Level Set Methods for Two-Phase Flows with FEM

Deniz Kennedy

Department of Information Technology
Uppsala University

November 25, 2014

Table of Contents

Table of Contents

Introduction

Importance of Multiphase Flow Simulations

- Multiphase flow simulations are fundamental tools in a wide range of industrial applications such as liquid phase sintering and inkjet printing

Introduction

Importance of Multiphase Flow Simulations

- Multiphase flow simulations are fundamental tools in a wide range of industrial applications such as liquid phase sintering and inkjet printing
- Replacing expensive and complicated laboratory experiments

Introduction

Importance of Multiphase Flow Simulations

- Multiphase flow simulations are fundamental tools in a wide range of industrial applications such as liquid phase sintering and inkjet printing
- Replacing expensive and complicated laboratory experiments
- **Two-phase flow** is a branch of multiphase flows

Introduction

Importance of Multiphase Flow Simulations

- Multiphase flow simulations are fundamental tools in a wide range of industrial applications such as liquid phase sintering and inkjet printing
- Replacing expensive and complicated laboratory experiments
- **Two-phase flow** is a branch of multiphase flows
    - Flows including two incompressible fluids that do not mix, such as oil and water, are called two-phase flows

Introduction

Numerical Aspect and Goal

- An efficient and accurate implementation of the numerical methods for two-phase flows is not trivial

Introduction

Numerical Aspect and Goal

- 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

Introduction

Discretization : FEM versus FVM

- Flow simulations are usually discretized using the Finite Volume Method (FVM) or Finite Element Method (FEM) in space along with finite difference method (FDM) for time discretization

Introduction

Discretization : FEM versus FVM

- Flow simulations are usually discretized using the Finite Volume Method (FVM) or Finite Element Method (FEM) in space along with finite difference method (FDM) for time discretization
- **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

Introduction

Discretization : FEM versus FVM

- Flow simulations are usually discretized using the Finite Volume Method (FVM) or Finite Element Method (FEM) in space along with finite difference method (FDM) for time discretization
- **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
- Therefore, we use the FEM for the spatial discretization and the FDM for the temporal discretization

Table of Contents

Level Set Methods

What is a Level Set Method?

- It is a numerical technique that is derived from the Hamilton-Jacobi equation

Level Set Methods

What is a Level Set Method?

- It is a numerical technique that is derived from the Hamilton-Jacobi equation
- The idea of the level set method is to define the interface between the two fluids by using an implicit function called *level set function*, $\phi(\mathbf{x})$

Level Set Methods

What is a Level Set Method?

- It is a numerical technique that is derived from the Hamilton-Jacobi equation
- The idea of the level set method is to define the interface between the two fluids by using an implicit function called *level set function*, $\phi(\mathbf{x})$
- After choosing the type of the implicit function, the domain values that give $\phi(\mathbf{x}) = 0$ represent the interface between the two incompressible fluids

Level Set Methods

What is a Level Set Method?

- It is a numerical technique that is derived from the Hamilton-Jacobi equation
- The idea of the level set method is to define the interface between the two fluids by using an implicit function called *level set function*, $\phi(\mathbf{x})$
- After choosing the type of the implicit function, the domain values that give $\phi(\mathbf{x}) = 0$ represent the interface between the two incompressible fluids
- Finally, *reinitialization* is used. 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.

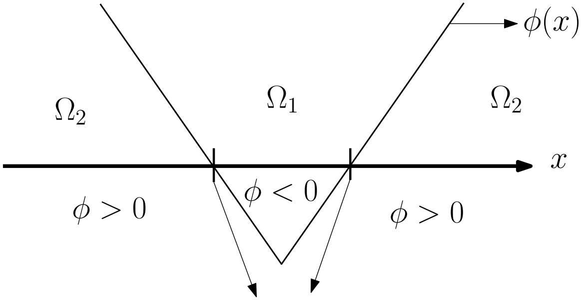
Level Set Methods

What is a Level Set Method?

- It is a numerical technique that is derived from the Hamilton-Jacobi equation
- The idea of the level set method is to define the interface between the two fluids by using an implicit function called *level set function*, $\phi(\mathbf{x})$
- After choosing the type of the implicit function, the domain values that give $\phi(\mathbf{x}) = 0$ represent the interface between the two incompressible fluids
- Finally, *reinitialization* is used. 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.
- Level Set Method = Level Set Function + Reinitialization

Level Set Methods

Level Set Functions



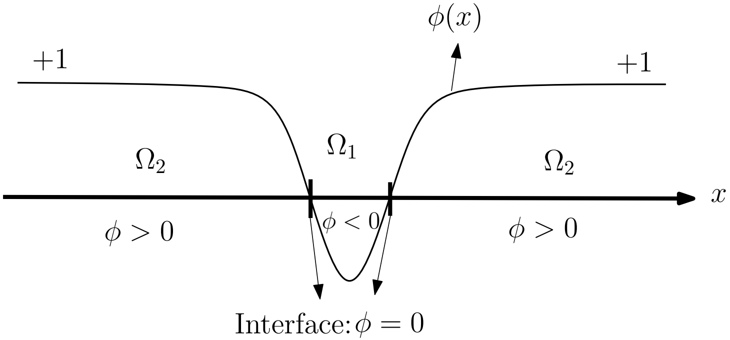
Interface: $\phi = 0$

Standard level set function:

$$\phi(\mathbf{x}) = d(\mathbf{x}) - R$$

Level Set Methods

Level Set Functions



Conservative level set function:

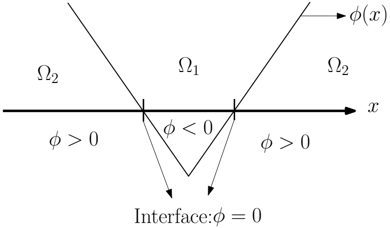
$$\phi(\mathbf{x}) = \tanh(d(\mathbf{x}) - R)$$

Level Set Methods

Reinitialization

Reinitialization : Standard Level Set Function

$$\underbrace{\frac{\partial \phi}{\partial \tau}}_{\text{Mass}} + \underbrace{S_\epsilon(\phi_0)(|\nabla \phi| - 1)}_{\text{Smoothing}} = 0$$



