Level Set Methods for Two-Phase Flows with FEM

Deniz Kennedy
Abstract

Level Set Methods for Two-Phase Flows with FEM

Deniz Kennedy

Two-phase flows is a branch of multiphase flows. The aim of the project is to implement two different level set methods and analyse and compare the numerical results. The level set method is used in order to represent the behaviour of the interface between two incompressible fluids in a flow. Reinitialization is a method to straighten a distorted shape of the level set function that might be caused by the numerical solution of the convection equation and/or by the complicated fluid velocity fields. The main purpose of reinitialization is to preserve the level set function, and thus the shape of the interface as much as possible throughout the simulation. In order to avoid the oscillations, the stationary weak form is approximated with Galerkin Least Squares (GLS) finite element instead of standard finite element approximation. In order to create the velocity field of the incompressible fluid for the benchmark case, the Stokes equations are solved. The stability has to be measured carefully as it has imbalance between space stability and time stability as well as with the reinitialization. The convergence rates in the numerical results for the both experiment and benchmark cases show that reinitializations usually give a better result. Further researches for this paper could be using another FEM stabilization method, which is other than GLS, in order to solve the stabilization problem in 2D.
1 Introduction

Multiphase flow simulations are fundamental tools in a wide range of industrial applications such as liquid phase sintering and inkjet printing\cite{1, 3}. These numerical simulations can replace expensive and complicated laboratory experiments. One of these applications is a so-called lab-on-a-chip process, see Figure 1. It can be used to diagnose diseases quickly by analyzing the patient’s blood drops instead of sending the blood sample to a laboratory\cite{11, 12}.

![Figure 1: Lab-on-a-chip](image)

Two-phase flows is a branch of multiphase flows. To be precise, flows including two incompressible fluids that do not mix, such as oil and water, are called two-phase flows. An important feature is the interface between two fluids, and to model two-phase flows the equations that normally describe fluid motion, the Navier-Stokes equations, need to be complemented with a numerical model for the representation of the interface. However, an efficient and accurate implementation of the numerical methods for two-phase flows is not trivial. The aim of the project is to implement two different level set methods and analyse and compare the numerical results.

1.1 The Model Cases

In this project, we consider two different cases. The first one is the experiment case, which has a uniform fluid flow. It is used for testing level set methods. The other one is the benchmark case “lid-driven cavity”. It has a viscous incompressible fluid inside a given box. The upper wall is moving, which causes a swirl inside it, see Figure 2a. The resulting fluid field can be
determined by solving the Stokes equations using for example the finite element method (FEM), see Figure 2. Here we extend the models by adding a small circle-shaped bubble consisting of another fluid that does not mix with the former as in Figure 3 and observe the motion of the new fluid.

![Figure 2: Lid-driven Cavity problem, i.e. the benchmark case](image)

**1.2 The Discretization Method Choice**

The equations from the numerical methods for two-phase flows are usually discretized using the Finite Volume Method (FVM) or FEM in space along with finite difference method (FDM) for time discretization. An important difference between these two methods is that the FEM could be easily improved in giving high order accuracy with more computational cost, whereas it becomes complex and difficult to solve in high order accuracy in the other methods. An example reference can be found in Gohil et al.’s paper [14]. In our case, we use the FEM for the spatial discretization and the FDM for the temporal discretization.

**1.3 The Outline of the Report**

There are three different parts in the paper. First, we start with explaining the theoretical foundations of the level set method for the modelling of two-phase flows. This part consists of three subparts: mathematical model, discretization, and boundary and initial conditions, respectively, which are explained in details. Second, we discuss the fluid mechanics and linear Stokes equations for calculating the stationary flow field in the lid-driven cavity. Third, we set up our numerical experiments by implementing two different level set methods and we test these on two different cases. In
one of the cases, the analytical solution to the level set method is known, whereas in the other case, the analytical solution of the level set method is unknown due to the numerical results from the stokes velocity field, which is the benchmark case. Finally, we discuss and summarise the important parts of this paper in the conclusion section.

In addition to the outline, we include an extra appendix, where it has better results. Since we ran out of time, the difference between the method in the paper and the new method from an article is compared and discussed briefly in Appendix C.

Figure 3: Representation of a two-phase flow. Each incompressible fluid occupies one of the regions $\Omega_1$ and $\Omega_2$. $\Gamma$ represents the interface between the two fluids.
2 The Level Set Method

The level set method is a numerical technique that was introduced in Osher and Sethian’s paper [6]. It creates new algorithms for following fronts propagating with curvature-dependent speed derived from the Hamilton-Jacobi equation. Furthermore, this level set approach has been applied to incompressible two-phase flow since the article of Sussman et al. [7]. Detailed information on the level set method could be found in the book by Osher and Fedkiw [8].

The idea of the level set method is to define the interface between the two fluids by using an implicit function, the level set function \( \phi(x) \), as in Figure 4. After choosing the type of the implicit function, the domain values that give \( \phi(x) = 0 \) represent the interface between the two incompressible fluids. One of the advantages of the level set method is that geometries that change topology such as splitting, developing holes, and merging can be traced easily and another one is that the grid of the domain does not need to be changed.

The approach of the level set method from its mathematical theory to its numerical calculations is presented in the following sections.

2.1 Mathematical Model

The level set method is used in order to represent the behaviour of the interface between two incompressible fluids in a flow. The movement of the interface is followed by the convection equation:

\[
\frac{\partial \phi}{\partial t} + u \cdot \nabla \phi = 0,
\]

where \( \phi \) is the level set function and \( u \) is the fluid velocity. The equation shows that the slope of the level set function, i.e. \( \nabla \phi \), changes and moves along the fluid velocity. This movement happens with time, \( t \), keeping the level set function values as in the part \( \frac{\partial \phi}{\partial t} \). There are more than one level set method to use, which could be different in terms of level set function and reinitialization, see Section 2.1.2 therefore we consider two different level set methods presented in the next subsection.

2.1.1 Two Different Types of Level Set Methods

Two level set methods are compared in this paper. The first one is the standard level set method, in which the implicit function is the signed distance function, an example is given in Figure 4 and the second is the conservative level set method, where the implicit function is the \( \text{tanh} \) of the signed distance function, see Figure 5.
Interface: $\phi = 0$

$\Omega_1$ $\Omega_2$

$\phi > 0$ $\phi < 0$ $\phi > 0$

Figure 4: Illustrating an interface by a signed distance level set function $\phi$ in 1D.

For a circle, the signed distance function is defined with the following condition:

$$\phi(x) = d(x) - R,$$

where $d(x) = \|x\|$ is the distance function and $R$ is the radius of the circle at the given center point $x_c$, in which both the magnitude of $R$ and the vector, $x_c$, are defined by the user. Note that both $x$ and $x_c$ are vectors in 2D. This condition says that $|\nabla \phi| = 1$ at every point at the initial condition.

