

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Anti-diffusion method for interface steepening in two-phase incompressible flow

K. K. So, X. Y. Hu and N. A. Adams

Lehrstuhl für Aerodynamik, Technische Universität München, 85748 Garching, Germany

Abstract

In this paper, we present a method for obtaining sharp interfaces in two-phase incompressible flow by an anti-diffusion correction. The underlying discretization is based on the volume-of-fluid (VOF) interface-capturing method. The key idea is to steepen the interface by solving the diffusion equation with reverse time, i.e. an anti-diffusion equation, after each advection step of the volume fraction. As a solution of the anti-diffusion equation requires regularization, a limiter based on the directional derivative is developed for calculating the gradient of the volume fraction. This limiter ensures the boundedness of the volume fraction. The formulation of the limiter and the algorithm for solving the anti-diffusion equation are suitable for 3-dimensional unstructured meshes. Validation computations are performed for 2- and 3- dimensional rising-bubble and rising-drop configurations, and for Cartesian and non-Cartesian meshes. The results demonstrate that sharp interfaces can be recovered reliably and that the results agree with previous simulations based on different interface methods and with experiments.

Key words: two-phase flow, volume-of-fluid, interface steepening, anti-diffusion, unstructured meshes

1. Introduction

The numerical simulation of two-phase incompressible flows is an area of high interest in academia and industry. Various approaches have been developed. Two

Preprint submitted to JCP

December 3, 2009

1
2
3 main directions are interface-tracking methods and interface-capturing methods.

4 With interface-tracking methods, the location of the interface is explicitly rep-
5 resented. Examples of interface-tracking methods include front tracking methods
6 [1, 2] and marker methods [3, 4]. These latter, e.g., can effectively locate the inter-
7 face position by interface markers. However, they encounter difficulties for large
8 interface deformations and topology changes, and require a special treatment of
9 the interface marker distribution when the interface is stretched or compressed.

10 On the other hand, interface-capturing methods do not explicitly track the loca-
11 tion of the interface, but capture the location of the interface implicitly. Examples
12 of interface-capturing methods include the level-set method and the volume-of-
13 fluid (VOF) method. With the level-set method the interface is defined as the
14 zero contour of a signed-distance function - the level-set function. The interface
15 is sharp by definition and sharpness is maintained by recovering the signed dis-
16 tance property of the level-set function through reinitialization. From the level-set
17 function the curvature of the interface and the surface tension can be calculated
18 with high accuracy. However, a main drawback of the level-set method is lack of
19 discrete conservation. We refer to references [5] and [6] for a detailed description
20 of the level-set method.

21 With VOF methods, the two phases are defined by the volume fraction which
22 assumes values between 0 and 1. The interface is represented by the transition
23 region where the volume fraction ramps up from 0 to 1. The main advantage of
24 VOF methods is the exact conservation of mass. One main drawback of VOF
25 methods is that the interface cannot be located precisely, which leads to inaccura-
26 cies in calculating interface curvature and thus surface tension. References [7] and
27 [4] provide an overall review of VOF methods.

28 To obtain a sharp interface for VOF methods, two methodologies are generally
29 used. First, with VOF volume-tracking methods, the interface is reconstructed

1
2
3 before each advection step, and subsequently the flow is updated by propagat-
4 ing the reconstructed interface. Different interface-reconstruction schemes have
5 been developed, the basic ones include simple line interface calculation (SLIC)
6 [8], SOLA-VOF [9] and piecewise-linear interface construction (PLIC) [4]. For the
7 interface-propagation step the operator split method is generally used [4]. Draw-
8 backs of the VOF volume-tracking method are: the curvature of the reconstructed
9 interface is not smooth, which leads to inaccuracies in the interface propagation
10 step; the interface propagation can become unstable for very complex interfaces.
11 Unphysical flows, termed “flotsam” and “jetsam”, can be created due to the errors
12 induced by the volume-tracking algorithm [7]. Second, instead of reconstructing
13 and propagating the interface as with the VOF volume-tracking method, in VOF
14 volume-capturing methods the volume fraction is advected with a special treat-
15 ment to reduce the numerical diffusion. Examples include the introduction of an
16 artificial-compression term in the advection equation [10].

17
18 For all methods, one prime criterion of accurate two-phase flow simulation is to
19 maintain a sufficiently sharp interface throughout the simulation. How to obtain a
20 sharp interface with interface-capturing methods has been studied by a number of
21 researchers in the past. Examples include the usage of a compressive scheme which
22 blends an upwind differencing scheme and a higher-order differencing scheme for
23 the advection step [11], limited downwind anti-diffusive flux correction [12], and
24 the use of artificial compression as an intermediate step [13]. In this paper, we
25 propose a regularized anti-diffusion correction technique which can be used in a
26 straight-forward fashion with underlying VOF discretization schemes on regular or
27 irregular meshes. We demonstrate that by this correction technique, applied after
28 the VOF advection step, a desired interface sharpness can be recovered reliably
29 and efficiently.