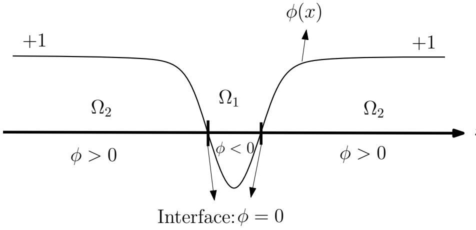
Level Set Methods

Reinitialization

Reinitialization : Conservative Level Set Function

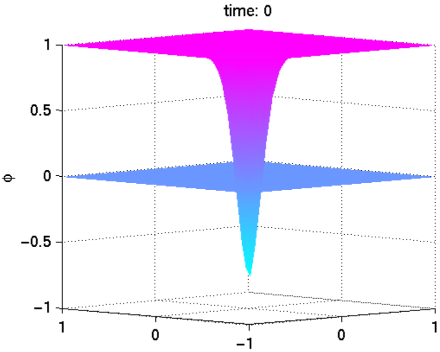
$$\underbrace{\frac{\partial \phi}{\partial \tau}}_{\text{Mass}} + \underbrace{\nabla \cdot (\mathbf{n}(1-\phi^2))}_{\text{Compressing}}$$

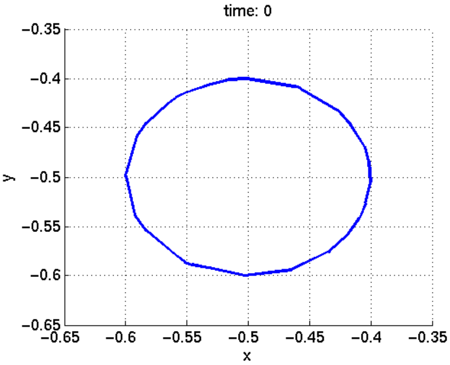
$$- \underbrace{\nabla \cdot (\mathbf{n}\varepsilon \nabla \phi \cdot \mathbf{n})}_{\text{Diffusing}} = 0$$



Level Set Methods

Initial position in our case





Conservative level set function in 2D

Level Set Methods

How a Level Set Function moves

The convection equation:

$$\underbrace{\frac{\partial \phi}{\partial t}}_{\text{Mass}} + \underbrace{\mathbf{u} \cdot \nabla \phi}_{\text{Movement}} = 0$$

where $\mathbf{u}$ is the fluid vector.

Level Set Methods

Why to use a Level Set Method?

Advantages of level set methods:

- Geometries that change topology such as splitting, developing holes, and merging can be traced easily

Level Set Methods

Why to use a Level Set Method?

Advantages of level set methods:

- Geometries that change topology such as splitting, developing holes, and merging can be traced easily
- The grid of the domain does not need to be changed

Level Set Methods

Discretizations

Discretization is straightforward:

- The convection equation is first discretized in space with the FEM using the most basic element, the hat function

Level Set Methods

Discretizations

Discretization is straightforward:

- The convection equation is first discretized in space with the FEM using the most basic element, the hat function
- Then it is discretized in time with Crank-Nicolson method (FDM) or vice versa

Level Set Methods

Discretizations

Discretization is straightforward:

- The convection equation is first discretized in space with the FEM using the most basic element, the hat function
- Then it is discretized in time with Crank-Nicolson method (FDM) or vice versa
- The full discretization becomes $(2M + \Delta tC)\xi^{n+1} = (2M - \Delta tC)\xi^{n}$, where

Level Set Methods

Discretizations

Discretization is straightforward:

- The convection equation is first discretized in space with the FEM using the most basic element, the hat function
- Then it is discretized in time with Crank-Nicolson method (FDM) or vice versa
- The full discretization becomes $(2M + \Delta t C)\xi^{n+1} = (2M - \Delta t C)\xi^{n}$, where
  - M : the mass matrix and C : the convection matrix.

Level Set Methods

Discretizations

Discretization is straightforward:

- The convection equation is first discretized in space with the FEM using the most basic element, the hat function
- Then it is discretized in time with Crank-Nicolson method (FDM) or vice versa
- The full discretization becomes $(2M + \Delta t C)\xi^{n+1} = (2M - \Delta t C)\xi^{n}$, where
  - M : the mass matrix and C : the convection matrix.
  - Δt : time step

Level Set Methods

Discretizations

Discretization is straightforward:

- The convection equation is first discretized in space with the FEM using the most basic element, the hat function
- Then it is discretized in time with Crank-Nicolson method (FDM) or vice versa
- The full discretization becomes $(2M + \Delta tC)\xi^{n+1} = (2M - \Delta tC)\xi^{n}$, where
  - M : the mass matrix and C : the convection matrix.
  - Δt : time step
  - ξ is the coefficient for the linear hat function, the solution for the level set function $\phi(x)$ at given time n.

Level Set Methods

Discretizations

Discretization is straightforward:

- The convection equation is first discretized in space with the FEM using the most basic element, the hat function
- Then it is discretized in time with Crank-Nicolson method (FDM) or vice versa
- The full discretization becomes $(2M + \Delta tC)\xi^{n+1} = (2M - \Delta tC)\xi^{n}$, where
  - M : the mass matrix and C : the convection matrix.
  - Δt : time step
  - ξ is the coefficient for the linear hat function, the solution for the level set function $\phi(x)$ at given time n.

Level Set Methods

Discretizations: Numerical Problems

- Solving the convection equation numerically is NOT trivial due to numerical problems

Level Set Methods

Discretizations: Numerical Problems

- Solving the convection equation numerically is NOT trivial due to numerical problems
- We have identified four different numerical problems such as

Level Set Methods

Discretizations: Numerical Problems

- Solving the convection equation numerically is NOT trivial due to numerical problems
- We have identified four different numerical problems such as
    1. Stabilization problem

Level Set Methods

Discretizations: Numerical Problems

- Solving the convection equation numerically is NOT trivial due to numerical problems
- We have identified four different numerical problems such as
  1. Stabilization problem
  2. Resolution problem

Level Set Methods

Discretizations: Numerical Problems

- Solving the convection equation numerically is NOT trivial due to numerical problems
- We have identified four different numerical problems such as
  1. Stabilization problem
  2. Resolution problem
  3. Necessity of Reinitialization

Level Set Methods

Discretizations: Numerical Problems

- Solving the convection equation numerically is NOT trivial due to numerical problems
- We have identified four different numerical problems such as
  1. Stabilization problem
  2. Resolution problem
  3. Necessity of Reinitialization
  4. Parameters (arises from solving the previous problems)

Level Set Methods

Discretizations: Stabilization Problem

We want a stable solution in terms of space and time:

- Stability in space:

Level Set Methods

Discretizations: Stabilization Problem

We want a stable solution in terms of space and time:

- Stability in space:
    - **Problem**: Standard finite element approximation causes oscillations due to the high sensitivity of big values of the term $\nabla\phi$

Level Set Methods

Discretizations: Stabilization Problem

We want a stable solution in terms of space and time:

- Stability in space:
  - **Problem**: Standard finite element approximation causes oscillations due to the high sensitivity of big values of the term $\nabla\phi$
  - **Solution**: Instead, it is approximated with Galerkin Least Squares (GLS) finite element and the following stabilization parameter is introduced:
    $\delta = ch/\|\mathbf{u}\|_{L^\infty(\Omega)}$
    where h is mesh size, c is a constant to be chosen, $\mathbf{u}$ is the fluid velocity and Ω is the given space domain

Level Set Methods

Discretizations: Stabilization Problem

We want a stable solution in terms of space and time:

- Stability in space:
  - **Problem**: Standard finite element approximation causes oscillations due to the high sensitivity of big values of the term $\nabla\phi$
  - **Solution**: Instead, it is approximated with Galerkin Least Squares (GLS) finite element and the following stabilization parameter is introduced:
    $\delta = ch/\|\mathbf{u}\|_{L^\infty(\Omega)}$
    where h is mesh size, c is a constant to be chosen, $\mathbf{u}$ is the fluid velocity and Ω is the given space domain
- Stability in time:

Level Set Methods

Discretizations: Stabilization Problem

We want a stable solution in terms of space and time:

- Stability in space:
  - **Problem**: Standard finite element approximation causes oscillations due to the high sensitivity of big values of the term $\nabla\phi$
  - **Solution**: Instead, it is approximated with Galerkin Least Squares (GLS) finite element and the following stabilization parameter is introduced:
    $\delta = ch/\|\mathbf{u}\|_{L^\infty(\Omega)}$
    where h is mesh size, c is a constant to be chosen, $\mathbf{u}$ is the fluid velocity and Ω is the given space domain
- Stability in time:
  - **Problem**: The ordinary differential equation (ODE), $\dot{\xi} = (-M^{-1}C)\xi$ is not stable in time due to the positive eigenvalues in the matrix $-M^{-1}C$

Level Set Methods

Discretizations: Stabilization Problem

We want a stable solution in terms of space and time:

- Stability in space:
  - **Problem**: Standard finite element approximation causes oscillations due to the high sensitivity of big values of the term $\nabla\phi$
  - **Solution**: Instead, it is approximated with Galerkin Least Squares (GLS) finite element and the following stabilization parameter is introduced:
    $\delta = ch/\|\mathbf{u}\|_{L^\infty(\Omega)}$
    where h is mesh size, c is a constant to be chosen, $\mathbf{u}$ is the fluid velocity and Ω is the given space domain
- Stability in time:
  - **Problem**: The ordinary differential equation (ODE), $\dot{\xi} = (-M^{-1}C)\xi$ is not stable in time due to the positive eigenvalues in the matrix $-M^{-1}C$
  - **Solution**: The GLS method gives us a parameter to play around, we could modify it accordingly

Level Set Methods

Discretizations: Resolution Problem

Discretization gives number of nodes which might not represent the solution in a correct way

- **Problem**: The level set function for all the nodes might give a positive value

Level Set Methods

Discretizations: Resolution Problem

Discretization gives number of nodes which might not represent the solution in a correct way

- **Problem**: The level set function for all the nodes might give a positive value
- **Solution**: In order to see two different fluids, there is at least one node that the level set function gives a negative value and one condition could be $h \leq R$

Level Set Methods

Discretizations: Resolution Problem

Discretization gives number of nodes which might not represent the solution in a correct way

- **Problem**: The level set function for all the nodes might give a positive value
- **Solution**: In order to see two different fluids, there is at least one node that the level set function gives a negative value and one condition could be $h \leq R$
- **Problem**: The conservative level set function might be too steep so that it can cause oscillations

Level Set Methods

Discretizations: Resolution Problem

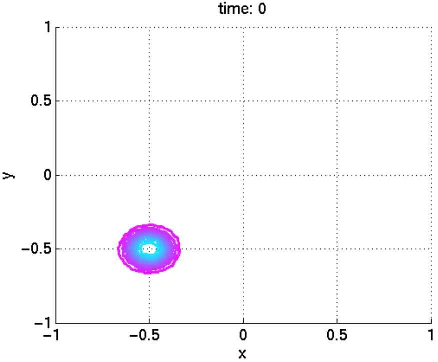
Discretization gives number of nodes which might not represent the solution in a correct way

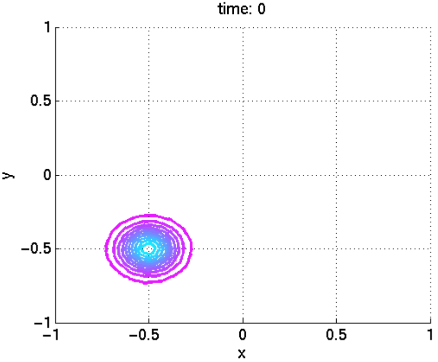
- **Problem**: The level set function for all the nodes might give a positive value
- **Solution**: In order to see two different fluids, there is at least one node that the level set function gives a negative value and one condition could be $h \leq R$
- **Problem**: The conservative level set function might be too steep so that it can cause oscillations
- **Solution**: The function can be formulated as $\phi(\mathbf{x}) = \tanh((d(\mathbf{x}) - R)/\epsilon h)$, where ϵ is an adjustment for the steepness of the function

Level Set Methods

Discretizations: Resolution Problem (only for the conservative level set function)

Example:





. $\epsilon = 1$ $\epsilon = 2$

Conservative level set function in 2D in contour

Level Set Methods

Discretizations: Necessity of Reinitialization

- **Problem**: The shape of the *interface* deforms in the numerical solution.

Level Set Methods

Discretizations: Necessity of Reinitialization

- **Problem**: The shape of the *interface* deforms in the numerical solution.
- **Solution**: Additional numerical method to fix: Reinitialization

Level Set Methods

Discretizations: Necessity of Reinitialization

- **Problem**: The shape of the *interface* deforms in the numerical solution.
- **Solution**: Additional numerical method to fix: Reinitialization
  - The reinitialization for the both level set functions are discretized in FEM with linear hat function and backward Euler method (FDM)

Level Set Methods

Discretizations: Necessity of Reinitialization

- **Problem**: The shape of the *interface* deforms in the numerical solution.
- **Solution**: Additional numerical method to fix: Reinitialization
  - The reinitialization for the both level set functions are discretized in FEM with linear hat function and backward Euler method (FDM)
  - Since it is an additional method, this method is iterated **twice** in each movement

Level Set Methods

Discretizations: Full Discretizations

- Full discretization for the convection equation:

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

where SD is the streamline-diffusion matrix.