On the other hand, the conservative level set method has the following condition for a circle:

$$\phi(x) = \tanh(d(x) - R),$$

where the part $d(x) - R$ comes from the distance function. The purpose of taking $\tanh$ is to separate the two fluids in a very steep way rather than in a linear way as in the signed distance function and to fit the relatively unimportant points in the function to a maximum $(+1)$ or minimum point $(-1)$.

These two types have their advantages and disadvantages. Firstly, the standard level set function has a more smooth solution comparing to the other since steep functions usually cause oscillations and make the solution unstable. Secondly, the standard level set function usually develops unphysical volume changes in the fluid phases, whereas the conservative level set earns the term conservative due to the mass conservation\[1, 2\]. Finally, the reinitialization method, see Section 2.1.2 for the conservative level set function depends on more parameters than the standard does, which causes
difficulty in finding approximately correct values for the parameters. More
details are explained in the section 2.2. Note that the two level set methods
have their own methods to reinitialize their level set function.

2.1.2 Reinitialization

Reinitialization is a method to straighten a distorted shape of the level set
function that might be caused by the numerical solution of the convection
equation and/or by the complicated fluid velocity fields. The main purpose
of reinitialization is to preserve the level set function, and thus the shape of
the interface as much as possible throughout the simulation.

The reinitialization method used for the signed distance function in the
standard level set method requires the equation (2.4) to be solved for a few
"artificial" time steps after every movement step of the level set function,

\[ \frac{\partial \phi}{\partial \tau} + S_{\epsilon}(\phi_0)(|\nabla \phi| - 1) = 0, \]  

(2.4)

where \( \tau \) is a pseudo-time for the reinitialization and \( S_{\epsilon}(\phi_0) \) is the smoothed
sign function depending on the initial position \( \phi_0 \) of each movement. It
is assumed one of the most simple reinitialization method, however it is
not trivial to acquire a good result along with it. A more sophisticated
method could be applied to the conservative level set method, which is solved
by the following equation

\[ \frac{\partial \phi}{\partial \tau} + \nabla \cdot (n(1 - \phi^2)) - \nabla \cdot (n \varepsilon \nabla \phi \cdot n) = 0, \]  

(2.5)

where \( n = \frac{\nabla \phi}{|\nabla \phi|} \) is the normal of the interface and \( \varepsilon \) is the diffusion parameter
to be chosen. The linear part \( \nabla \cdot (n \varepsilon \nabla \phi \cdot n) \) corresponds to diffusion in the
direction of the normal of the interface and the nonlinear part \( \nabla \cdot (n(1-\phi^2)) \) corresponds to the compressive flux. These two parts in the equation create a balance by diffusing and compressing around the interface.

2.2 Discretizations

The convection equation (2.1) is first discretized in space with the FEM using the most basic element, the hat function, and then discretized in time with Crank-Nicolson method or vice versa. In order to use the FEM, the weak form for the convection equation is found. Therefore, the function space \( W \) for the level set function \( \phi \) is defined as

\[
W = \{ \omega \in H^1(\Omega) \},
\]

where \( \omega \) is the test function in Hilbert space, and the weak form is acquired by multiplying with the test function and integrating over the domain \( \Omega \)

\[
(\dot{\phi}, \omega) + (u \cdot \nabla \phi, \omega) = 0, \quad \forall \omega \in W
\]

where \( \dot{\phi} = \frac{\partial \phi}{\partial t} \). However, solving this equation is not trivial as certain numerical problems arise in the solution. These problems can be classified as in the following subsections.

2.2.1 Stabilization

The convection equation is discretized in two different dimensions such as space and time. Therefore, we consider the stabilization for both in different subparts.

**Stability in Space**

Solving the stationary equation of the convection equation in its weak form (2.7), i.e. when \( \dot{\phi} = 0 \), with standard Galerkin finite element approximation causes oscillations due to the high sensitivity of big values of the term \( \nabla \phi \), which is difficult to handle for the numerical method. In order to avoid the oscillations, the stationary weak form is approximated with Galerkin Least Squares (GLS) finite element instead of standard finite element approximation (see Appendix A for the solution). The GLS method requires the user to choose the constant value \( c \) as it is used in the following stabilization parameter

\[
\delta = ch/\|u\|_{L^\infty(\Omega)}.
\]

where \( h \) is the mesh size. Note that here another finite element approximation called Streamline Upwind Petrov Galerkin (SUPG) has the exact same variational formulation as the GLS due to the zero load vector, i.e. \( f = 0 \) and
since only the stationary version of the convection equation is considered. The value \( \delta \) should be sufficiently small so that the desired solution will not be affected by the new method.

**Stability in Time**

An ordinary differential equation (ODE) such as \( \dot{y} = Dy \), where \( y = y(t) \) and \( D \) is a matrix, gives us an idea about whether the simulation is stable forever or blows up after certain time. It is measured via the eigenvalues of the matrix \( D \) and if there is at least one positive eigenvalue, it signals that the solution will blow up at certain time, in which the magnitude of the eigenvalue(s) shows how fast it will be. Otherwise, the simulation is stable.

Since there is no force in the convection equation, the ODE form is easily acquired. After semi-discretization with the FEM, see equation (4.2), it turns out that the temporal stability also depends on the stability parameter, \( \delta \), which means that both stabilities depend on the same parameter.

**2.2.2 Resolution**

The interface is followed by the sign change of the level set function, i.e. it needs the level set function to have certain logical slopes that make the function positive and negative as in Figure 4 and 5 and the grid points almost never coincide with the points where the level set function is zero. Moreover, since the conservative level set method is very steep, the following equation is used

\[
\phi(x) = \tanh\left(\frac{(d(x) - R)}{\epsilon h}\right), \tag{2.9}
\]

where \( \epsilon \) is an adjustment for the steepness of the function in Figure 5 and thus, the size of the contour as in Figure 6. If \( \epsilon \) is very small, the slope is very steep and if \( \epsilon \) is very large, the slope is very gradual. It is chosen according to remove the oscillations due to the steepness of the level set function, i.e. the contour is too dense, and \( \epsilon = 2 \) in our case. Moreover, the condition \( h \leq R \) should be sufficient in general as there will be at least one node that the level set function gives a negative value.

**2.2.3 Discretization of Reinitialization Equations**

Reinitialization is needed in order to conserve the shape of the interface due to a deformation occurred in the numerical solution. Since we use the FEM for the spatial discretization, the two different reinitialization equations (2.4) and (2.5) are found in their weak form for the function space \( W \), see the equation (2.6):

\[
(\phi, \omega) + S_\epsilon(\phi_0)(|\nabla \phi|, \omega) = S_\epsilon(\phi_0)(1, \omega) \quad \forall \omega \in W, \quad (2.10)
\]
where $\phi_T = \frac{\partial \phi}{\partial T}$ and the smoothed sign function $S_{\epsilon}(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \epsilon^2}}$ for the distance level set function. For the conservative level set function:

$$\left(\phi_T, \omega\right) + \epsilon(\nabla \phi, \nabla \omega) = -\nabla \cdot ((1 - \phi^2)n)\omega(1, \omega) \quad \forall \omega \in W \quad (2.11)$$

with homogeneous Neumann boundary $(\hat{n} \cdot \nabla \phi, \omega)_{\partial W} = 0$ due to no exerted force on the boundary, i.e. $\hat{n} \cdot \nabla \phi = 0$, where $\hat{n}$ is the normal to the boundary of the given geometry $\Omega$.