30 For interface steepening methods it is possible to impose a parameter to define

1
2
3 the desired interface thickness for obtaining convergence and stability, for example
4 in [13, 14] the interface thickness is related to the grid size through a parameter d .
5
6 However, determination of parameters defining the interface thickness is empirical.
7
8 Alternatively, as is the case for our anti-diffusion-correction algorithm presented
9
10 in this paper, a stopping criterion for the correction step can be developed. The
11
12 stopping criterion should be normalized such that it is case- and grid-resolution-
13
14 independent.

15
16 The governing equations and an overview of the solution procedure are pre-
17
18 sented in Section 2. The main focus of the paper, which is the formulation of the
19
20 anti-diffusion equation and the solution algorithm, is presented in Section 3. Nu-
21
22 merical cases and results are presented in Section 4. Finally, concluding remarks
23
24 are given in Section 5.

25 26 27 **2. Governing equations**

28
29 In this paper, the open source CFD package, *OpenFOAM* [15] is employed as
30
31 the simulation platform. This procedure underlines the fact that the proposed anti-
32
33 diffusion correction can be used in a straight-forward fashion with any underlying
34
35 VOF discretization scheme. *OpenFOAM* is a finite-volume package for solving
36
37 partial differential equations on 3-dimensional unstructured meshes. The solver
38
39 *interFoam* in the *OpenFOAM* package is designated for solving two-phase flow
40
41 using a VOF volume-capturing method and is utilized as the basis for developing
42
43 the anti-diffusion correction method in this paper. The main part of the solution
44
45 procedure of the solver is described by Rusche [10]. The governing equations
46
47 for unsteady, incompressible, viscous, immiscible two-phase flow are given by the
48
49 continuity equation and the Navier-Stokes equation

$$50
51
52
53 \nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot \mathbf{T} + \rho \mathbf{g} + \mathbf{f}_\sigma \quad , \quad (2)$$

where \mathbf{u} is the velocity, ρ is the density, t is the time, p is the pressure, \mathbf{T} is the stress tensor, \mathbf{g} is the gravitational acceleration, and \mathbf{f}_σ is the force due to surface tension. For a two-phase flow, the two fluids are represented by the volume fraction α which is defined as

$$0 \leq \alpha \leq 1 \quad , \quad (3)$$

where $\alpha = 0$ refers to the first fluid, $\alpha = 1$ refers to the second fluid, and $0 < \alpha < 1$ refers to the transitional region, i.e. the interface between the two fluids. The volume fraction is advected by the flow, resulting in the volume-fraction-transport equation

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) = 0 \quad . \quad (4)$$

The local density ρ and viscosity μ are given by

$$\rho = \alpha \rho_1 + (1 - \alpha) \rho_2 \quad (5)$$

$$\mu = \alpha \mu_1 + (1 - \alpha) \mu_2 \quad , \quad (6)$$

where the subscripts 1 and 2 denote the respective fluids of the two-phase flow. The stress tensor is given by Newton's law

$$\mathbf{T} = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad . \quad (7)$$

The surface curvature of the interface κ is given by

$$\kappa = \nabla \cdot \left(\frac{\nabla \alpha}{|\nabla \alpha|} \right) \quad . \quad (8)$$

The force due to surface tension is formulated by the Continuum Surface Force (CSF) as given by Brackbill et al. [16]

$$\mathbf{f}_\sigma = -\sigma \kappa \nabla \alpha \quad , \quad (9)$$

where σ is the surface tension.

The above governing equations define the continuity of mass and momentum of the incompressible two-phase flow. For obtaining a sharp phase-interface, a new element of the solution algorithm - the anti-diffusion correction - is introduced and defined by

$$\frac{\partial \alpha}{\partial \tau} = \nabla \cdot (D \nabla \alpha) \quad , \quad (10)$$

where D is the diffusion coefficient and τ is a pseudo time for evolving the anti-diffusion correction. In the diffusion equation (10), positive D represents a normal diffusion process, while negative D represents a diffusion process with reverse time, which can be regarded as an anti-diffusion process.

The discretization of the advection equation and momentum equation follows the underlying *OpenFOAM* algorithm as given in [10]. First, the advection equation (4) is evaluated for one time step based on the TVD limiter by Jasak et al. [17]. Subsequently, the new part, the anti-diffusion-correction, equation (10) is solved after each time step of the advection equation (4). The density and viscosity updates are computed from the new volume-fraction field. Subsequently, the momentum and continuity equations are solved by the PISO algorithm by Issa [18], as available in *OpenFOAM*.

