \phi_{i,j-1}^n - \frac{\Delta t}{2} \mathbf{V}_{i,j-1} \cdot \nabla \phi_{i,j-1}^n\end{aligned}$$

The gradient is approximated as:

$$\begin{aligned}\nabla \phi_{i,j}^n &\approx \nabla \phi(t^n, x_i, y_j) = \begin{pmatrix} \partial_x \phi_{i,j}^n \\ \partial_y \phi_{i,j}^n \end{pmatrix} = \begin{pmatrix} \kappa(u_{i,j}) \frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{h} + (1 - \kappa(u_{i,j})) \frac{\phi_{i,j}^n - \phi_{i-1,j}^n}{h} \\ \kappa(v_{i,j}) \frac{\phi_{i,j+1}^n - \phi_{i,j}^n}{h} + (1 - \kappa(v_{i,j})) \frac{\phi_{i,j}^n - \phi_{i,j-1}^n}{h} \end{pmatrix} \\ \nabla \phi_{i+1,j}^n &\approx \nabla \phi(t^n, x_{i+1}, y_j) = \begin{pmatrix} \partial_x \phi_{i+1,j}^n \\ \partial_y \phi_{i+1,j}^n \end{pmatrix} = \begin{pmatrix} \kappa(u_{i+1,j}) \frac{\phi_{i+2,j}^n - \phi_{i+1,j}^n}{h} + (1 - \kappa(u_{i+1,j})) \frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{h} \\ \kappa(v_{i+1,j}) \frac{\phi_{i+1,j+1}^n - \phi_{i+1,j}^n}{h} + (1 - \kappa(v_{i+1,j})) \frac{\phi_{i+1,j}^n - \phi_{i+1,j-1}^n}{h} \end{pmatrix} \\ \nabla \phi_{i,j+1}^n &\approx \nabla \phi(t^n, x_i, y_{j+1}) = \begin{pmatrix} \partial_x \phi_{i,j+1}^n \\ \partial_y \phi_{i,j+1}^n \end{pmatrix} = \begin{pmatrix} \kappa(u_{i,j+1}) \frac{\phi_{i+1,j+1}^n - \phi_{i,j+1}^n}{h} + (1 - \kappa(u_{i,j+1})) \frac{\phi_{i,j+1}^n - \phi_{i-1,j+1}^n}{h} \\ \kappa(v_{i,j+1}) \frac{\phi_{i,j+2}^n - \phi_{i,j+1}^n}{h} + (1 - \kappa(v_{i,j+1})) \frac{\phi_{i,j+1}^n - \phi_{i,j}^n}{h} \end{pmatrix}\end{aligned}$$

The boundary conditions are no flux boundary conditions.

## 4 Fast Marching Method

The fast marching method is implemented using the equations in [2]. Specifically equation (2.2) is used to propagate  $\phi$  and the bicubic interpolation detailed in section (3.2) in Chopp's paper was used. As noted in the article, the method is only first order accurate and locally second order accurate around the interface due to the bicubic interpolation.

## 5 Test Case 1

The first test case had a time dependent velocity and it was run until  $t = 8$ . The velocity was taken from [1]:

$$\mathbf{V}(t, x, y) = 2\cos\left(\frac{\pi t}{8}\right)(u, v) \quad (13)$$

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

The bubble of  $r = 0.15$  was placed at  $(x, y) = (0.5, 0.75)$  and the simulation was run until  $t = 8$ . Max deformation is reached at  $t = 4$  and it should return to its original position at  $t = 8$ . Essentially it is a twisting until  $t = 4$  followed by the reversal twist.

## 6 Test Case 2

The second test case had a time dependent velocity and it was run until  $t = 2$ . The velocity was:

$$\mathbf{V}(t, x, y) = \begin{cases} (0.5, 0.5) & 0 \leq t \leq 1 \\ (-0.5, -0.5) & 1 < t \leq 2 \end{cases} \quad (14)$$

The bubble of  $r = 0.15$  was placed at  $(x, y) = (0.25, 0.25)$  and the simulation was run until  $t = 2$ . It is furthest away at  $t = 1$  and then it slides back to the original position.

## 7 Accuracy of reinitialization

The reinitialization was checked by doing 20 repetitive reinitializations. The input is a  $\phi$  that is exactly a signed distance function. A circle was used for the test case because it is easy to construct the signed distance function. The error is visually visible in Figure 1. The  $L_2 E$  (E for error) for different mesh sizes is shown in Figure 2. The definition used is:

$$L_2 E_{reinitialization} = \frac{\sqrt{\int_{\Omega} (\phi^{th} - \phi^R)^2 d\Omega}}{\sqrt{\int_{\Omega} (\phi^{th})^2 d\Omega}} \quad (15)$$

where  $\phi^R$  is the reinitialized version of  $\phi^{th}$  (or theoretical  $\phi$ ). The Fast Marching Method implemented was first order accurate so these results are expected (the slope is close to 1).