Furthermore, since the reinitialization method is an additional requirement only to preserve the interface, it needs to be solved for a few pseudo-time steps in every time step for the level set function. The pseudo-algorithm for the whole level set method is provided in Algorithm 1 in which $\text{iter}$ is adjusted by the user.

### 2.2.4 Parameters

The complete numerical solution depends on six different parameters: the spatial size $h$, the temporal size $\Delta t$, the constant value $c$ for the GLS stability parameter, the reinitialization temporal size $\Delta \tau$, the number of reinitialization iterations $\text{iter}$, and the diffusion parameter $\epsilon$. The last parameter affects only the conservative level set method. The interdependencies of these parameters are shown in Table 1.

![Figure 6: The effect of the $\epsilon$ of the conservative level set function when mesh size $h = 0.05$ at the initial point, i.e. $T = 0$](image)

(a) The contour with $\epsilon = 1$  \quad (b) The contour with $\epsilon = 2$
**Input:** Necessary FEM matrices, parameters, initial value and boundary conditions;

while $t < T$ do

$t = t + \Delta t$;  
Find FEM for $LHS$ and $RHS$ of the convection equation;  
Solve for $\phi = LHS\backslash RHS$;

**Reinitialization:**

Save necessary previous data;  
for $l = 1 : \text{iter}$ where iter is chosen by the user do  
Find FEM for $LHS$ and $RHS$ of the reinitialization equation;  
Solve for $\phi = LHS\backslash RHS$;  
end

end

**Algorithm 1:** Pseudo-code for a level set method with the solutions of the convection and reinitialization equations

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Feature and Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh Size</td>
<td>$h$</td>
<td>The smaller $h$, the higher resolution</td>
</tr>
<tr>
<td>Time Step</td>
<td>$\Delta t$</td>
<td>Proportional to the number of time steps for a fixed time in the simulation</td>
</tr>
<tr>
<td>Stability Parameter Constant</td>
<td>$c$</td>
<td>Stabilization for the convection equation using the FEM, it does not depend on $h$, but the stability parameter $\delta$ does</td>
</tr>
<tr>
<td>Pseudo-Time Step</td>
<td>$\Delta \tau$</td>
<td>Pseudo time step for the reinitialization, depends on $\Delta t$</td>
</tr>
<tr>
<td>Number of Reinitialization Iterations</td>
<td>$\text{iter}$</td>
<td>Adjustment to the reinitialization, mainly depends on $\Delta \tau$ but may also depend on $\Delta t$</td>
</tr>
<tr>
<td>Diffusion Parameter</td>
<td>$\varepsilon$</td>
<td>Smoother for the numerical solution of the conservative level set function, may depend on $h$</td>
</tr>
</tbody>
</table>
2.2.5 Full Discretizations

The convection equation is discretized in space with second order accuracy FEM, that is the elements are linear hat function, and in time with second order accuracy FDM, which is Crank-Nicolson method. The full discretization works for both level set methods and is shown as

\[(2M + \Delta t(C + \delta SD))\xi^{n+1} = (2M - \Delta t(C + \delta SD))\xi^n, \quad (2.12)\]

where $M$ is mass matrix, $C$ is convection matrix, $SD$ is streamline-diffusion matrix, and $\xi$ is the coefficient for the numerical solution $\phi_j = \sum n \xi_j \phi_j$, where $j$ is a node in the FEM, also see Appendix[A] for the full solution. The implementation of the matrices and the numerical calculations are explained in the section[B].

Moreover, the number of iterations, $iter$, of reinitialization are used for their level set function after every time step in the convection equation. The reinitialization equations are discretized in space with second order accuracy FEM and in time with implicit first order accuracy FDM, which is backward Euler method. The full discretization for the standard level set method is shown as

\[M\xi^{n+1} = M\xi^n + \Delta \tau b, \quad (2.13)\]

where $b = S_\epsilon(\phi_0)(1 - |\nabla \tilde{\phi}|)$ is the load vector, in which $\tilde{\phi}$ is the previous solution, and for the conservative level set method as

\[(M + \Delta \tau \varepsilon A)\xi^{n+1} = M\xi^n + \Delta \tau b, \quad (2.14)\]

where $b = 2\tilde{\phi}|\nabla \tilde{\phi}| - (1 - \tilde{\phi}^2)\frac{\Delta \tilde{\phi}}{|\nabla \tilde{\phi}|}$ with the same values as in above. See Appendix[B] for the full solutions.

2.3 Boundary and Initial Conditions

Initially, we give our values here to create the interface. Due to the type of level set method, the boundary conditions are adjusted accordingly.

In the level set function, at the beginning, certain initial values are defined in order to give a unique solution to the convection equation. The initial parameters for creating the level set function are the centre of the circle of the interface, $x_c = (-0.5, -0.5)^T$, and the radius of the circle, $R = 0.1$ for our numerical experiments.

Furthermore, the level set methods are calculated, in which they are already discussed and defined in the section[2.1.1] and the boundary conditions are calculated accordingly to the given level set method. In order to make sure that the standard level set function’s boundary conditions are fixed with the correct values, the analytical solution of the velocity of the
circle should be known. This way, the boundary values change with the correct values every time step. On the other hand, the boundary values for the conservative level set function is always fixed as +1. There is no error added in the boundary values in the numerical experiments. Moreover, one has to be careful how close the interface passes near the boundaries so that there should not be a sudden abruption in the solution. The summary of the set up values can be found in Table 2.

Table 2: Values for setting up

<table>
<thead>
<tr>
<th>Level Set Function</th>
<th>Standard</th>
<th>Conservative</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_c )</td>
<td>((-0.5, -0.5)^T)</td>
<td>((-0.5, -0.5)^T)</td>
</tr>
<tr>
<td>( R )</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>BCs</td>
<td>Fixed, moving with ( x_c )</td>
<td>Completely fixed</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>Not applicable</td>
<td>2</td>
</tr>
</tbody>
</table>
3 Fluid Mechanics

In order to create the velocity field of the incompressible fluid from the benchmark case, we need to solve the Stokes equations. This part explains how it approaches from its mathematical theory to its numerical calculations. The solution follows strictly as in Larson and Bengzon’s book [5].