The main focus of this paper is the anti-diffusion correction after the advection

step, and the rest of the paper is devoted to the detailed description of the anti-diffusion correction. For a comprehensive description of the other details of the solution procedure as provided by *OpenFOAM* the reader is referred to references [11, 10, 15, 17, 18, 19].

3. Anti-diffusion correction

3.1. Formulation of the 1-dimensional anti-diffusion equation

Before formulating the anti-diffusion correction for 3-dimensional flow simulations, the essential properties of the anti-diffusion correction are illustrated for 1-dimensional setting. The key idea is to sharpen the interface by applying anti-diffusion to the interface which has been smeared out due to numerical diffusion during the volume-fraction advection step. For this purpose the diffusion equation (10) is solved with a negative diffusion coefficient D . The diffusion equation (10) can be reformulated for the diffusion coefficient D set to -1, as anti-diffusion process with constant diffusion coefficient

$$\frac{\partial \alpha}{\partial \tau} = \nabla \cdot (-\nabla \alpha) \quad . \quad (11)$$

The anti-diffusion equation is ill-posed, therefore an approximate numerical solution requires regularization. A solution to an anti-diffusion equation was first proposed by Boris and Book, where the flux corrected transport (FCT) algorithm is adopted [20]. Different methods attempting to solve the anti-diffusion equation in a stabilized way by formulating a non-linear diffusion coefficient have been reported and applied in the field of image processing, e.g. “stabilized inverse diffusion equations” [21] and “forward-and-backward adaptive diffusion process” [22]. On the other hand, monotonicity preserving constraints for ensuring boundedness were studied in references [23, 24]. Imposing maximum and minimum bounds by

1
2
3 considering the solution in the neighboring cells for ensuring boundedness on un-
4 structured meshes was proposed by Ubbink and Issa [11]. The usage of a *minmod*
5 function for a discrete inverse diffusion for filtering purposes in image processing
6 was proposed by Osher and Rudin [25]. An analysis of the stabilized inverse dif-
7 fusion based on a *minmod* function was carried out by Breuß et al. [26, 27]. In
8 this paper we propose a discrete anti-diffusion equation regularized by a *minmod*
9 function.

10 Below we will first illustrate the 1-dimensional discretization procedure based
11 on a *minmod* function and then propose a modification which is suitable for a later
12 extension to multiple dimensions and unstructured meshes.
13
14

15 3.1.1. Regularized anti-diffusion equation by *minmod* function

16 Considering a 1-dimensional equidistantly spaced grid, the anti-diffusion equa-
17 tion can be regularized and solved numerically by using a *minmod* function as
18 given in references [26, 27] for the numerical flux calculation
19
20

$$21 F_{i-1/2} = \text{minmod} \left(\frac{\alpha_{i+1} - \alpha_i}{\Delta x}, \frac{\alpha_i - \alpha_{i-1}}{\Delta x}, \frac{\alpha_{i-1} - \alpha_{i-2}}{\Delta x} \right) , \quad (12)$$

22 where i indicates the cell under consideration, $F_{i-1/2}$ is the numerical flux
23 between the cell $i - 1$ and the cell i , and
24
25

$$26 \text{minmod} (a, b, c) = \text{sgn} (b) \max (0, \min (\text{sgn} (b) a, |b|, \text{sgn} (b) c)) . \quad (13)$$

27 The calculation of $F_{i-1/2}$ requires α_{i+1} , α_i , α_{i-1} and α_{i-2} . However, the algo-
28 rithm cannot be applied directly for 3-dimensional and irregular meshes as α_{i+1}
29 and α_{i-2} are not well defined with respect to the cell face between cell i and cell
30 $i - 1$. For this reason we propose a compact re-formulation of eq. (12) as follows:
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1
2
3 *Modification 1.* We decompose the numerical flux calculation by eq. (12) into the
4 following two steps:
5

- 6 1. $\nabla\alpha$ at cell center is calculated based on a *minmod* function.
- 7 2. The numerical flux is calculated based on selecting $\nabla\alpha$ with minimum mag-
8 nitude.
9

10
11
12 *Step 1.* $(\nabla\alpha)_i$ at the cell center is calculated as
13

$$14 \quad (\nabla\alpha)_i = \text{minmod} \left(\frac{\alpha_{i+1} - \alpha_i}{\Delta x}, \frac{\alpha_i - \alpha_{i-1}}{\Delta x} \right) \quad , \quad (14)$$

15
16 where

$$17 \quad \text{minmod}(a, b) = \begin{cases} a & \text{if } a \cdot b > 0 \text{ and } |a| \leq |b| \\ b & \text{if } a \cdot b > 0 \text{ and } |b| \leq |a| \\ 0 & \text{else} \end{cases} \quad . \quad (15)$$