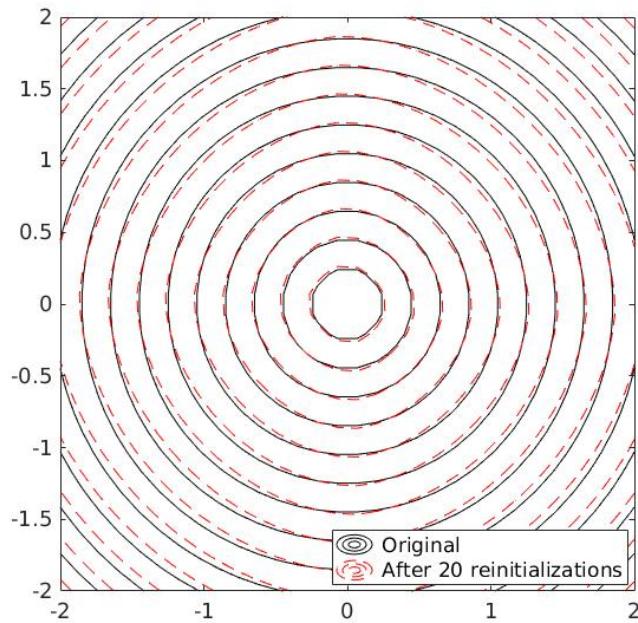


Figure 1: 20 Reinitializations of a Circle of  $r = 0.25$

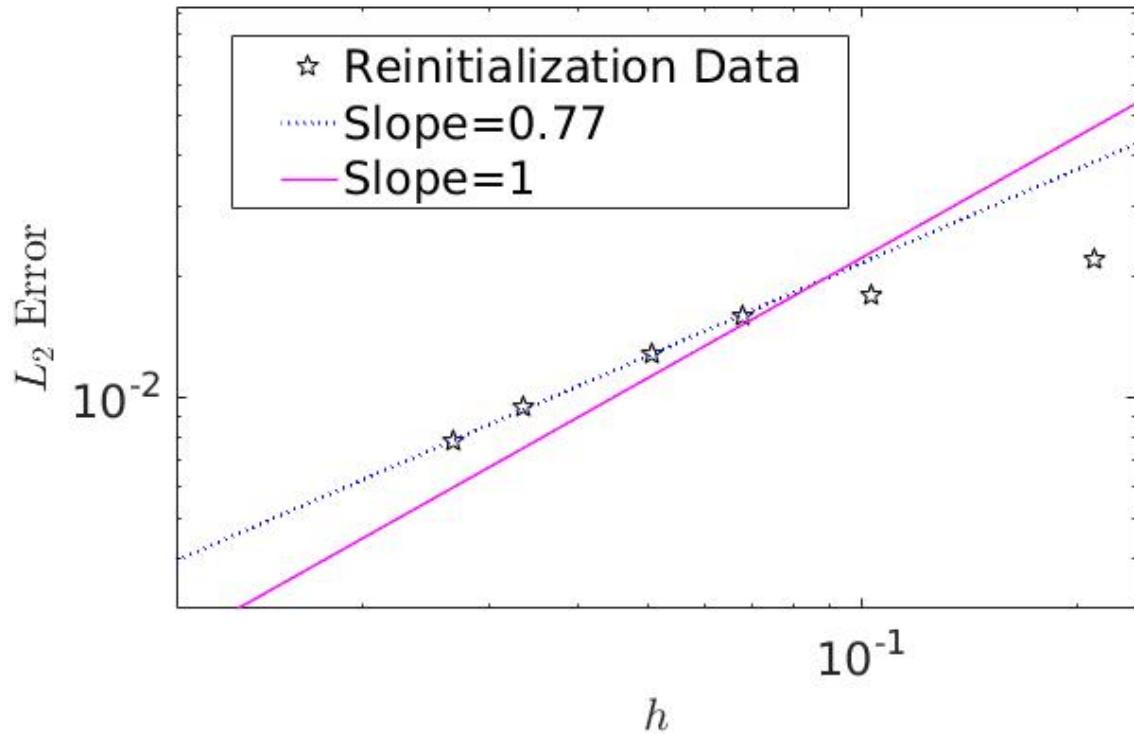


Figure 2:  $L_2$  error of 20 reinitializations

## 8 FEM Results for Test Case 1

The bubble mass was completely lost as seen in Figure 3. The discretization was not conservative so these results are expected. The simulations also took a long time because all of the matrices except the mass matrix were dependent on the velocity so they had to be recomputed at every time step. Due to how long these simulations took and the lack of accuracy in the discretization I only ran two simulations for 50x50 and 100x100 elements. For both cases the bubble was lost, so I decided to move on to the simpler Test Case 2 to evaluate the finite element discretization.

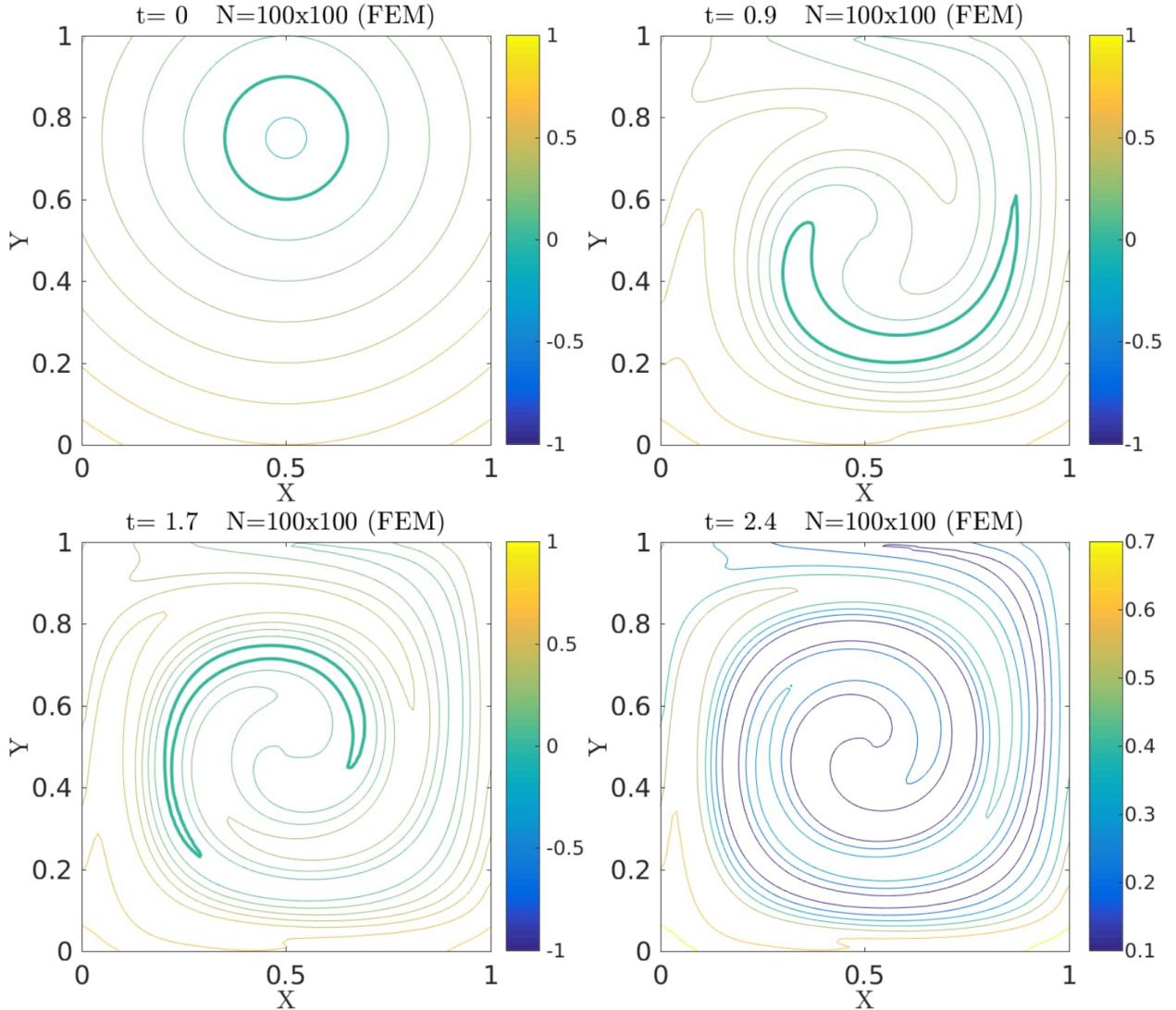


Figure 3: FEM Test Case 1

## 9 FEM Results for Test Case 2

The bubble shrunk significantly as seen in Figure 4. Reinitialization helps in not losing as much volume but it adds a strange volume lump as seen in Figure 5 for the 50x50 mesh when reinitialization is done at every time step. However, if reinitialization is done every two time steps for a 100x100 mesh then the volume is conserved better as seen in Figure 6. The unexpected volume lump is probably due to the reinitialization being first order accurate.