3.1 Mathematical Model

The governing equations describing fluid motion are derived assuming the conservation of mass and momentum. For incompressible Newtonian fluids, the assumptions lead to the famous Navier-Stokes equations:

\[
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nu \Delta \mathbf{u} - \frac{\nabla p}{\rho} + \mathbf{f}
\]

(3.1a)
\[
\nabla \cdot \mathbf{u} = 0,
\]

(3.1b)

where \( \mathbf{u} \) is flow velocity vector, \( \rho \) is density of the fluid, \( p \) is pressure, \( \nu \) is viscosity, and \( \mathbf{f} \) is a given body force. If laminar flow is assumed, the flow is slow and the nonlinear part \( (\mathbf{u} \cdot \nabla)\mathbf{u} \) and time dependent terms can be omitted. With the assumption of laminar flow, the Navier-Stokes equations become a linear stationary Stokes system, here with the no-slip boundary condition and unit viscosity (\( \nu = 1 \)):

\[
\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega
\]

(3.2a)
\[
\mathbf{u} = g_D, \quad \text{on } \partial \Omega,
\]

(3.2c)

where \( \Omega = \Omega_1 \cup \Omega_2 \) is the domain of a given geometry and \( \partial \Omega \) is the boundary parts of the given geometry as in Figure 3. Imposing boundary conditions gives a unique velocity-pressure \( (\mathbf{u}, p) \) pair solution and a common boundary condition is no-slip, which means that the velocity \( \mathbf{u} \) agrees with a known vector \( g_D \) on the boundary. Note that since there is only the gradient of the pressure \( p \) in the equations, the pressure itself is determined by an arbitrary constant called the hydrostatic pressure level, which requires the pressure to have a zero mean value:

\[
\frac{\int_{\Omega} p \, d\Omega}{|\Omega|} = 0,
\]

(3.3)

where \( |\Omega| \) is the area of the domain. This equation is a characteristic feature for all enclosed flows, i.e. when the boundaries are no-slip condition.
3.2 Discretization

Since we want to solve the problem with the FEM, the equations (3.2) and (3.3) are solved in their weak forms (or variational formulations). The two function spaces $V_g$ and $Q$ for the velocity $u$ and pressure $p$, respectively, are defined as

$$V_g = \{ \nu \in [H^1(\Omega)]^d : \nu|_{\partial \Omega} = g_D \}$$

(3.4a)

$$Q = \{ q \in L^2(\Omega) : (q, 1) = 0 \},$$

(3.4b)

where $\nu$ and $q$ are the test functions in Hilbert and $L^2$-norm spaces, respectively, and $d$ is the space dimension of the domain $\Omega \subset \mathbb{R}^d$, which is $d = 2$ in our case. The following weak forms for each are acquired by multiplying with their test functions and integrating over the domain $\Omega$:

$$(-\hat{n} \cdot \nabla u, \nu)_{\partial \Omega} + (\nabla u : \nabla \nu) + (p\hat{n}, \nu) - (p, \nabla \cdot \nu) = (f, \nu), \quad \forall \nu \in V_g$$

(3.5a)

$$(\nabla \cdot u, q) = 0 \quad \forall q \in Q$$

(3.5b)

where $(a(x), b(x)) = \int_{\Omega} ab \, dx$, $(a(x), b(x))_{\partial \Omega} = \int_{\partial \Omega} ab \, ds$ and $\hat{n}$ is the normal proportional to the boundary $\partial \Omega$. Moreover, the MINI element provides the pair solution $(u, p)$ in solving the resulting form of FEM from the weak forms due to being the simplest inf-sup stable element and easy to implement. The MINI element is illustrated in Figure 7. Note that we use the resulting velocity in the benchmark case and disregard the pressure.

Figure 7: A triangulation of MINI element in 2D where • represents velocity and ○ represents pressure at the given nodes
In our case, the problem is two-dimensional, and we have a box as in Figure 9b. Let \( \Omega = [-1,1] \times [-1,1] \) and the \( g_D = 0 \) at \( \partial \Omega \) as there is no flows at the boundaries, or the flows are at rest, except for the top lid, where \( \mathbf{u}_T = (u_x, 0)^T \). The weak forms (3.5) become

\[
(\nabla \mathbf{u} : \nabla \nu) - (p, \nabla \cdot \nu) = (\mathbf{f}, \nu), \quad \forall \nu \in V_g \tag{3.6a}
\]

\[
(\nabla \cdot \mathbf{u}, q) = 0 \quad \forall q \in Q \tag{3.6b}
\]

and we apply MINI element to these forms along with the zero mean value for pressure condition and the final matrix form can be written as in blocks:

\[
\begin{bmatrix}
A & 0 & B_x & 0 \\
0 & A & B_y & 0 \\
B_x^T & B_y^T & 0 & a \\
0 & 0 & a^T & 0
\end{bmatrix}
\begin{bmatrix}
u_x \\
u_y \\
\mu
\end{bmatrix}
=
\begin{bmatrix}
b_x \\
b_y \\
0 \\
0
\end{bmatrix},
\tag{3.7}
\]

where \( A \) is the stiffness matrix, \( B_x \) and \( B_y \) are divergence matrices for \( x \) and \( y \) directions, \( a \) is the operator from the discretization of the equation (3.3), \( b_x \) and \( b_y \) are the load vectors respect to the \( x \) and \( y \) direction, \( \mu \) is a Lagrangian multiplier and the numerical vector \( \mathbf{u} = (u_x, u_y)^T \) and numerical pressure \( p \). Note that \( \mu \) is used as it is complementary part for the pressure \( p \) to make the complete matrix symmetric.
4 Numerical Experiments and Results

Many experiments have been run due to several different parameters and the most important ones are chosen. This section is divided into different parts. Firstly, we explain what we measure in our numerical solutions. Secondly, we submit values to certain parameters in the discretizations and use these parameters as a reference. Thirdly, we plot and explain different results in the both experiment and benchmark cases. Finally, we discuss the results in the last part.

4.1 Error Measurements

We consider two different types of errors, here referred to as "the level set function error" and "the area error". A simple level set function in 2D is shown in Figure 8a. If the analytical solution of this function at a given time is known, the numerical error can be calculated by comparing the numerical solution to the analytical in each grid point, which is here referred to as the level set function error. Further, since we use the level set method, we are interested in the contour at where the function is zero, see Figure 8b. The area of this interface is measured since our main purpose is to preserve the area as much as possible. We always know the analytical area since we use a circle with the radius $R$, which is chosen by us. However, since we use finite number of grid points, there is always a small error in the measurement of the area from the numerical solution. Nevertheless, this error should not have a big effect on the total numerical error. The area error is the difference between the analytical area and the numerical area including the measurement error. Note that all the errors here are measured as relative errors and we consider the area error to be relatively more important than the level set function error for two-phase flows.