18
19
20
21
22 *Step 2.* $F_{i-1/2}$ is calculated as

$$23 \quad F_{i-1/2} = \begin{cases} (\nabla\alpha)_i & \text{if } |(\nabla\alpha)_i| \leq |(\nabla\alpha)_{i-1}| \\ (\nabla\alpha)_{i-1} & \text{if } |(\nabla\alpha)_{i-1}| < |(\nabla\alpha)_i| \end{cases} \quad . \quad (16)$$

24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
By combining (14) and (16) it can be shown that the resulting numerical flux function is equivalent to that of (12). Note that by (14) the calculation of $(\nabla\alpha)_i$ requires only α_{i+1} , α_i and α_{i-1} , and similarly the calculation of $(\nabla\alpha)_{i-1}$ requires only α_i , α_{i-1} and α_{i-2} .

66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
Modification 2. We further propose an alternative to (14) and (15) for the calculation of $(\nabla\alpha)_i$

$$86 \quad (\nabla\alpha)_i = \frac{\alpha_{i+1/2} - \alpha_{i-1/2}}{\Delta x} \quad , \quad (17)$$

$$\alpha_{i+1/2} = \begin{cases} \alpha_{i+1} & \text{if } a \cdot \frac{a+b}{2} > 0 \text{ and } |a| < \left| \frac{a+b}{2} \right| \\ \frac{\alpha_{i+1} + \alpha_i}{2} & \text{if } a \cdot \frac{a+b}{2} > 0 \text{ and } |a| = \left| \frac{a+b}{2} \right| \\ \alpha_i & \text{else} \end{cases} \quad , \quad (18a)$$

$$\alpha_{i-1/2} = \begin{cases} \alpha_{i-1} & \text{if } b \cdot \frac{a+b}{2} > 0 \text{ and } |b| < \left| \frac{a+b}{2} \right| \\ \frac{\alpha_i + \alpha_{i-1}}{2} & \text{if } b \cdot \frac{a+b}{2} > 0 \text{ and } |b| = \left| \frac{a+b}{2} \right| \\ \alpha_i & \text{else} \end{cases} \quad , \quad (18b)$$

where $a = \frac{\alpha_{i+1} - \alpha_i}{\Delta x}$, $b = \frac{\alpha_i - \alpha_{i-1}}{\Delta x}$ and $\frac{a+b}{2} = \frac{\alpha_{i+1} - \alpha_{i-1}}{2\Delta x}$ which is the gradient evaluated by central differencing.

By comparing (14) and (15), with (17), (18a) and (18b), it can be shown that the resulting $(\nabla\alpha)_i$ are equivalent. Note that by (18a) the evaluation of $\alpha_{i+1/2}$ requires only the directional derivative $a = \frac{\alpha_{i+1} - \alpha_i}{\Delta x}$ and the gradient evaluated by central differencing $\frac{a+b}{2} = \frac{\alpha_{i+1} - \alpha_{i-1}}{2\Delta x}$. The same holds for the calculation of $\alpha_{i-1/2}$ by (18b). $(\nabla\alpha)_i$ can then be calculated from $\alpha_{i+1/2}$ and $\alpha_{i-1/2}$ by (17).

3.1.2. Time discretization

After the numerical flux is calculated, α is forwarded in pseudo time by an explicit Euler scheme as

$$\alpha_i^{n+1} = \alpha_i^n + \frac{(F_{i-1/2} - F_{i+1/2})}{\Delta x} \Delta\tau \quad , \quad (19)$$

where α_i^{n+1} and α_i^n are the volume fractions at the new and the old time step, respectively. $\Delta\tau$ is the pseudo-time step of the anti-diffusion process. Based on the stability analysis by Breuß [26] of the stabilized inverse diffusion, regularized by a *minmod* function, the time-step-size constraint for stable pseudo-time advancement of our anti-diffusion equation is derived accordingly as

$$\Delta\tau = \frac{(\Delta x)^2}{2|D|} . \quad (20)$$

The effect of the anti-diffusion correction is illustrated in Fig. 1. Initially a 1-dimensional profile, increasing monotonically from $\alpha = 0$ to $\alpha = 1$ with a transition region across 6 cells, is defined, which is an analog to the numerically diffused interface of a two-phase flow. By solving the anti-diffusion equation repeatedly, the transition region becomes thinner which represents a steepened interface. As can be seen from the result, the anti-diffusion correction exhibits the desired properties which are crucial for two-phase flows with sharp interfaces: reduction of the thickness of the transition region with respect to its central position and boundedness of the volume fraction between 0 and 1.

[Figure 1 about here.]