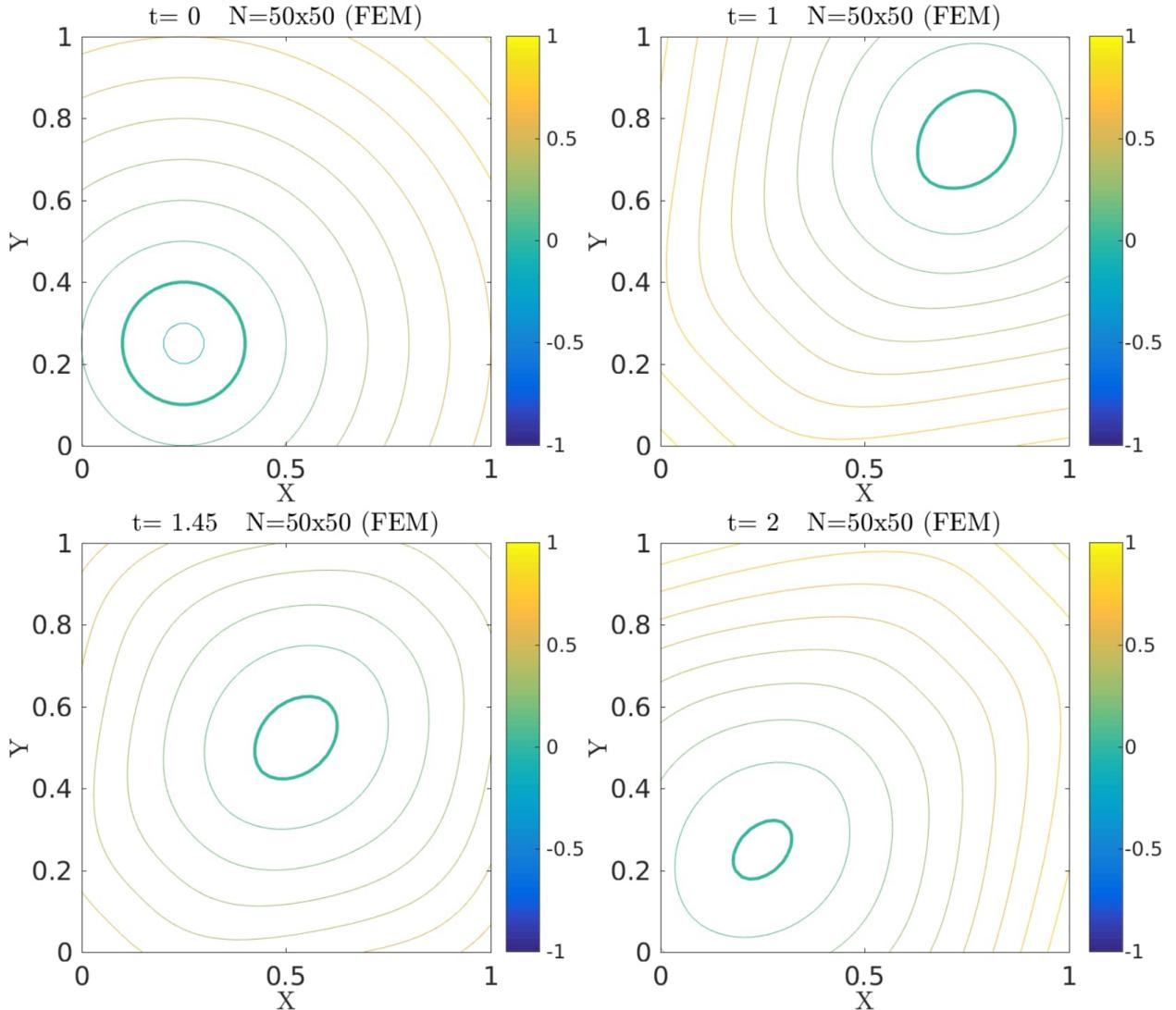


Figure 4: FEM Test Case 2

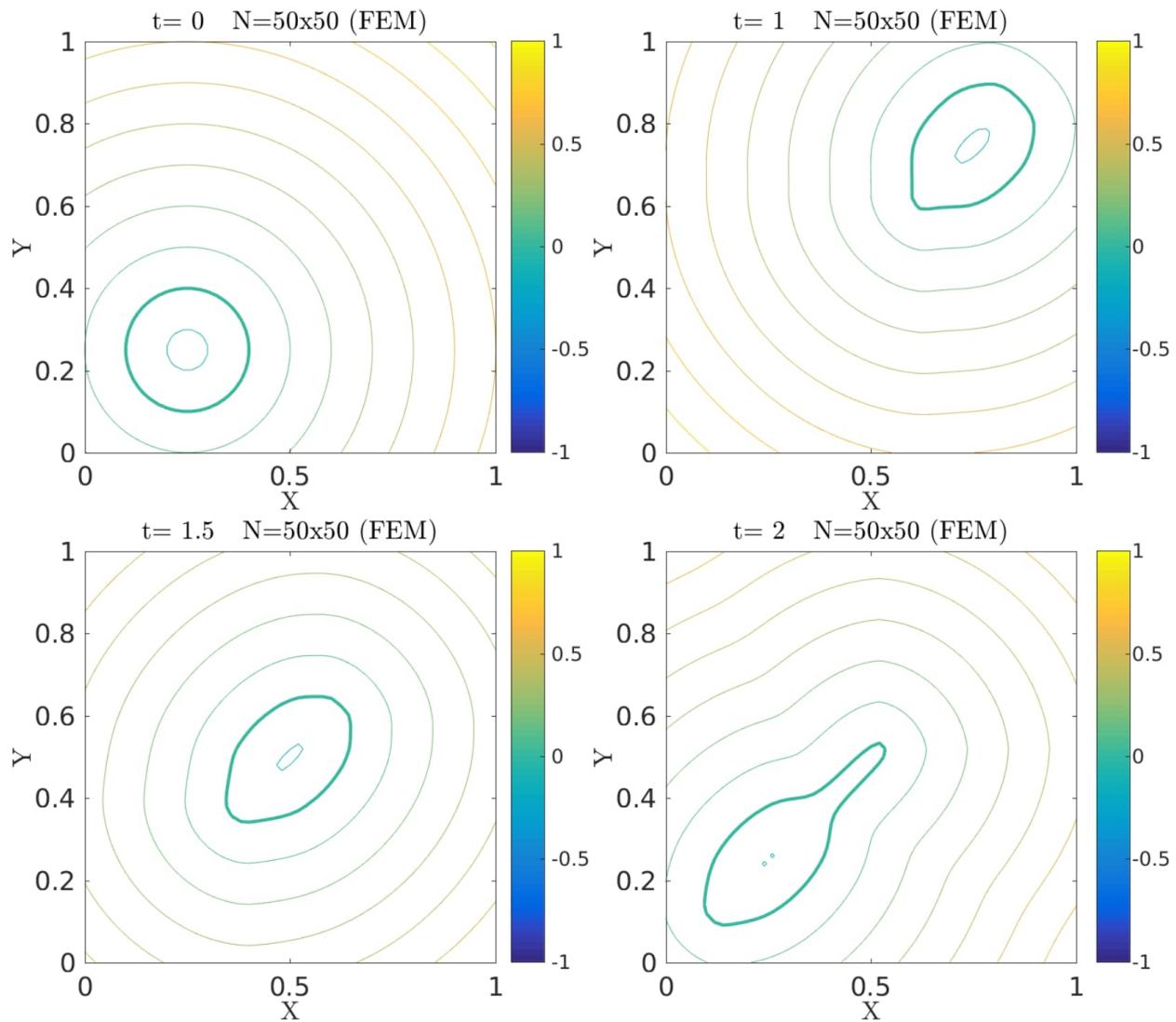


Figure 5: FEM Test Case 2 with reinitialization

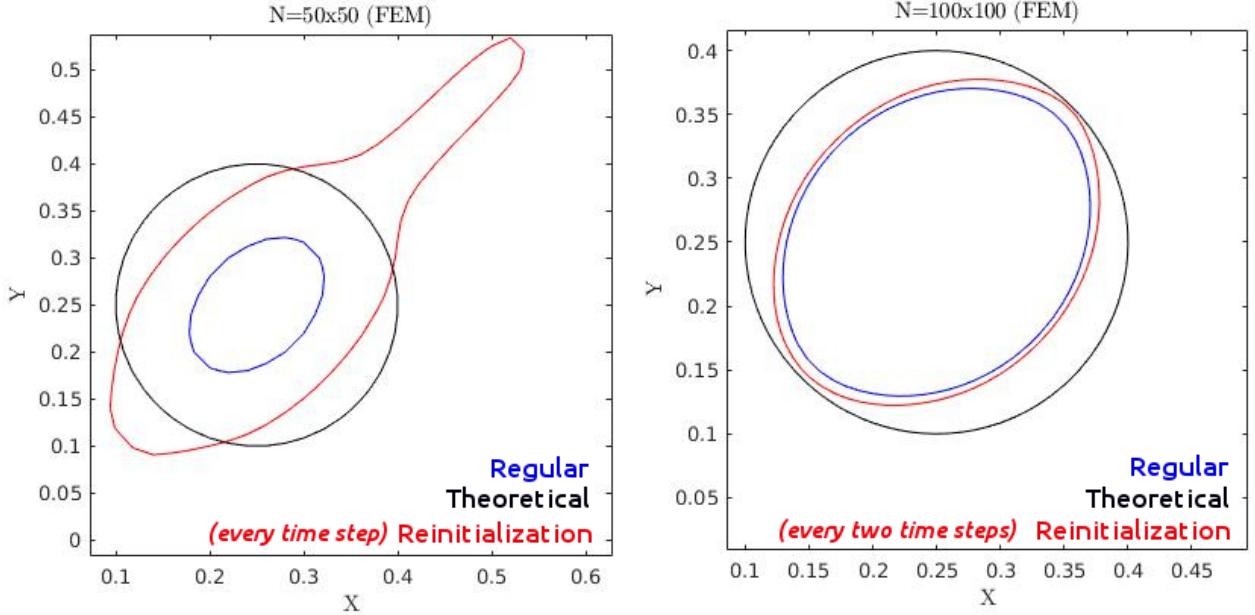


Figure 6: FEM Test Case 2 comparison for different reinitializations

The error for the volume conservation (for this 2D problem) is:

$$L_2 E_{volume} = \frac{|A^{th} - A^n|}{|A^{th}|} \quad (16)$$

where  $A^{th}$  is the theoretical area, while  $A^n$  is the FVM area at the end of the test.

In order to get the area, I took the grid points from the zero contour and then just used the matlab function "polyarea" to get the approximate area. I was mostly looking for the trend in volume conservation for different number of elements. A more accurate procedure is to use a bicubic interpolation to extract the polynomial in each voxel (and to get a better approximation of the interface location). This way the area would be second order accurate. Regardless, I found that the FEM discretization is a bit better than first order accurate based on the slope of the error plot in Figure 7. This is expected because the SUPG term is upwinding and first order accurate.

Reinitialization changes the volume conservation. In Figure 7 the area of the bubble (on the left) with reinitialization at every time step was 0.07456 which is actually greater than the area of the original bubble (0.07069). Of course, the bubble on the right with reinitialization every two time steps holds volume conservation better since it has more volume than the one without reinitialization. The addition of volume in the level method due to reinitialization is a known phenomenon as noted in [3].