Level Set Methods

Discretizations: Full Discretizations

- Full discretization for the convection equation:

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

  where SD is the streamline-diffusion matrix.

- Full discretization for the reinitialization for the standard level set function:

$$M\xi^{n+1} = M\xi^{n} + \Delta\tau(S_{\varepsilon}(\phi_0)(1 - |\nabla\tilde{\phi}|))$$

  where $\Delta\tau$ is pseudo-time step, $\tilde{\phi}$ is the previous solution.

Level Set Methods

Discretizations: Full Discretizations

- Full discretization for the convection equation:

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

  where SD is the streamline-diffusion matrix.
- Full discretization for the reinitialization for the standard level set function:

$$M\xi^{n+1} = M\xi^{n} + \Delta\tau(S_{\varepsilon}(\phi_0)(1 - |\nabla\tilde{\phi}|))$$

  where $\Delta\tau$ is pseduo-time step, $\tilde{\phi}$ is the previous solution.
- Full discretization for the reinitialization for the conservative level set function:

$$(M + \Delta\tau\varepsilon A)\xi^{n+1} = M\xi^{n} + \Delta\tau(2\tilde{\phi}|\nabla\tilde{\phi}| - (1 - \tilde{\phi}^2)\frac{\Delta\tilde{\phi}}{|\nabla\tilde{\phi}|})$$

  where ε is diffusion parameter.

Level Set Methods

Discretizations: A new problem: Parameters

Number of parameters increases to 6 from 2, which makes it difficult to control and obtain the best result

Name	**Symbol**	**Feature and Dependency**
Mesh Size	h	The smaller h, the higher resolution
Time Step	Δt	Proportional to the number of time steps for a fixed time in the simulation
Stability Parameter Constant	c	Stabilization for the convection equation using the FEM, it does not depend on h, but the stability parameter δ does
Pseudo-Time Step	$\Delta\tau$	Pseudo time step for the reinitialization, depends on Δt
Number of Reinitialization Iterations	$iter$	Adjustment to the reinitialization, mainly depends on $\Delta\tau$ but may also depend on Δt
Diffusion Parameter	ε	Smoother for the numerical solution of the conservative level set function, may depend on h

Level Set Methods

Initial and Boundary Conditions

- *Initial Condition*: Initial level set function value with the centre of the circle, $\mathbf{x}_c$, and the radius, R, in 2D to give a unique solution

Level Set Methods

Initial and Boundary Conditions

- *Initial Condition*: Initial level set function value with the centre of the circle, $\mathbf{x}_c$, and the radius, R, in 2D to give a unique solution
- *Boundary Conditions*: No external forces

Level Set Methods

Initial and Boundary Conditions

- *Initial Condition*: Initial level set function value with the centre of the circle, $\mathbf{x}_c$, and the radius, R, in 2D to give a unique solution
- *Boundary Conditions*: No external forces
-

Level Set Function	Standard	Conservative
$\mathbf{x}_c$	$(-0.5, -0.5)^T$	$(-0.5, -0.5)^T$
R	0.1	0.1
BCs	Fixed, moving with $\mathbf{x}_c$	Completely fixed

Table of Contents

Numerical Experiments and Results: Experiment Case

What is Experiment Case?

We want to test how well the level set methods work. For this:

- We must know the analytical solution when the simulation finishes at time T

Numerical Experiments and Results: Experiment Case

What is Experiment Case?

We want to test how well the level set methods work. For this:

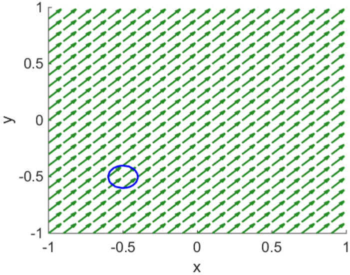
- We must know the analytical solution when the simulation finishes at time T
- The fluid vector $\mathbf{u}$ must be constant and uniform

Numerical Experiments and Results: Experiment Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.

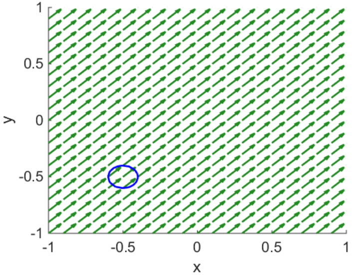


Numerical Experiments and Results: Experiment Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.
- The fluid vector, $\mathbf{u}(\mathbf{x}) = \begin{pmatrix} u_x(\mathbf{x}) \\ u_y(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 0.05 \\ 0.05 \end{pmatrix}$

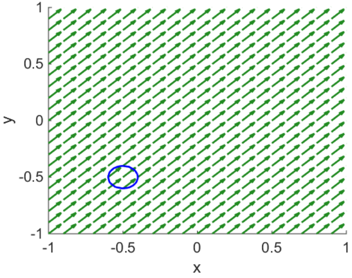


Numerical Experiments and Results: Experiment Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.
- The fluid vector, $\mathbf{u}(\mathbf{x}) = \begin{pmatrix} u_x(\mathbf{x}) \\ u_y(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 0.05 \\ 0.05 \end{pmatrix}$
- The circle is at the centre $\mathbf{x}_c = (-0.5, -0.5)^T$ with $R = 0.1$



Numerical Experiments and Results: Experiment Case

Setting up the numerical experiment

Parameters for the level set methods:

Level Set Method	Standard	Conservative
h	varies as 0.1, 0.05, and 0.025	varies as 0.1, 0.05, and 0.025
Δt	0.01	0.01
c	0.5	0.5
$\Delta\tau$	$0.01\Delta t$	$0.001\Delta t$
$iter$	2	2
ε	Not applicable	0.001

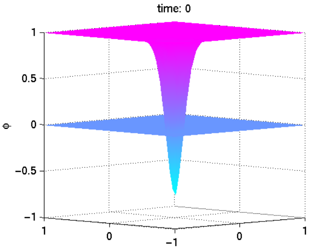
Simulation stops when $T = 1.00$

Numerical Experiments and Results: Experiment Case

Error Measurement

We consider two different types of error:

- The level set function error

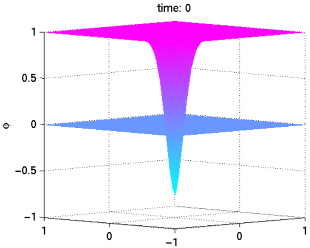


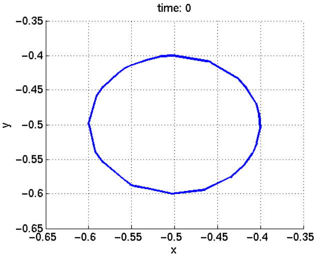
Numerical Experiments and Results: Experiment Case

Error Measurement

We consider two different types of error:

- The level set function error
- The area error (more important)





Numerical Experiments and Results: Experiment Case

Results

We have identified three different effects:

1. Effect of the stability constant parameter c

Numerical Experiments and Results: Experiment Case

Results

We have identified three different effects:

1. Effect of the stability constant parameter c
2. Effect of the position of the interface

Numerical Experiments and Results: Experiment Case

Results

We have identified three different effects:

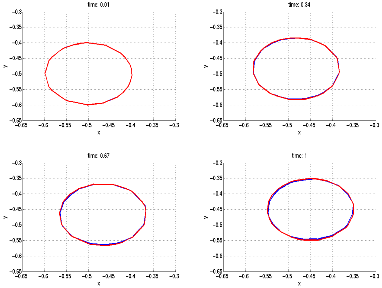
1. Effect of the stability constant parameter c
2. Effect of the position of the interface
3. Effect of the reinitialization

Numerical Experiments and Results: Experiment Case

Results : Effect of the stability constant parameter c

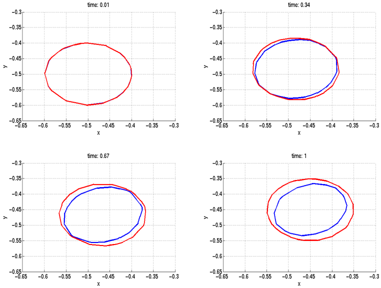
The ODE (semi-discretized) form of the new convection equation:

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



Zero contours for $c = 0.05$

$\lambda_{max} = +0.64822$



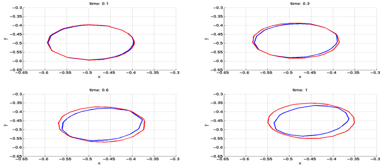
Zero contours for $c = 0.5$

$\lambda_{max} = +1.3641 \times 10^{-15}$

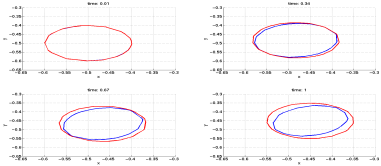
The plots are the standard level set function when $h = 0.05$ without reinitialization

Numerical Experiments and Results: Experiment Case

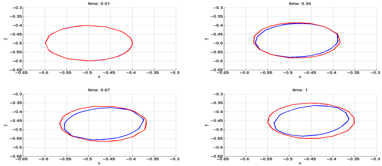
Results : Effect of the position of the interface



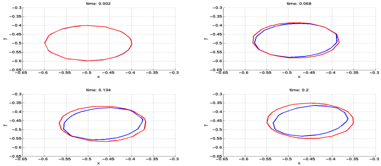
$\mathbf{u} = (0.05, 0.05)^T$, $\Delta t = 0.1$, and $T = 1.0$



$\mathbf{u} = (0.05, 0.05)^T$, $\Delta t = 0.01$, and $T = 1.00$



$\mathbf{u} = (0.05, 0.05)^T$, $\Delta t = 0.001$, and $T = 1.000$



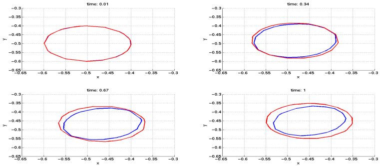
$\mathbf{u} = (0.25, 0.25)^T$, $\Delta t = 0.001$, and $T = 0.200$

The plots are the conservative level set function when $h = 0.05$ without reinitialization

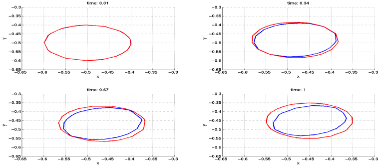
Numerical Experiments and Results: Experiment Case

Results : Effect of the reinitialization

Different results for the different level set function for $h = 0.05$



Standard level set function WITHOUT reinitialization

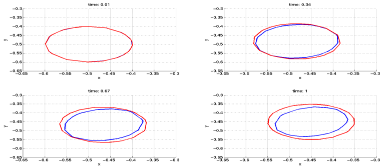


Standard level set function WITH reinitialization

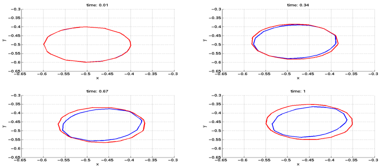
Numerical Experiments and Results: Experiment Case

Results : Effect of the reinitialization

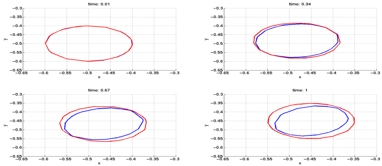
Different results for the different level set function for $h = 0.05$



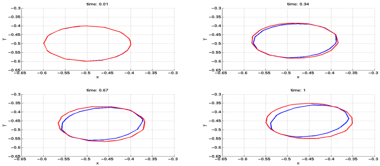
Standard level set function WITHOUT reinitialization



Conservative level set function WITHOUT reinitialization



Standard level set function WITH reinitialization

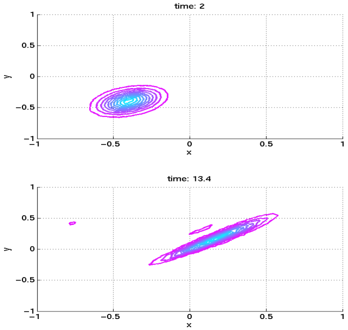


Conservative level set function WITH reinitialization

Numerical Experiments and Results: Experiment Case

Results : Effect of the Reinitialization

The reinitialization acts as an exerted force on the level set function. And indeed, new arbitrary contours occurs after a long time in the conservative level set method for $c = 0.5$, which is a sign for **instability**.