4.2 Velocity Fields and Parameters

In order to test whether the methods work or not, they are tried for an experiment case where the analytical solution to the level set function is known. The artificial velocity field is created with the following formula:

$$u(x) = \begin{pmatrix} u_x(x) \\ u_y(x) \end{pmatrix} = \begin{pmatrix} 0.05 \\ 0.05 \end{pmatrix}$$  \hspace{1cm} (4.1)

where $x = (x, y)^T$ is a point in the given geometry $\Omega$. After validating the results from the experiment case, the same methods are also applied to the benchmark case, see section 3. The velocity fields can be seen in Figure 9.

Moreover, since the interface evolves by the level set method, all necessary parameters are given for different cases in Table 3. The reason why we
(a) The level set function at the initial position, the plane represents the cut at the zero contour in Figure 8b.

(b) The interface with the radius $R = 0.1$ at the centre $x_c = (-0.5, -0.5)^T$.

Figure 8: Initial shape of the level set function and the interface, i.e. the zero contour, with the mesh size $h = 0.05$

(a) Velocity field for the experiment case, the exact solution to the level set function is known due to the constant velocity.

(b) Velocity field for the benchmark case, the exact solution to the level set function is unknown due to the varying velocity field from the solution to the stokes equations.

Figure 9: Velocity fields, $u$, for the different cases with the mesh size $h = 0.05$
choose these values will be clearer in the next sections. Note that we often refer to this table in the results.

Table 3: Parameters and their values in the different cases

<table>
<thead>
<tr>
<th>Level Set Method</th>
<th>Experiment case</th>
<th>Benchmark case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
<td>Conservative</td>
</tr>
<tr>
<td>$h$</td>
<td>varies as $0.1$, $0.05$, and $0.025$</td>
<td>varies as $0.1$, $0.05$, and $0.025$</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>$c$</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>$\Delta \tau$</td>
<td>0.01$\Delta t$</td>
<td>0.001$\Delta t$</td>
</tr>
<tr>
<td>$\text{iter}$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>Not applicable</td>
<td>0.001</td>
</tr>
</tbody>
</table>

4.3 The Experiment Case

The experiment case provides the information about how well the numerical solutions approximate the analytical solutions and what side effects the numerical solutions have. In this section, three different effects are explained and the convergence is illustrated.

4.3.1 Effect of the Stability Constant Parameter $c$

First of all, we want to be sure that the solution is stable in terms of both space and time. In order to be stable in space, a small value for the stability constant parameter, for example $c = 0.05$, is sufficient as it will not change the solution too much (i.e. not add too much diffusion.) However, the time stability also depends on the same parameter. Therefore, in order to check the time stability, we find the eigenvalues, see section 2.2.1, of the matrix in front of the variable $\xi(t)$ in the following system of ODEs, which is derived from the equation (A.8):

$$\dot{\xi}(t) = -M^{-1}(C + \delta SD)\xi(t) \quad (4.2)$$

For $h = 0.05$ and $c = 0.05$, the stability parameter for the experiment case, regardless of any level set function, also becomes $\delta = 0.05$ and the maximum eigenvalue is $\lambda_{max} = +0.64822$, which shows that the solution is unstable in time. Nevertheless, the time stability can be obtained by increasing the value of the constant $\delta$ for the given parameters, however this will cause a harsh stability in space and it might change the solution. For instance,
when $c = 0.5$, $\delta = 0.5$ the maximum eigenvalue is $\lambda_{\text{max}} = +1.3641 \times 10^{-15}$, which is close enough to zero to have time stability.

Figure 10: The blue line is the zero contour of the level set function representing the numerical interface and the red one is the analytical interface. The results using different stability constant parameter are compared. Here the standard level set function is used without reinitialization, and the mesh size is $h = 0.05$. The rest of the parameters are as in Table 3.

Figure 10 shows the results for different values of $c$ using the standard level set function without reinitialization. It is obvious that the solution depends on the value $c$ a lot. When $c$ is small, the circle shape is close to the correct one, but when $c$ is large, the circle area shrinks with time. However, when the convergences are plotted for different mesh sizes such as $h = 0.1, 0.05, 0.025$ (along with the condition $\Delta t/h = 0.1$), there is an obvious convergence when $c = 0.5$ and not a convergence when $c = 0.05$ as in Figure 11. Note that the relative errors are much lower when $c = 0.05$ because of the preservation of the level set function. However, it seems that it cannot preserve the area error during the simulation, which is due to the instability.

For these reasons, we prefer the stable one and we will keep the one with $\lambda_{\text{max}}$ not positive or very close to zero and in this case, it corresponds to $c = 0.5$. 

(a) Zero contours for $c = 0.05$  
(b) Zero contours for $c = 0.5$
Figure 11: Grid convergences for different stability constant parameters using the standard level set function without reinitialization. Here $\Delta t$ is chosen such as $\Delta t/h = 0.1$ for all simulations. The simulations finish when $T = 1$ and the rest of the parameters are as in Table 3.
4.3.2 Effect of the Position of the Interface

Another important effect is how far the circle has "moved" from its initial position. We test four different sets as in Figure 12. These sets vary according to time step and velocity field. The solutions are shown when the circle is at the same position and the results are almost the same for all different sets. This indicates that the distortion of the interface depends on the distance from the initial position of the interface. It also means that the farther the interface is from the initial position, the more shrunken it is. The reason for this is that streamline-diffusion matrix, $SD$, from the stabilization in the FEM depends on the velocity field. Moreover, it also explains why the circle shrinks more in the direction perpendicular to the velocity field than in the direction parallel to the velocity field.

We keep the velocity field as $u = (0.05, 0.05)^T$ and the simulation finishes when $T = 1$, further the value of $\Delta t$ is not very important and we keep the value in the reference table.

4.3.3 Effect of the Reinitialization

The last important part to investigate is the reinitialization. As it is mentioned in section 2.1.2, each level set function has its reinitialization. Therefore, we consider the two level set methods separately. Furthermore, we check the stability of the reinitialization.

The Standard Level Set Method

During one time step $\Delta t$, the signed distance function both moves according to the velocity field and is reinitialized two times. The solutions using the standard level set function is shown both for the case with and without reinitialization in Figure 13. Even though, it is not very clear in the figure, the reinitialization improves, or more precisely preserves, the area of the circle. This reinitialization cannot overlap the area between the analytical interface and the numerical interface completely; neither it can make the numerical area surpass the analytical area.