3.1.3. Measurement of interface sharpness

After each volume-fraction advection step, the anti-diffusion correction is performed repeatedly to attain a sharp interface. In multiple dimensions and for unstructured grids a grid-independent measure of interface sharpness is required to derive a stopping criterion for the anti-diffusion iteration. For this purpose, we use the flux difference $\nabla \cdot (-\nabla\alpha_i) = \frac{(F_{i-1/2} - F_{i+1/2})}{\Delta x}$ as a measurement of interface sharpness.

For a case- and grid-resolution-independent criterion, interface sharpness tolerances TOL_1 and TOL_2 are defined, and the term $\nabla \cdot (-\nabla\alpha_i)$ is normalized as

$$TOL_1 \leq \frac{\sum_i |\nabla \cdot (\nabla\alpha_i)|}{\sum_i |\nabla\alpha_i|^2} , \quad (21)$$

$$TOL_2 \leq \frac{\max_i (|\nabla \cdot (\nabla \alpha_i)|)}{\max_i (|\nabla \alpha_i|^2)} \quad , \quad (22)$$

where \sum_i denotes the summation over all cells, \max_i determines the maximum value from all cells, and $\nabla \alpha_i = \frac{\alpha_{i+1} - \alpha_{i-1}}{2\Delta x}$. The evolution of TOL_1 and TOL_2 corresponding to the anti-diffusion correction in Fig. 1 is shown in Fig. 2.

[Figure 2 about here.]

3.2. Formulation of the anti-diffusion equation for multiple dimensions and unstructured meshes

To extend the idea of interface steepening by anti diffusion to multiple dimensions and unstructured meshes, and to couple the anti-diffusion algorithm with volume-fraction advection and the Navier-Stokes equations for realistic flow simulations, the diffusion coefficient is related to the numerical diffusion due to advection. A modified-differential equation analysis of the numerical diffusion induced by an upwind scheme shows that the numerical diffusion coefficient is $\frac{1}{2}\mathbf{u}\Delta x (1 - \frac{\mathbf{u}\Delta t}{\Delta x})$ e.g. [28]. In our formulation of the anti-diffusion correction the aim is to counteract the numerical diffusion resulting from discrete advection. For this purpose the diffusion coefficient in the anti-diffusion equation is chosen to be $|\mathbf{u}|$, i.e. the diffusion coefficient D is set to $-|\mathbf{u}|$

$$\frac{\partial \alpha}{\partial \tau} = \nabla \cdot (-|\mathbf{u}| \nabla \alpha) \quad , \quad (23)$$

where $|\mathbf{u}|$ is constant in τ .

For multiple dimensions, the interface normal direction is to be taken into account so that the interface is sharpened in its normal direction. For this purpose, the anti-diffusion correction flux is projected onto the interface normal direction. Hence the diffusion equation (23) is reformulated as

$$\frac{\partial \alpha}{\partial \tau} = \nabla \cdot [-|\mathbf{u}| (\nabla \alpha \cdot \mathbf{n}_I) \mathbf{n}_I] \quad . \quad (24)$$

The interface unit normal \mathbf{n}_I is calculated as

$$\mathbf{n}_I = \frac{(\nabla \alpha)^*}{|(\nabla \alpha)^*|} \quad , \quad (25)$$

where $(\nabla \alpha)^*$ is calculated by the Gauss theorem from

$$(\nabla \alpha)^* = \frac{\sum_{cf} (\alpha \mathbf{S})}{V} \quad . \quad (26)$$

Here \sum_{cf} denotes the summation over all cell faces, \mathbf{S} is the cell surface area vector, V is the cell volume. α is obtained by averaging the volume fractions of cells P and N (refer to Fig. 3), i.e. $\frac{\alpha_P + \alpha_N}{2}$, where α_P and α_N are the volume fractions of cell P and N respectively.

[Figure 3 about here.]

After each advection time step the anti-diffusion correction is evolved in pseudo time. Note that in order to maintain the interface location during the successive correction steps n_I is calculated at the first anti-diffusion correction step [14], i.e. $\tau = 0$. The anti-diffusion equation (24) is evaluated with constant \mathbf{n}_I

$$\mathbf{n}_I = \frac{(\nabla \alpha)_{\tau=0}^*}{|(\nabla \alpha)_{\tau=0}^*|} \quad . \quad (27)$$

3.3. Ensuring boundedness of the volume fraction when solving the anti-diffusion equation

Based on the regularization of the anti-diffusion equation by a *minmod* function in the 1-dimensional setting detailed in section 3.1.1, we propose a limiter based on the directional derivative to ensure boundedness of the volume fraction on unstructured meshes. The anti-diffusion algorithm involves the following steps:

- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
 - 8
 - 9
 - 10
 - 11
 - 12
 - 13
 - 14
 - 15
 - 16
 - 17
 - 18
 - 19
 - 20
 - 21
 - 22
 - 23
 - 24
 - 25
 - 26
 - 27
 - 28
 - 29
 - 30
 - 31
 - 32
 - 33
 - 34
 - 35
 - 36
 - 37
 - 38
 - 39
 - 40
 - 41
 - 42
 - 43
 - 44
 - 45
 - 46
 - 47
 - 48
 - 49
 - 50
 - 51
 - 52
 - 53
 - 54
 - 55
 - 56
 - 57
 - 58
 - 59
 - 60
 - 61
 - 62
 - 63
 - 64
 - 65
1. A cell average of $\nabla\alpha$ is calculated following the Gauss theorem with a limiter based on the directional derivative.
 2. The calculated cell-averaged $\nabla\alpha$ is projected onto the interface-normal direction and multiplied by the diffusion coefficient and the interface-normal vector to obtain $[-|\mathbf{u}|(\nabla\alpha \cdot \mathbf{n}_I)\mathbf{n}_I]$.
 3. The cell-averaged divergence $\nabla \cdot [-|\mathbf{u}|(\nabla\alpha \cdot \mathbf{n}_I)\mathbf{n}_I]$ is calculated from the Gauss theorem.
 4. An Euler explicit scheme is used for pseudo-time discretization of the anti-diffusion equation.

During pseudo-time advancement \mathbf{n}_I is fixed at its initial value.

3.3.1. Calculation of $\nabla\alpha$

A limited value of the volume-fraction gradient $\nabla\alpha$ is calculated by Gauss theorem with an interpolation of the volume fraction at the cell face based on the following algorithm:

Step 1. Given cells P and N (see Fig. 3), we first calculate:

- The cell-averaged gradient $(\nabla\alpha)^*$ from equation (26) for cells P and N.
- The directional derivative $\frac{\partial\alpha}{\partial c}$ from

$$\frac{\partial\alpha}{\partial c} = \frac{(\alpha_N - \alpha_P)}{\|\mathbf{c}_N - \mathbf{c}_P\|} \quad , \quad (28)$$

where \mathbf{c}_P and \mathbf{c}_N are the cell-center-position vectors of cell P and N respectively.

Step 2. $(\nabla\alpha)_P^*$ is projected onto the cell-face-normal direction, leading to the term $(\nabla\alpha)_P^* \cdot \hat{\mathbf{n}}_{cf}$, where $\hat{\mathbf{n}}_{cf}$ is the cell-face unit normal of the face between the cells P and N. $(\nabla\alpha)_P^* \cdot \hat{\mathbf{n}}_{cf}$ and $\frac{\partial\alpha}{\partial c}$ are compared to select the volume fraction α' from α_P and α_N for calculating the limited gradient of volume fraction in the next

step. The selection is equivalent to the step (18a) and (18b) in the 1-dimensional setting. Direction and magnitude of $(\nabla\alpha)_P^* \cdot \hat{\mathbf{n}}_{cf}$ and $\frac{\partial\alpha}{\partial c}$ are compared for choosing the volume fraction as

$$\alpha' = \begin{cases} \alpha_N & \text{if } [(\nabla\alpha)_P^* \cdot \hat{\mathbf{n}}_{cf}] \frac{\partial\alpha}{\partial c} > 0 \text{ and } |(\nabla\alpha)_P^* \cdot \hat{\mathbf{n}}_{cf}| < \left| \frac{\partial\alpha}{\partial c} \right| \\ \frac{\alpha_P + \alpha_N}{2} & \text{if } [(\nabla\alpha)_P^* \cdot \hat{\mathbf{n}}_{cf}] \frac{\partial\alpha}{\partial c} > 0 \text{ and } |(\nabla\alpha)_P^* \cdot \hat{\mathbf{n}}_{cf}| = \left| \frac{\partial\alpha}{\partial c} \right| \\ \alpha_P & \text{else} \end{cases} . \quad (29)$$

This procedure is applied to all cell faces of each cell with the respective cell neighbors N.

Step 3. The cell-averaged value of the volume-fraction gradient $(\nabla\alpha)_P$ for cell P is calculated by the Gauss theorem based on the α' selected from the *Step 2* above.

$$(\nabla\alpha)_P = \frac{\sum_{cf} (\alpha' \mathbf{S})}{V} . \quad (30)$$

This procedure is equivalent to step (17) in the 1-dimensional setting. Comparing (30) to the equation (26) it can be recognized that regularization is achieved by replacing cell-face values of α obtained from an arithmetic average by the α obtained from the above selection procedure.