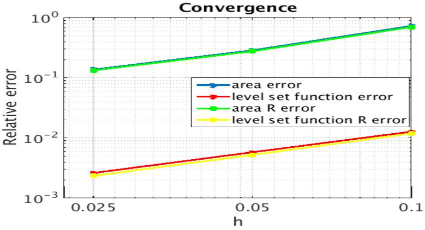


The plots are the conservative level set function when $h = 0.05$

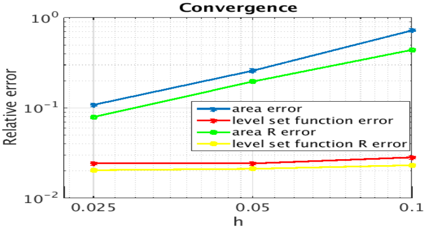
Numerical Experiments and Results: Experiment Case

Results : Grid Convergence and Convergence Rates

Here Δt is chosen such as $\Delta t / h = 0.1$



Standard level set method



Conservative level set method

Level set function	Standard		Conservative	
Reinitialization included?	NO	YES	NO	YES
Relative level set function error	1.1436	1.1718	0.1086	0.1039
Relative area error	1.2065	1.1973	1.3701	1.2362

Simulation terminates when $T = 1.00$

Table of Contents

Fluid Mechanics

Why to use Fluid Mechanics?

- We want to test the level set methods on a more complicated system, which is called *benchmark case*

Fluid Mechanics

Why to use Fluid Mechanics?

- We want to test the level set methods on a more complicated system, which is called *benchmark case*
- We find the numerical solution for the fluid velocity, $\mathbf{u}$

Fluid Mechanics

Why to use Fluid Mechanics?

- We want to test the level set methods on a more complicated system, which is called *benchmark case*
- We find the numerical solution for the fluid velocity, $\mathbf{u}$
- The solution follows strictly as in Larson and Bengzon's book: *The Finite Element Method, Theory, Implementation, and Applications*

Fluid Mechanics

Mathematical Model

- Assuming the conservation of mass and momentum, incompressible Newtonian fluids, we have the famous Navier-Stokes equation

Fluid Mechanics

Mathematical Model

- Assuming the conservation of mass and momentum, incompressible Newtonian fluids, we have the famous Navier-Stokes equation
- 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 ($\nu = 1$):

$$\Delta \mathbf{u} + \nabla p = \mathbf{f}, \quad \text{in } \Omega$$
$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega$$
$$\mathbf{u} = g_D, \text{ on } \partial\Omega$$

Fluid Mechanics

Mathematical Model

- Assuming the conservation of mass and momentum, incompressible Newtonian fluids, we have the famous Navier-Stokes equation
- 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 ($\nu = 1$):

$$\Delta \mathbf{u} + \nabla p = \mathbf{f}, \quad \text{in } \Omega$$
$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega$$
$$\mathbf{u} = g_D, \text{ on } \partial\Omega$$

- Note that since there is only ∇p in the equations, the pressure needs to have a zero mean value (the hydrostatic pressure level):

$$\frac{\int\limits_{\Omega} p \, \mathrm{d}\Omega}{|\Omega|} = 0$$

Fluid Mechanics

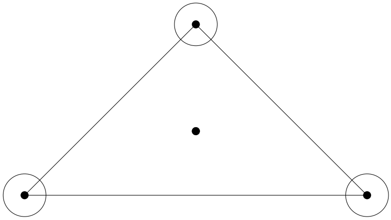
Discretization

- Since we need to solve for the velocity $\mathbf{u}$ and pressure p, we create two different test functions in Hilbert and L^2-norm spaces, respectively

Fluid Mechanics

Discretization

- Since we need to solve for the velocity $\mathbf{u}$ and pressure p, we create two different test functions in Hilbert and L^2-norm spaces, respectively
- Moreover, the MINI element provides the pair solution $(\mathbf{u}, p)$ in solving the resulting form of FEM from the weak forms due to being the simplest inf-sup stable element and easy to implement.



A triangulation of MINI element in 2D where • represents velocity and ◯ represents pressure at the given nodes

Fluid Mechanics

Discretization

The final discretization is:

$$\begin{bmatrix} A & 0 & B_x & 0 \\ 0 & A & B_y & 0 \\ B_x^T & B_y^T & 0 & a \\ 0 & 0 & a^T & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \\ \mu \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \\ 0 \\ 0 \end{bmatrix}$$

where A is the stiffness matrix, B_x and B_y are divergence matrices, a is the operator from the hydrostatic pressure level, b_x and b_y are the load vectors, μ is a Lagrangian multiplier and the numerical velocity $\mathbf{u} = (u_x, u_y)^T$ (taken) and numerical pressure p (disregarded)

Table of Contents

Numerical Experiments and Results: Benchmark Case

What is Benchmark Case?

We want to test how well the level set methods work *in a complicated system*. For this:

- We do not know the analytical solution (for the level set function) when the simulation finishes at time T

Numerical Experiments and Results: Benchmark Case

What is Benchmark Case?

We want to test how well the level set methods work *in a complicated system*. For this:

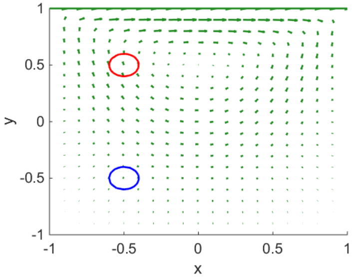
- We do not know the analytical solution (for the level set function) when the simulation finishes at time T
- The fluid vector $\mathbf{u}$ is already complicated from the fluid mechanics

Numerical Experiments and Results: Benchmark Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.

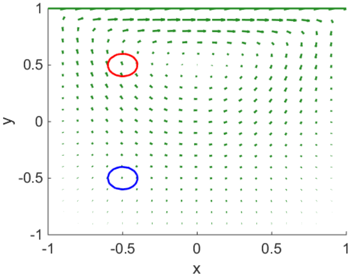


Numerical Experiments and Results: Benchmark Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.
- The fluid vector, $\mathbf{u}$, is the numerical result from fluid mechanics when the top lid has $\mathbf{u}_T = (1, 0)^T$, whereas the other boundaries are at rest

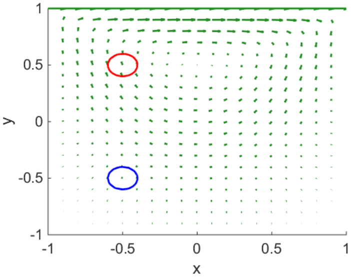


Numerical Experiments and Results: Benchmark Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.
- The fluid vector, $\mathbf{u}$, is the numerical result from fluid mechanics when the top lid has $\mathbf{u}_T = (1, 0)^T$, whereas the other boundaries are at rest
- Since the velocity field varies, we consider two different initial positions of the interface