The Conservative Level Set Method

The conservative level set function is transported as similar to the standard level set function although it has its own sophisticated reinitialization. The different solutions when using and not using reinitialization for the conservative level set function is shown in Figure 14. The area is enlarged a little in all directions. This reinitialization might not overlap the area between the analytical interface and the numerical interface completely, but it can make the numerical area surpass the analytical area.
(a) Zero contours from simulation where \( \mathbf{u} = (0.05, 0.05)^T \), \( \Delta t = 0.1 \), and \( T = 1.0 \)

(b) Zero contours from simulation where \( \mathbf{u} = (0.05, 0.05)^T \), \( \Delta t = 0.001 \), and \( T = 1.000 \)

(c) Zero contours from simulation where \( \mathbf{u} = (0.05, 0.05)^T \), \( \Delta t = 0.01 \), and \( T = 1.00 \)

(d) Zero contours from simulation where \( \mathbf{u} = (0.25, 0.25)^T \), \( \Delta t = 0.001 \), and \( T = 0.200 \)

Figure 12: The blue line is the zero contour of the level set function representing the numerical interface and the red one is the analytical interface. The results are shown for different time steps and velocity fields using the conservative level set function without reinitialization, and a mesh size of \( h = 0.05 \). The rest of the parameters are as in Table 3.
(a) Standard level set function WITHOUT reinitialization
(b) Standard level set function WITH reinitialization

Figure 13: The blue line is the zero contour of the level set function representing the numerical interface and the red one is the analytical interface, and the mesh size is $h = 0.05$. The difference of using and not using the reinitialization for the standard level set function is observed. The rest of the parameters are as in Table 3.

(a) Conservative level set function WITHOUT reinitialization
(b) Conservative level set function WITH reinitialization

Figure 14: The blue line is the zero contour of the level set function representing the numerical interface and the red one is the analytical interface, and the mesh size is $h = 0.05$. The difference of using and not using the reinitialization for the conservative level set function is observed. The rest of the parameters are as in Table 3.
Instability

We apply reinitialization in order to preserve the shape. The reinitialization acts as an exerted force on the level set function and thus the stability condition with the reinitialization should be checked. And indeed, as Figure 15 shows, new arbitrary contours occurs after a long time in the conservative level set method for $c = 0.5$, which is a sign for instability. Also note how the main contour becomes thin and long with time.

Figure 15: Contours for the conservative level set method with reinitialization and $c = 0.5$, where the mesh size is $h = 0.05$. The solution becomes unstable after a long time. The rest of the parameters are as in Table 3.
4.3.4 Grid Convergence

Considering all the effects above, the most suitable parameter values are represented in Table 3. For these parameters, the convergences when decreasing the mesh size are shown in Figure 16. As the figures illustrate, almost all the relative errors converge, whereas the level set functions in the conservative level set method converge does not seem to converge or converge very slowly, and all the reinitialization reduces the relative error for the two level set functions although the standard level set method is less improved than the conservative level set method.

The convergence rates in Table 4 are found by using linear regression. An explanation why the big difference between the rates when measuring the relative level set function error could be because of that the standard level set function is deformed at all the points in the given geometry Ω whereas the conservative level set function is deformed in small part of the geometry closest to the interface.

![Convergence](image)

(a) Standard level set method  
(b) Conservative level set method

Figure 16: Grid convergences for different stability constant parameters using the standard level set function without reinitialization. Here Δt is chosen such as Δt/h = 0.1 for all simulations. The letter R in the label box stands for the one with reinitialization. The simulations finish when T = 1 and the rest of the parameters are as in Table 3.
Table 4: Grid convergence rates for the experiment case in Figure 16

<table>
<thead>
<tr>
<th>Level set function</th>
<th>Standard</th>
<th>Conservative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reinitialization included?</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Relative level set function error</td>
<td>1.1436</td>
<td>1.1718</td>
</tr>
<tr>
<td>Relative area error</td>
<td>1.2065</td>
<td>1.1973</td>
</tr>
</tbody>
</table>
4.4 The Benchmark Case

The same method as in the experiment case is used for the benchmark case, where the analytical solution of the level set function is unknown due to the complexity of the velocity field. Because of this complexity, the value $c$ has to increase to 1 in order to make the solution stable, in which case the maximum positive eigenvalue is $\lambda_{\text{max}} = +2.1436 \times 10^{-4}$. Since the velocity field varies, we consider two different initial positions of the interface: one where the velocity is large and one where the velocity is small. These two positions are named "Lower Position" and "Upper Position" with respect to the initial centre point of the circle.

4.4.1 Upper Position

Considering the velocity field, see Figure 9b, it is reasonable to start the circle at the upper side of the geometry such as $x_c = (-0.5, 0.5)^T$. The rest of the input values are as in Table 2. Zero contours from the simulation is shown in Figure 17 with and without reinitialization for the conservative level set function. It can be seen that the reinitialization gives a larger area. The reason why there is difference between the ones with and without reinitialization is mostly due to the effect of the position of the interface, see section 4.3.2, and more distortion in the interface because of different velocity values in each point in the benchmark case.

4.4.2 Lower Position

To observe how the level set method works when the velocity is small, the values in Table 2 should be sufficiently good as the centre of the circle is at $x_c = (-0.5, -0.5)^T$. Zero contours from the simulation is shown in Figure 18 with and without reinitialization for the conservative level set function. It can be seen that there is almost no difference when the reinitialization is applied. In this case, there is no much difference because of the relatively slow velocity, where the error is small.

4.4.3 Grid Convergences

We can measure only the relative area error due to unknown analytical solution. The convergences when decreasing the mesh size are shown for different starting positions of the interface. When the circle starts at the upper position in the velocity field, the convergence is illustrated in Figure 19 and the rates in Table 3. Further, when the circle starts at the lower position in the velocity field, the convergence is illustrated in Figure 20 and the rates in Table 6.
Figure 17: Zero contours from simulation using the conservative level set function with and without reinitialization for the upper starting position. The green interface represents the one without reinitialization, whereas the blue interface is the one with reinitialization, and the mesh size $h = 0.05$.
Figure 18: Zero contours from simulation using conservative level set function with and without reinitialization at the lower starting position. The green interface represents the one without reinitialization, whereas the blue interface is the one with reinitialization, where the mesh size $h = 0.05$. 
For the upper position interface, even though the both level set methods converge with certain rates, there are different results in the convergence rates. The standard level set method converges slowly, because it depends on the boundary condition every time step, in which the analytical solution is unknown and calculated by using an interpolation. Moreover, the reinitialization for the conservative level set function does not improve, because it might be due to the distortion in the benchmark case, where each point has different velocity value, or the parameters for the reinitialization are not well-chosen. On the other hand, for the lower position interface, the standard level set method does not converge due to the possibility of the domination of the interpolation for the boundary conditions, whereas the improvement in the conservative level set method is due to the slow movement and thus small error.

Table 5: Grid convergence rates for the benchmark case for the upper starting position, values extracted from Figure 19

<table>
<thead>
<tr>
<th>Level set function</th>
<th>Standard</th>
<th>Conservative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reinitialization included?</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Relative area error</td>
<td>0.6951</td>
<td>1.0557</td>
</tr>
<tr>
<td></td>
<td>0.7152</td>
<td>0.9343</td>
</tr>
</tbody>
</table>
Figure 20: Grid convergences the benchmark case when the interface is at the lower position. Here $\Delta t$ is chosen such as $\Delta t/h = 0.1$ for all simulations. The letter R in the label box stands for the one with reinitialization. The simulations finish when $T = 1$ and the rest of the parameters are as in Table 3.