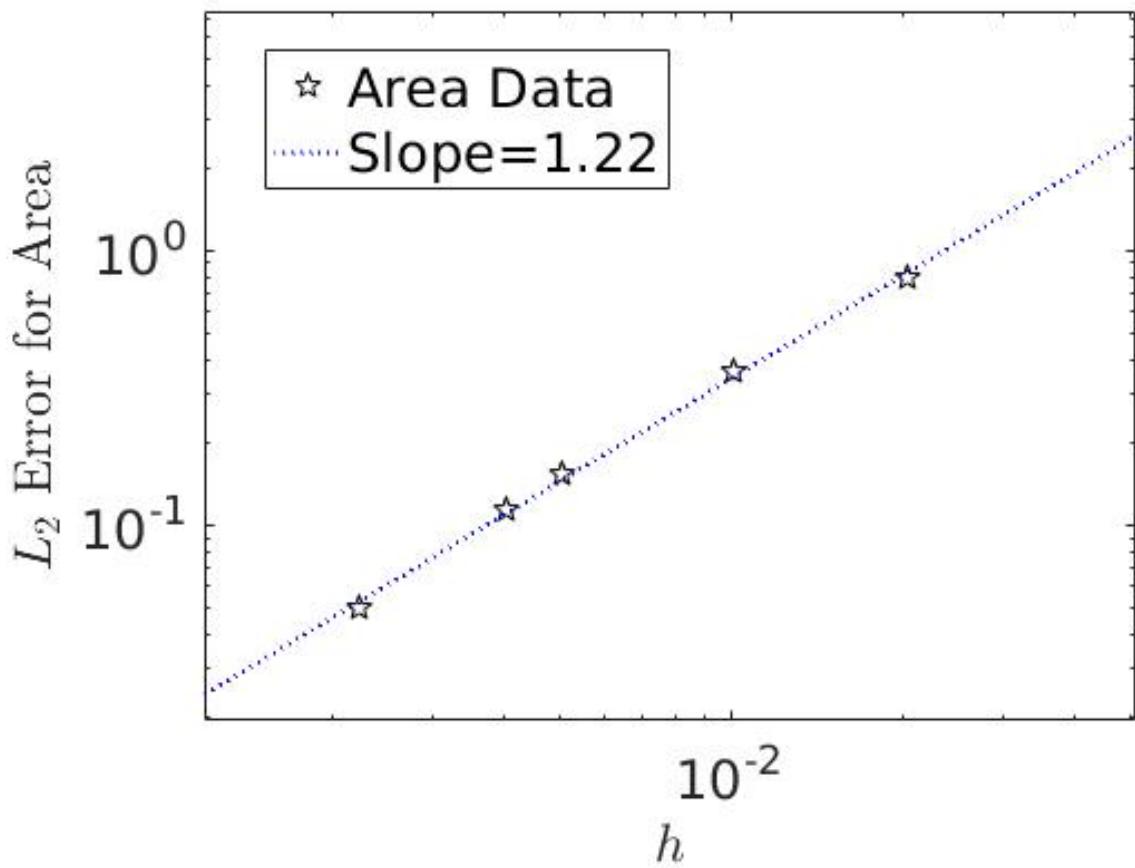


Figure 7: FEM Volume Conservation Accuracy

## 10 FVM Results for Test Case 1

The bubble is distorted and brought back as shown in Figure 8. The first order reinitialization performed at  $t = 2, 4, 6$  distorts the bubble even more as shown in Figure 9. This is probably because the reinitialization is less accurate than the discretization scheme (the FVM discretization). See the calculation details after the figures.