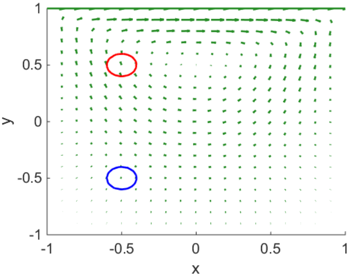


Numerical Experiments and Results: Benchmark Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.
- The fluid vector, $\mathbf{u}$, is the numerical result from fluid mechanics when the top lid has $\mathbf{u}_T = (1, 0)^T$, whereas the other boundaries are at rest
- Since the velocity field varies, we consider two different initial positions of the interface
    - **Upper Position**: The circle is at the centre $\mathbf{x}_c = (-0.5, 0.5)^T$ with $R = 0.1$ (red)

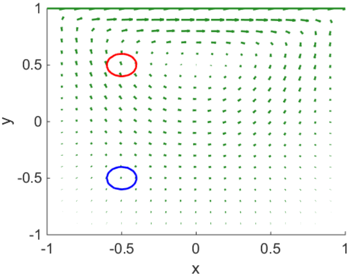


Numerical Experiments and Results: Benchmark Case

Setting up the numerical experiment

Experiment values:

- The grid domain is $[-1, 1] \times [-1, 1]$.
- The fluid vector, $\mathbf{u}$, is the numerical result from fluid mechanics when the top lid has $\mathbf{u}_T = (1, 0)^T$, whereas the other boundaries are at rest
- Since the velocity field varies, we consider two different initial positions of the interface
    - **Upper Position**: The circle is at the centre $\mathbf{x}_c = (-0.5, 0.5)^T$ with $R = 0.1$ (red)
    - **Lower Position**: The circle is at the centre $\mathbf{x}_c = (-0.5, -0.5)^T$ with $R = 0.1$ (blue)



Numerical Experiments and Results: Benchmark Case

Setting up the numerical experiment

Parameters for the level set methods:

	Benchmark case	
Level Set Method	Standard	Conservative
h	varies as 0.1, 0.05, and 0.025	varies as 0.1, 0.05, and 0.025
Δt	0.01	0.01
c	1.0	1.0
$\Delta \tau$	$0.01\Delta t$	$0.001\Delta t$
iter	2	2
ε	Not applicable	0.001

Simulation stops when $T = 1.00$

Numerical Experiments and Results: Benchmark Case

Error Measurement

We consider only one type of error

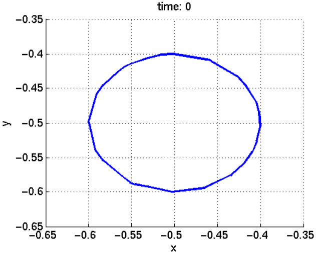
- Since we do not know the analytical solution of the level set function at time T, there is no level set function error

Numerical Experiments and Results: Benchmark Case

Error Measurement

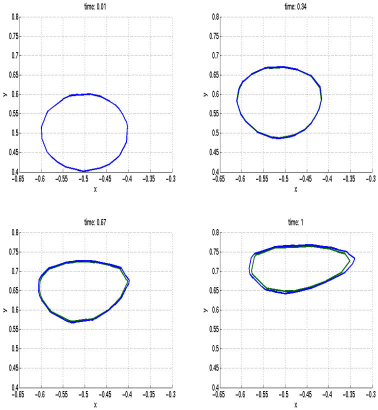
We consider only one type of error

- Since we do not know the analytical solution of the level set function at time T, there is no level set function error
- We know the analytical area from the initial position, πR^2, so we can calculate the area error (more important)



Numerical Experiments and Results: Benchmark Case

Results: Upper Position

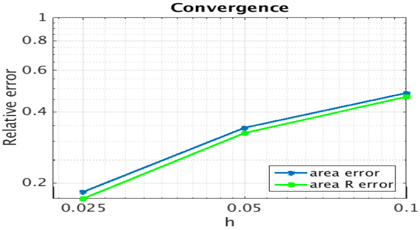


The plots are the conservative level set function when $h = 0.05$

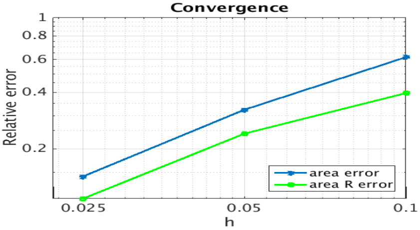
Numerical Experiments and Results: Benchmark Case

Results: Upper Position : Grid Convergence and Convergence Rates

Here Δt is chosen such as $\Delta t/h = 0.1$



Standard level set method



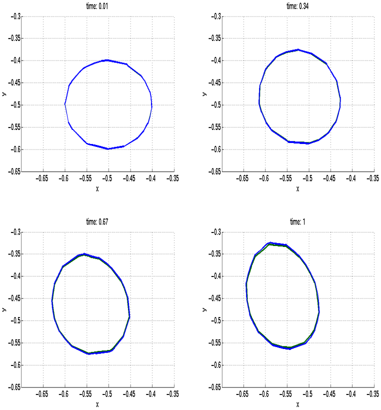
Conservative level set method

Level set function	Standard		Conservative	
Reinitialization included?	NO	YES	NO	YES
Relative area error	0.6951	0.7152	1.0557	0.9343

Simulation terminates when $T = 1.00$

Numerical Experiments and Results: Benchmark Case

Results: Lower Position

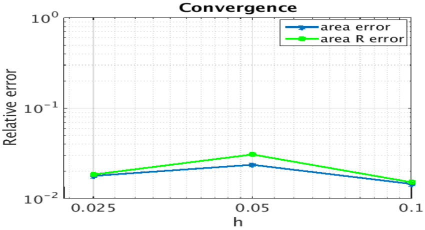


The plots are the conservative level set function when $h = 0.05$

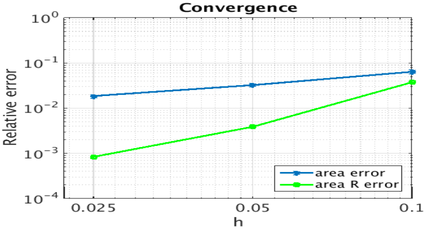
Numerical Experiments and Results: Benchmark Case