Table 6: Grid convergence rates for the benchmark case for lower starting position, values extracted from Figure 20.

<table>
<thead>
<tr>
<th>Level set function</th>
<th>Standard</th>
<th>Conservative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reinitialization included?</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Relative area error</td>
<td>−0.1478</td>
<td>−0.1418</td>
</tr>
</tbody>
</table>
5 Conclusion

To summarise, we started explaining what the level set methods are in details. We showed how the mathematical model is projected onto the numerical calculation step by step and we illustrated what the side effects could be. Moreover, we created two different velocity fields in order to use the level set methods and observe the results. In the results, different approaches were shown and explained.

The stability has to be measured carefully as it has imbalance between space stability and time stability as well as with the reinitialization. When the stability constant parameter is small, the interface deforms very slowly and thus preserves the shape, but the level set method is unstable in time. When the parameter is large, it becomes stable; however the GLS method causes the interface to shrink. Unfortunately, this stability is valid for only the convection equation and when reinitialization is applied to the equation, it can make the stable solution unstable as it is shown in the experiment case. Moreover, the value $c$ significantly increases when we use it in the benchmark case, because the complex velocity field of the case causes a big twist in the level set function and it requires relatively large stability parameter compared to the one in the experiment case.

The convergence rates in the numerical results for the both experiment and benchmark cases show that reinitializations usually give a better result by reducing the errors, because they try to preserve or expand the distorted interface due to the convection method by using certain formulas. From the results, it is also observed that the reinitialization in the standard level set method tries to preserve as much as it can, whereas the reinitialization in the conservative level set method expands the interface.

From the results of the benchmark case, we could say when the interface is at the velocity field where the magnitude is small, there is more preservation due to the GLS method. Also, the standard level set method does not work well due to the reinitialization, the complexity of the velocity field in the whole geometry, and the interpolation error from the boundaries. Moreover, the reinitialization method might not always improve due to a high level of distortion in the case. Nevertheless, the other methods improve solutions.

In general, the level set methods depend on the stability parameter or the GLS method very much in order to obtain the best solution. For the given values for parameters in the two level set methods, we were able to show that the conservative level set function is more suitable than the standard level function and that the reinitialization methods give a better solution under certain conditions. Nevertheless, they all depend on velocity field, the type of level set functions and reinitializations, boundary conditions, and parameters.

Further researches for this paper could be using another FEM stabiliza-
tion method, which is other than GLS, in order to solve the stabilization problem in 2D or using different kinds of level set methods in order to find one that affects the solution in a much better way.

We found an error in our solution and due to insufficient time, a preliminary is represented in Appendix C.
6 MATLAB and Implementation

All numerical calculations have been done in MATLAB version 2013b on a Linux system with 64 bit at Uppsala University. The matrices depending on a given mesh size, $h$, are implemented accordingly to Larson and Bergzon’s book [5]. All the assemblies of the matrices are introduced from the book and the boundary conditions are implemented accordingly to the book. Moreover, the file from the MathWorks’s link[1] is used in order to calculate the area of a contour in pde structure in MATLAB as the software does not have a suitable function in the given version. Even though, the whole code is written from scratch by the author, the significant part of the code belongs to the book.

Acknowledgements

I would like to thank my supervisor, Hanna Holmgren, for encouraging me in the subject via discussing and helping me out when I am stuck. I am also very grateful for that many thoughts and ideas are simplified by her. One of my thanks goes to my reviewer, Gunilla Kreiss, as well for giving me new ideas and approaches. Another thank is for the IT department of Uppsala University for providing me a good study environment. Finally, I thank all my relatives and friends who have supported me during my master thesis term.
A Discretization of the Convection Equation with the GLS Method

According to Larson and Bengzon’s book[5], when the following equation:
\[ \hat{L}u = f, \]  
(A.1)
where \( \hat{L} \) is a differential operator, \( u \) is the sought solution, and \( f \) is a function, is solved with the GLS method, it gives the following result with the suitable test function, \( \nu \), that is a subset of a function space, \( V \):
\[ (\hat{L}u, \nu) + \delta(\hat{L}u, \hat{L}\nu) = (f, \nu) + \delta(f, \hat{L}\nu) \quad \forall \nu \in V, \]  
(A.2)
where \( \delta \) is the stabilization parameter. We start solving the equation (2.1) in the stationary form
\[ u \cdot \nabla \phi = 0, \]  
(A.3)
where \( \hat{L} = u \cdot \nabla \), \( u = \phi \), and \( f = 0 \). Therefore, the GLS method for the stationary equation with the given test function, \( \omega \), and function space is, \( W \):
\[ (u \cdot \nabla \phi, \omega) + \delta(u \cdot \nabla \phi, u \cdot \nabla \omega) = 0 \quad \forall \omega \in W, \]  
(A.4)
and thus, the convection equation becomes
\[ (\frac{\partial \phi}{\partial t}, \omega) + (u \cdot \nabla \phi, \omega) + \delta(u \cdot \nabla \phi, u \cdot \nabla \omega) = 0 \quad \forall \omega \in W. \]  
(A.5)
The FEM reads “Find the numerical solution \( \phi_h \in W \) of the following equation
\[ (\frac{\partial \phi_h}{\partial t}, \omega) + (u \cdot \nabla \phi_h, \omega) + \delta(u \cdot \nabla \phi_h, u \cdot \nabla \omega) = 0 \quad \forall \omega \in W.” \]  
(A.6)
Since we use the hat function for the test function, \( \omega = \varphi_i \) and \( \phi_h = \sum_{j=1}^{n} \xi_j(t) \varphi_j \) are imported into the last equation:
\[ \sum_{j=1}^{n} \hat{\xi}_j(\varphi_j, \varphi_i) + \sum_{j=1}^{n} \xi_j(u \cdot \nabla \varphi_j, \varphi_i) + \delta \sum_{j=1}^{n} \xi_j(u \cdot \nabla \varphi_j, u \cdot \nabla \varphi_i) = 0 \quad \text{for} \quad i = 1, \ldots, n \]  
(A.7)
and the spatial discretization with the FEM is finalized as:
\[ M\ddot{\xi} + C\dot{\xi} + \delta SD\xi = 0, \]  
(A.8)
where $M$ is the mass matrix, $C$ is the convection matrix, and $SD$ is the streamline-diffusion matrix. In addition to this discretization, the Crank-Nicolson method is used in the time discretization, therefore the full discretized convection equation with GLS method can be written as:

$$(2M + \Delta t(C + \delta SD))\xi^{n+1} = (2M - \Delta t(C + \delta SD))\xi^n,$$