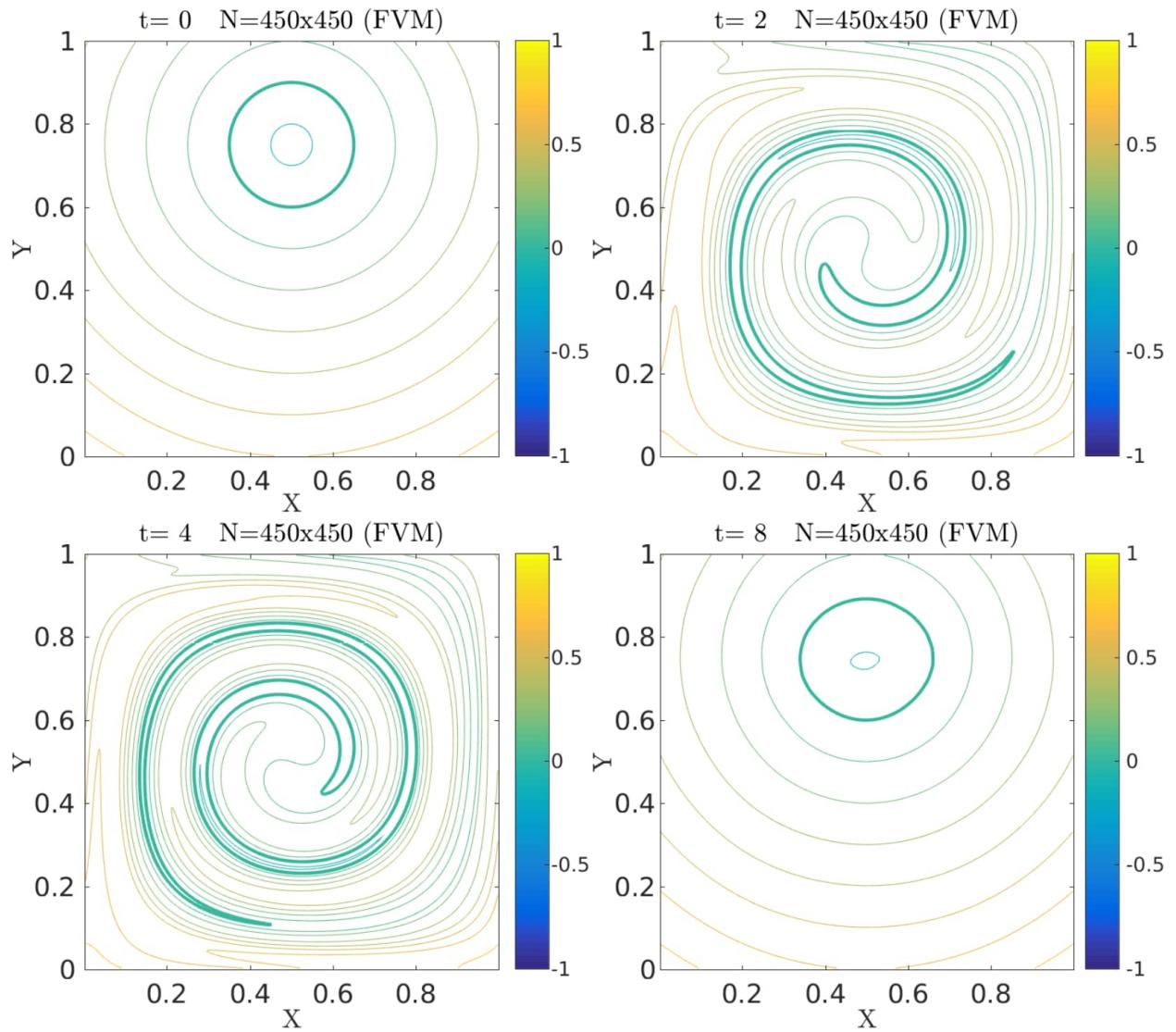


Figure 8: FVM Test Case 1

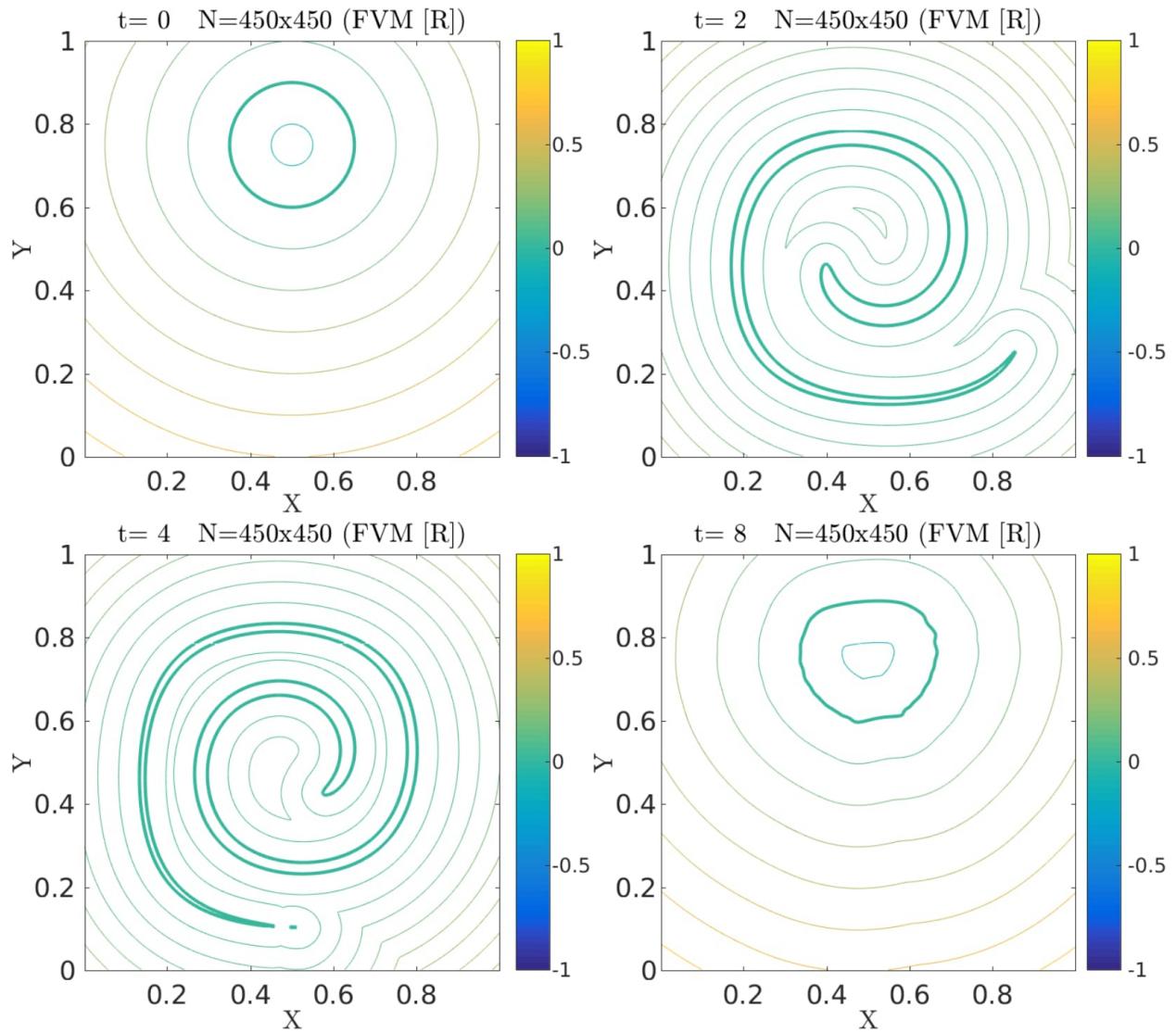


Figure 9: FVM Test Case 1 with reinitialization at  $t = 2, 4, 6$

The error was calculated by running this test for various volumes and calculating the error as:

$$L_2 E = \frac{\sqrt{\int_{\Omega} (\phi^{th} - \phi^n)^2 d\Omega}}{\sqrt{\int_{\Omega} (\phi^{th})^2 d\Omega}} \quad (17)$$

where  $\phi^n$  is the FVM  $\phi$  at  $t = 8$  and  $\phi^{th}$  is the theoretical  $\phi$ . The scheme is approximately second order based on the slope of the error plot, which is consistent