Results: Lower Position : Grid Convergence and Convergence Rates

Here Δt is chosen such as $\Delta t/h = 0.1$



Standard level set method



Conservative level set method

Level set function	Standard		Conservative	
Reinitialization included?	NO	YES	NO	YES
Relative area error	-0.1478	-0.1418	0.8893	2.7398

Simulation terminates when $T = 1.00$

Table of Contents

Conclusion

Comments on stability

- The stability has to be measured carefully as it has imbalance between space stability and time stability as well as with the reinitialization

Conclusion

Comments on stability

- The stability has to be measured carefully as it has imbalance between space stability and time stability as well as with the reinitialization
- The value c significantly increases when we use it in the benchmark case comparing to the one in the experiment case

Conclusion

Comments on reinitializations

- 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

Conclusion

Comments on reinitializations

- 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
  - The reinitialization in the standard level set method tries to preserve as much as it can

Conclusion

Comments on reinitializations

- 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
  - The reinitialization in the standard level set method tries to preserve as much as it can
  - The reinitialization in the conservative level set method expands the interface

Conclusion

Comments in general

- The level set methods depend on the stability parameter or the GLS method very much in order to obtain the best solution

Conclusion

Comments 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:

Conclusion

Comments 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:
    - The conservative level set function is better than the standard level function

Conclusion

Comments 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:
  - The conservative level set function is better than the standard level function
  - The reinitialization methods give a better solution under certain conditions

Conclusion

Comments 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:
  - The conservative level set function is better than the standard level function
  - 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

Conclusion

Further Research

- Another FEM stabilization method, which is other than GLS method, in order to solve the stabilization problem in 2D (in the next slide)

Conclusion

Further Research

- Another FEM stabilization method, which is other than GLS method, in order to solve the stabilization problem in 2D (in the next slide)
- Different kinds of level set methods in order to find one that affects the solution in a much better way

Table of Contents

More on the Research

New Method : Residual based artificial viscosity method

- During finalising the report, 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

More on the Research

New Method : Residual based artificial viscosity method

- During finalising the report, 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 tried the new method called "Residual based artificial viscosity method" from the article: *Convergence of a residual based artificial viscosity finite element method* by M. Nazarov

More on the Research

New Method : Residual based artificial viscosity method

- During finalising the report, 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 tried the new method called "Residual based artificial viscosity method" from the article: *Convergence of a residual based artificial viscosity finite element method* by M. Nazarov
- The changes are:

More on the Research

New Method : Residual based artificial viscosity method

- During finalising the report, 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 tried the new method called "Residual based artificial viscosity method" from the article: *Convergence of a residual based artificial viscosity finite element method* by M. Nazarov
- The changes are:
    - $M\dot{\xi} + C\xi + \delta SD\xi = 0 \quad \rightarrow \quad M\dot{\xi} + C\xi + \delta A\xi = 0$

More on the Research

New Method : Residual based artificial viscosity method

- During finalising the report, 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 tried the new method called "Residual based artificial viscosity method" from the article: *Convergence of a residual based artificial viscosity finite element method* by M. Nazarov
- The changes are:
  - $M\dot{\xi} + C\xi + \delta SD\xi = 0 \quad \rightarrow \quad M\dot{\xi} + C\xi + \delta A\xi = 0$
  - $\delta = ch/\|\mathbf{u}\|_{L^\infty(\Omega)} \quad \rightarrow \quad \delta = 2ch\|\mathbf{u}\|_{L^\infty(\Omega)}$

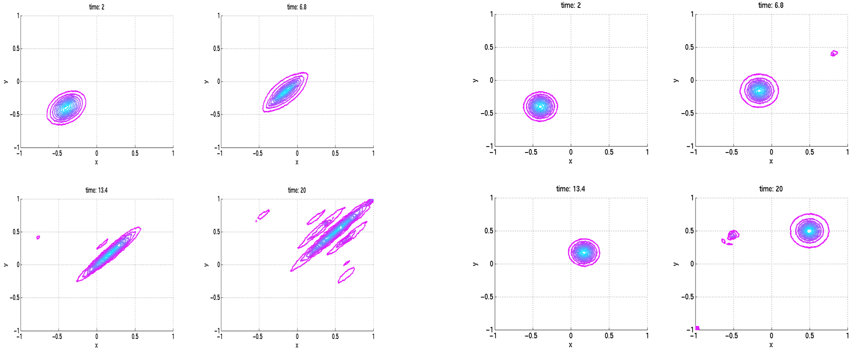
More on the Research

New Method : Residual based artificial viscosity method

- During finalising the report, 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 tried the new method called "Residual based artificial viscosity method" from the article: *Convergence of a residual based artificial viscosity finite element method* by M. Nazarov
- The changes are:
  - $M\dot{\xi} + C\xi + \delta SD\xi = 0 \quad \rightarrow \quad M\dot{\xi} + C\xi + \delta A\xi = 0$
  - $\delta = ch/\|\mathbf{u}\|_{L^\infty(\Omega)} \quad \rightarrow \quad \delta = 2ch\|\mathbf{u}\|_{L^\infty(\Omega)}$
  - $c = 0.5 \quad \rightarrow \quad c = 0.03$

More on the Research

New Method : Residual based artificial viscosity method : Results



GLS method

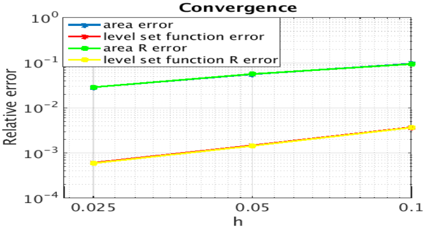
.

Residual based artificial viscosity method

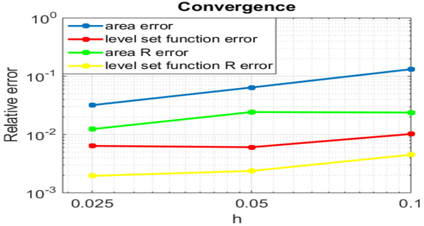
More on the Research

New Method : Residual based artificial viscosity method : Results

Here Δt is chosen such as $\Delta t/h = 0.1$



Standard level set method



Conservative level set method

Level set function	Standard		Conservative	
Reinitialization included?	NO	YES	NO	YES
Relative level set function error	1.3063	1.3158	0.3387	0.6027
Relative area error	0.8645	0.8531	1.0234	0.4744

Simulation terminates when $T = 1.00$

Questions?