(A.9)

where $\Delta t$ is the time step. Since we have an initial level set function value, $\xi^0 = \phi_0$ gives an unique numerical solution.
B Discretization of the Reinitializations

B.1 The Standard Level Set Method

We resume from Equation (2.10). Since the part $|\nabla \phi|$ is the nonlinear, we use the previous solution $\tilde{\phi}$. Moreover, the hat function for the test function, $\omega = \varphi_i$ and $\phi_h = \sum_{j=1}^{n} \xi_j(\tau) \varphi_j$ are introduced, the equation thus becomes

$$\sum_{j=1}^{n} \xi_j(\varphi_j, \varphi_i) + S_{\varepsilon}(\phi_0)(|\nabla \tilde{\phi}|, \varphi_i) = S_{\varepsilon}(\phi_0)(1, \varphi_i) \text{ for } i = 1 \ldots n. \quad (B.1)$$

The semi discretized equation becomes

$$M\dot{\xi} = b, \quad (B.2)$$

where $M$ is the mass matrix and $b_i = S_{\varepsilon}(\phi_0)(1 - |\nabla \tilde{\phi}|)(1, \varphi_i)$ is the load vector on node $i$. The equation is explicit in the term of time discretization. The following full discretized equation is acquired with forward Euler, backward Euler, or Crank-Nicolson method applied in the time discretization:

$$M\xi^{n+1} = M\xi^n + \Delta \tau b, \quad (B.3)$$

where $\Delta \tau$ is the time step for the reinitialization method. Note that since the value $|\nabla \tilde{\phi}|$ can be as small as zero, we multiply the equation by $|\nabla \tilde{\phi}|$ in order to prevent the division with zero. Warning: Even though, it is

B.2 The Conservative Level Set Method

We resume from Equation (2.11). Since the part $\nabla \cdot ((1 - \phi^2)\mathbf{n}) = \nabla \cdot ((1 - \phi^2)\frac{\nabla \tilde{\phi}}{|\nabla \tilde{\phi}|})$ is the nonlinear, we use the previous solution $\tilde{\phi}$. Moreover, the hat function for the test function, $\omega = \varphi_i$ and $\phi_h = \sum_{j=1}^{n} \xi_j(\tau) \varphi_j$ are introduced, the equation thus becomes

$$\sum_{j=1}^{n} \xi_j(\varphi_j, \varphi_i) + \varepsilon \sum_{j=1}^{n} \xi_j(\nabla \varphi_j, \nabla \varphi_i) = -(1 - \tilde{\phi}^2)(\nabla \cdot \frac{\nabla \tilde{\phi}}{|\nabla \tilde{\phi}|}) + \frac{\nabla \tilde{\phi}}{|\nabla \tilde{\phi}|} \nabla (1 - \tilde{\phi}^2)(1, \varphi). \quad (B.4)$$

where $M$ is the mass matrix, $A$ is the stiffness matrix and $b_i = [2\tilde{\phi}|\nabla \tilde{\phi}| - (1 - \tilde{\phi}^2)\frac{\Delta \phi}{|\nabla \tilde{\phi}|}](1, \varphi_i)$ is the simplified load vector on node $i$. The equation can be fully discretized with the backward Euler method applied in the time discretization:

$$(M + \Delta \tau \varepsilon A)\xi^{n+1} = M\xi^n + \Delta \tau b, \quad (B.5)$$

where $\Delta \tau$ is the time step for the reinitialization method. Note that since the value $|\nabla \tilde{\phi}|$ can be as small as zero, we multiply the equation by $|\nabla \tilde{\phi}|$ in order to prevent the division with zero. Warning: Even though, it is
mathematically correct, the computation for a very small mesh size such as $h = 0.025$ might give completely wrong depending on the limits of a computer. If this is the case, add a small value to the denominator $|\nabla \phi|$ instead of multiplying for the entire equation.
C More on the Report

This part is added due to discovering our error in the report during finalising it. We discovered that we made an error by considering the GLS stabilization method for the convection part of the level set method and not the whole time-dependent equation. We also decided that the alternative to add the stabilization of the whole time-dependent equation would be insufficient for the time left on the project. Therefore, we decided to change to another method, the "Residual based artificial viscosity method" from Nazarov’s article \[15\] instead of GLS stabilization method, which is also easy to implement.

Instead of using the stabilized convection method GLS, Residual based artificial viscosity method is used. Thus, the equation \(A.8\) becomes

\[ M\dot{\xi} + C\xi + \delta A\xi = 0, \]

where \(M\) is the mass matrix, \(C\) is the convection matrix, and \(A\) is the stiffness matrix and the stability parameter is changed to \(\delta = 2ch \|u\|_{L^\infty(\Omega)}\). The rest values for the parameters and the reinitializations are the same. We present different plots for the experiment case when \(c = 0.03\).

Figure 21 is the comparison to Figure 15 in the report. Since the parameters for the reinitialization is not modified according to the new discretized convection equation, unnecessary zero contours occur in unrelated region. Nevertheless, the main part preserves its shape for a long time, even though it expands a little, and gives a stable result, in which it was unstable with GLS method and reinitialization for the conservative level set function.

Figure 22 is the comparison to Figure 14 in the report. The new method shows that a good preservation for the interface due to distribution in the normal of the interface, whereas the interface is distorted in the direction perpendicular to the velocity field in the old method. Nevertheless the reinitialization works well for the both cases.

Figure 23 is the comparison to Figure 16 in the report. Note that the relative errors are smaller. There is no big difference for the standard level set method, whereas in the case of the conservative level set method, the convergences are not very straight. The reason is that the parameter values in Table 3 for the experiment case except for \(c = 0.03\) are probably not sufficiently good. The rates are as in Table 7 and they also suffer from the same reason.
Figure 21: Contours for the conservative level set method with reinitialization and $c = 0.03$, where the mesh size is $h = 0.05$. The solution is still stable after a long time. The rest of the parameters are as in Table 3.

Table 7: Grid convergence rates for the experiment case in Figure 23

<table>
<thead>
<tr>
<th>Level set function</th>
<th>Standard</th>
<th>Conservative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reinitialization included?</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Relative level set function error</td>
<td>1.3063</td>
<td>1.3158</td>
</tr>
<tr>
<td>Relative area error</td>
<td>0.8645</td>
<td>0.8531</td>
</tr>
</tbody>
</table>
Figure 22: The blue line is the zero contour of the level set function representing the numerical interface and the red one is the analytical interface, and the mesh size is $h = 0.05$. The difference of using and not using the reinitialization for the standard level set function is observed. The rest of the parameters are as in Table 3 except for $c = 0.03$.

Figure 23: Grid convergences the benchmark case when the interface is at the upper position. Here $\Delta t$ is chosen such as $\Delta t/h = 0.1$ for all simulations. The letter R in the label box stands for the one with reinitialization. The simulations finish when $T = 1$ and the rest of the parameters are as in Table 3 except for $c = 0.03$. 
References