with what the authors describe in [1]. No reinitialization were done to construct this error plot.

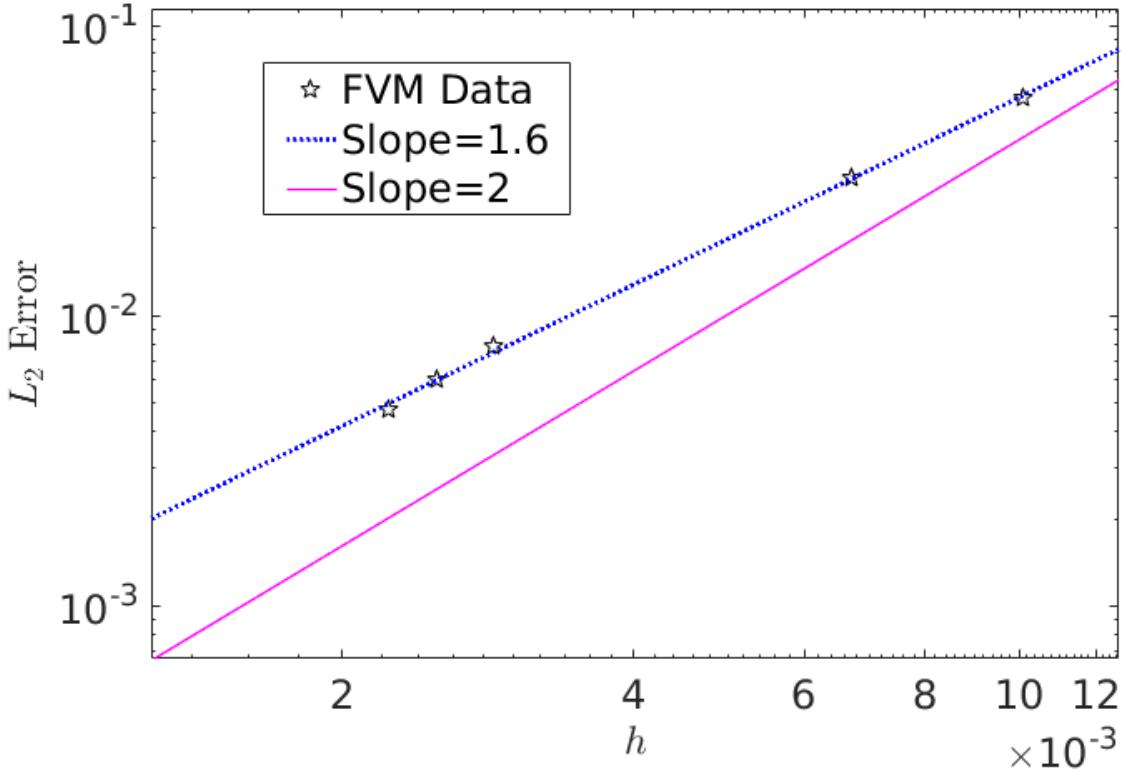


Figure 10: FVM Test Case 1 with reinitialization at  $t = 2, 4, 6$

In Figure 11, the shape distortions caused by the reinitialization on the bubble are clear. Even though the reinitialization scheme was run only three times, it still distorted the bubble shape significantly. There was also significant volume addition as the reinitialized bubble area was 0.07600 while the regular FVM bubble area was 0.07250. The theoretical area was 0.07069.