3.3.2. Divergence of $[-|\mathbf{u}|(\nabla\alpha \cdot \mathbf{n}_I) \mathbf{n}_I]$

After obtaining $(\nabla\alpha)_P$, it is projected onto the interface-normal direction by multiplication with \mathbf{n}_I which is given by equation (27). Equivalent to the calculation of the term $\nabla \cdot (-\nabla\alpha) = \frac{(F_{i-1/2} - F_{i+1/2})}{\Delta x}$ in the 1-dimensional setting, the flux at the cell face between cell P and cell N is limited by selection based on the minimum of $\|((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I\|$ and $\|((\nabla\alpha)_N \cdot \mathbf{n}_I) \mathbf{n}_I\|$

$$((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I = \begin{cases} ((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I & \text{if } \|((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I\| \leq \|((\nabla\alpha)_N \cdot \mathbf{n}_I) \mathbf{n}_I\| \\ ((\nabla\alpha)_N \cdot \mathbf{n}_I) \mathbf{n}_I & \text{if } \|((\nabla\alpha)_N \cdot \mathbf{n}_I) \mathbf{n}_I\| < \|((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I\| \end{cases} \quad (31)$$

The limited flux is further multiplied by the diffusion coefficient $-|\mathbf{u}_P|$, i.e. the cell-averaged velocity magnitude for cell P. Then the cell average of $\nabla \cdot [-|\mathbf{u}_P|((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I]$ is calculated by the Gauss theorem as

$$\nabla \cdot [-|\mathbf{u}_P|((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I] = \frac{\sum_{cf} ([-|\mathbf{u}_P|((\nabla\alpha)_P \cdot \mathbf{n}_I) \mathbf{n}_I] \cdot \mathbf{S})}{V} \quad . \quad (32)$$

3.3.3. Time-step criterion

The time derivative is discretized by an explicit Euler scheme. The volume fraction is forwarded in pseudo time by

$$\alpha^{n+1} = \alpha^n + (\nabla \cdot [-|\mathbf{u}|(\nabla\alpha \cdot \mathbf{n}_I) \mathbf{n}_I]) \Delta\tau \quad , \quad (33)$$

where α^{n+1} and α^n are the volume fractions at the new and the old time step respectively. In the 1-dimensional setting the CFL requirement (20) for the stabilized inverse diffusion equation applies. From numerical experimentation we find that stable time integration for multiple dimensions, unstructured meshes and variable diffusion coefficient is achieved by

$$\Delta\tau = \frac{1}{4} \frac{(\Delta x)^2}{|\mathbf{u}|_{max}} \quad , \quad (34)$$

where Δx is the minimum cell width and $|\mathbf{u}|_{max}$ the maximum velocity magnitude over the entire computational domain.

1
2
3 *3.4. Stopping criterion for the anti-diffusion correction*

4
5 For defining a stopping criterion for the anti-diffusion correction formulation in
6 multiple dimensions and unstructured meshes, the term $\nabla \cdot [-|\mathbf{u}|((\nabla\alpha) \cdot \mathbf{n}_I) \mathbf{n}_I]$
7 is used to measure the interface sharpness. Sharpness tolerances TOL_1 and TOL_2
8 are defined as
9
10

$$11 \quad TOL_1 \leq \frac{\sum_i |\nabla \cdot [(\nabla\alpha_i \cdot \mathbf{n}_I) \mathbf{n}_I]| \cdot V}{\sum_i |(\nabla\alpha_i)^*|^2 \cdot V} \quad (35)$$

$$12 \quad TOL_2 \leq \frac{\max_i (|\nabla \cdot [(\nabla\alpha_i \cdot \mathbf{n}_I) \mathbf{n}_I]| \cdot V)}{\max_i (|(\nabla\alpha_i)^*|^2 \cdot V)} \quad (36)$$

13
14
15
16
17
18
19 For demonstrating the effect of the tolerances the anti-diffusion correction is
20 applied to a steady diffused 3-dimensional profile, Fig. 4. The corresponding
21 evolution of TOL_1 and TOL_2 is shown in Fig. 5. For realistic flow simulations
22 we find by numerical experimentation that a desired interface sharpness and the
23 simulation stability can be achieved by $TOL_1 = 0.75$ and $TOL_2 = 0.75$. These
24 interface sharpness tolerances are used in all numerical validation computations of
25 the next section.
26
27
28
29
30
31
32
33
34
35
36

37
38 [Figure 4 about here.]

39
40
41 [Figure 5 about here.]