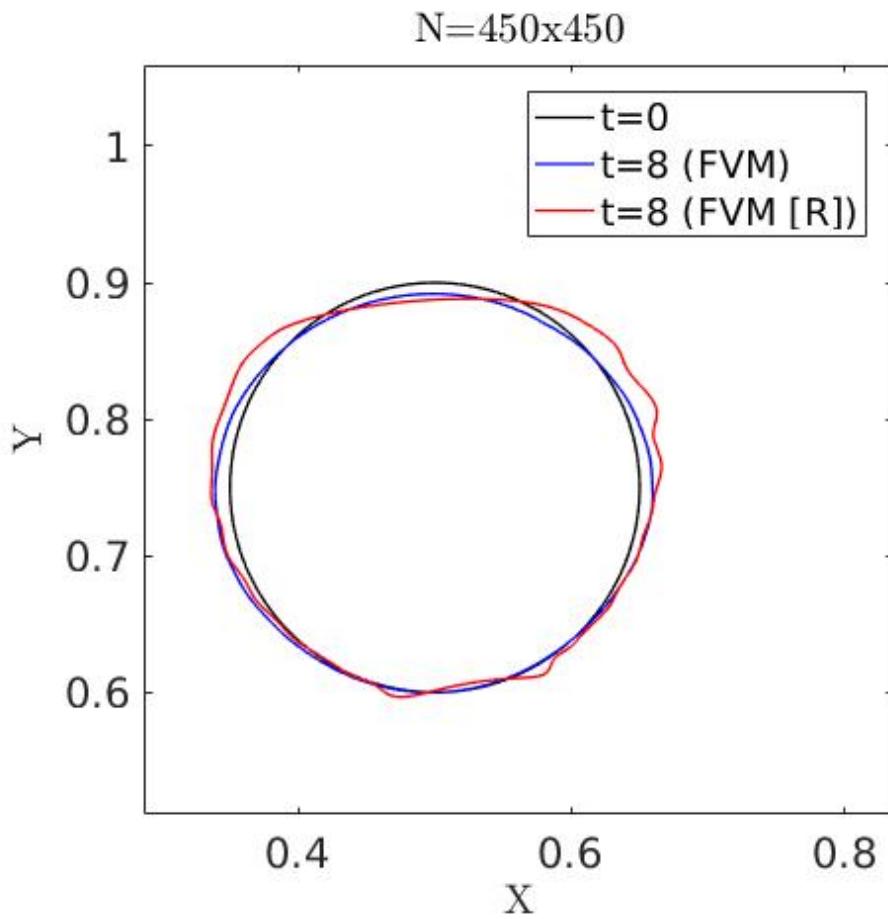


Figure 11: FVM Test Case 1 comparison with reinitialization as the red line

Using Equation 16 to get the error in volume conservation, I confirmed that it is a second order method as seen with the slope in Figure 12.

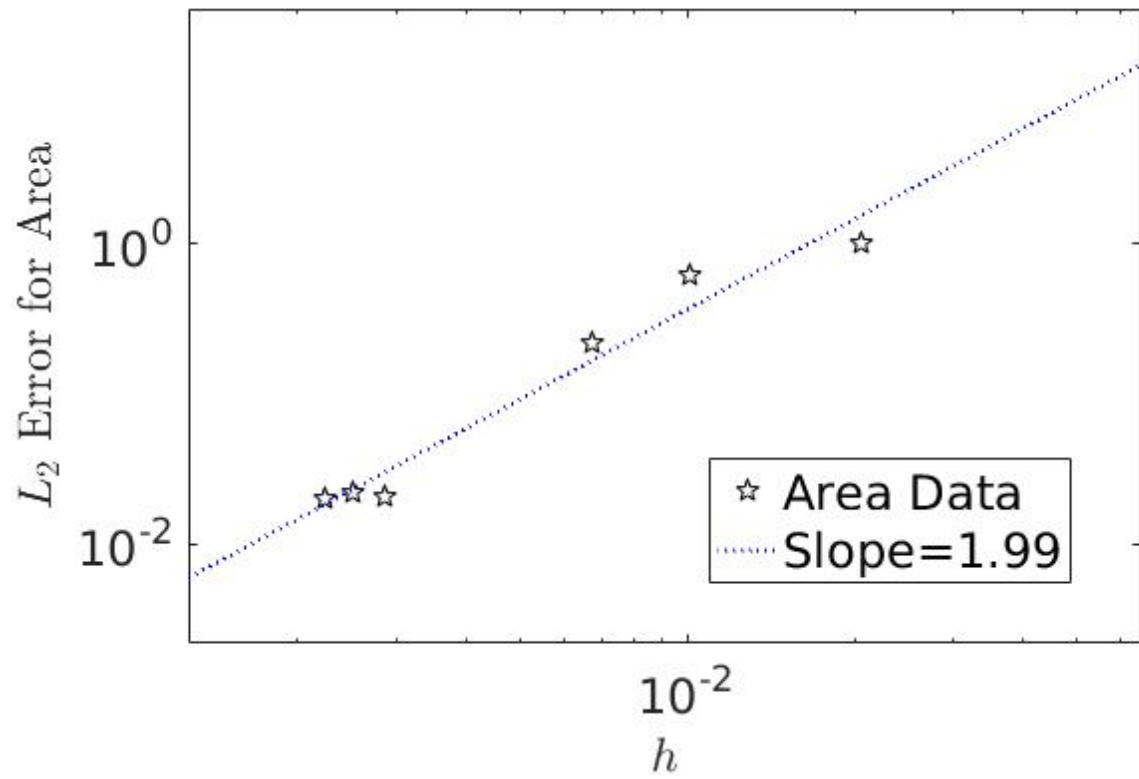


Figure 12: FVM Test Case 1 comparison with reinitialization as the red line

## 11 FVM Results for Test Case 2

The bubble translates back and forth as expected without much volume change as seen in Figure 13.

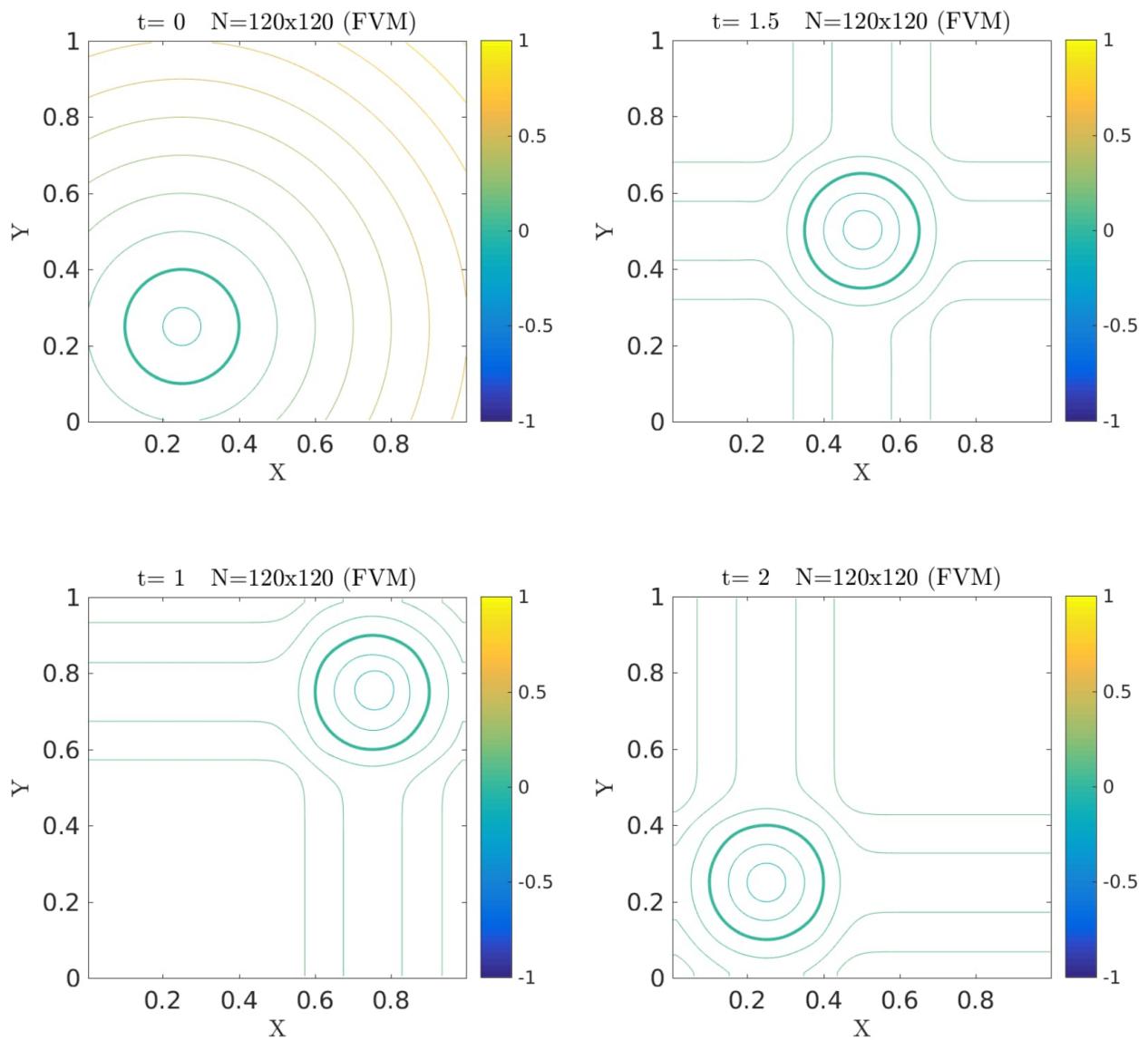


Figure 13: FVM Test Case 2

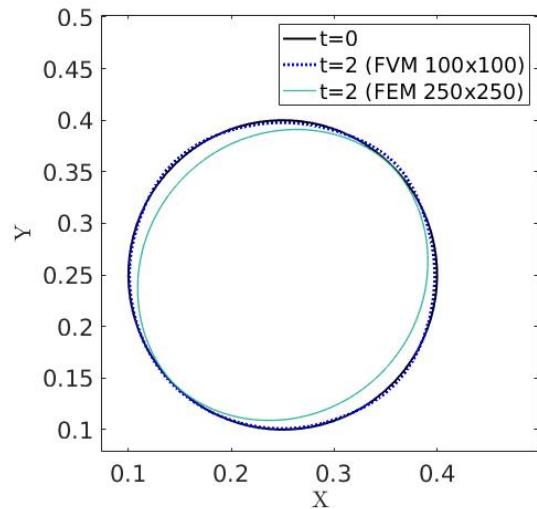


Figure 14: FVM Test Case 2 comparison between FVM and FEM

Using Equation 16 to get the error in volume conservation, I once again confirmed that it is a second order method as seen with the slope in Figure 15.

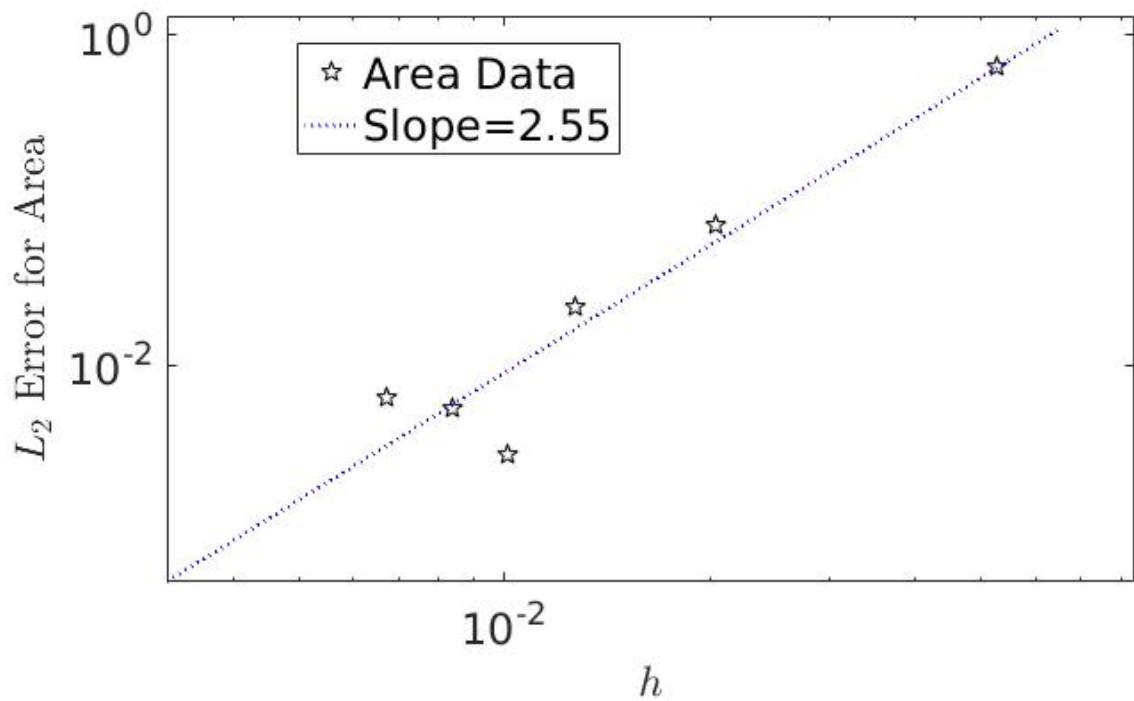


Figure 15: FVM Test Case 1 comparison with reinitialization as the red line

## 12 Conclusion

This project allowed me to explore the level set evolution equation and two different solution methods. I found that the regular finite element discretization with a SUPG stabilization in it doesn't work well for volume conservation especially when there are large topological changes. A good alternative as demonstrated in this report is the finite volume discretization (described by [1]). Reinitialization can help with volume conservation if done carefully. It is also important to mention that if conservation is the most important aspect than the volume of fluid method is a good choice because it is fully conservative by construction. The drawback to this is that the interface is arbitrary so surface tension and any physical phenomenon that relate to the interface will be inaccurate. This is why the level set method would beat out this method for this point and it is still used a great deal for interface problems.

## References

- [1] Peter Frolkovič and Christian Wehner. Flux-based level set method on rectangular grids and computation of first arrival time functions. *Computing and visualization in science*, 12(6):297–306, 2009.
- [2] David L Chopp. Some improvements of the fast marching method. *SIAM Journal on Scientific Computing*, 23(1):230–244, 2001.
- [3] Hendrik C Kuhlmann and Hans-Josef Rath. *Free surface flows*, volume 391. Springer, 2014.