42
43
44 **4. Numerical results**

45
46
47 The anti-diffusion interface-steepening algorithm is validated with the following
48 numerical examples. First, a 2-dimensional rising-bubble case is considered. A
49 comparison with reference data from literature and a convergence study are carried
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1
2
3 out. Second, a 3-dimensional rising-bubble, and third a 2-dimensional and a 3-
4 dimensional axisymmetric rising-drop on a non-Cartesian mesh are examined. In
5 all cases for the numerical discretization of the volume-fraction-transport equation
6 the van Leer limiter [29] is employed for calculating the flux and an Euler explicit
7 scheme is employed for time integration. All computations are carried out with a
8 CFL number of 0.5.
9

14 4.1. 2-dimensional rising bubble

17 First, the 2-dimensional rising-bubble case of Olsson and Kreiss [13] is consid-
18 ered. The parameters used in this numerical example are

$$21 \quad \rho_1 = 1, \quad \rho_2 = 0.0013,$$

$$22 \quad \mu_1 = 1, \quad \mu_2 = 0.016,$$

$$23 \quad \sigma = 7.3 \times 10^{-2} \text{ N/m},$$

$$24 \quad Re = 500, \quad Fr = 0.45, \quad We = 0.68,$$

25 where ρ is the density, μ is the liquid viscosity, σ is the surface tension of the
26 liquid, and the subscripts 1 and 2 refer respectively to the water phase and the air
27 phase. The reference parameters are $\rho_{ref} = 1.0 \times 10^3 \text{ kg/m}^3$, $l_{ref} = 5.0 \times 10^{-3} \text{ m}$
28 and $u_{ref} = 0.1 \text{ m/s}$. The computational domain size is $2l_{ref} \times 4l_{ref}$. The bubble
29 is initialized at the position (l_{ref}, l_{ref}) . Four different grid resolutions are used:
30 $\Delta x = 2/25$, $\Delta x = 2/50$, $\Delta x = 2/100$, $\Delta x = 2/200$.
31
32
33
34
35
36
37
38
39

40 The volume-fraction contours 0.05, 0.5 and 0.95 of the bubble at $t = 0.5$ ob-
41 tained by the anti-diffusion interface steepening method are shown in Fig. 6(a).
42 The result of Olsson and Kreiss [13] is reproduced in Fig. 6(b) for comparison.
43 Bubble-shape convergence is observed for a grid refinement from $\Delta x = 2/25$ to
44 $\Delta x = 2/200$. The bubble shapes at lower grid resolutions are significantly more
45 accurate than that of [13]. The bubble rising velocity is shown in Fig. 7(a), and
46 compared with that of [13] in Fig. 7(b). The grid convergence of the rising velocity
47 is observed to be faster than that of [13].
48
49
50
51
52
53
54
55

[Figure 6 about here.]

[Figure 7 about here.]

4.2. 3-dimensional rising bubble

Second, for validation in 3 dimensions, the case of an air bubble rising in a water-glucose solution based on the experiment carried out by Bhaga and Weber [30] is studied. The case can be characterized by the Reynolds number, the Eötvös number and the Morton number

$$Re = \frac{\rho d_B U_B}{\mu} = 2.47, \quad E\ddot{o} = \frac{g d_e^2 \rho}{\sigma} = 116, \quad Mo = \frac{g \mu^4}{\rho \sigma^3} = 848 \quad ,$$

where ρ is the liquid density, $d_B = 0.0261 m$ is the bubble volume-equivalent diameter, U_B is the bubble terminal rising velocity, μ is the liquid viscosity, σ is the surface tension of the liquid and g is the gravitational acceleration.

The computational domain size is $5d_e \times 10d_e \times 5d_e$. The grid resolution is $50 \times 100 \times 50$. A spherical bubble of $d_e = 1$ is initialized at position $(2.5d_e, 1d_e, 2.5d_e)$. The boundary condition at the front, back, left, right and bottom domain boundaries is set as no-slip wall. Corresponding to the experimental setup of a rising bubble in a vertical tube open to the atmosphere, the top boundary is set to a free-surface boundary condition. For such a boundary condition the volume fraction can be either Neumann-type (zero gradient) or Dirichlet-type, depending on the direction of the flux [10], where the boundary condition is defined as *inlet / outlet* in the *OpenFOAM* library [15]. The pressure is adjusted by the Bernoulli equation, where the boundary condition is defined as *totalPressure* in the *OpenFOAM* library. The velocity is evaluated from the pressure and the direction of the flux, where the boundary condition is defined as *pressure inlet / outlet velocity* in the *OpenFOAM* library.

1
2
3 The bubble reaches a steady shape after an initial transient period. The
4 volume-fraction contour 0.5 of the steady bubble at $t = 0.55 s$ is shown in Fig.
5 8(a). The interface sharpness can be evaluated by the volume-fraction contours
6 0.05, 0.5 and 0.95 in Fig. 8(b). A terminal oblate-ellipsoidal-cap bubble shape
7 is obtained, which corresponds to the experimental observation [30]. The rising
8 velocity of the bubble is plotted in Fig. 9. The computed Reynolds number based
9 on the terminal rising velocity from the simulation is 2.40 and compares well to
10 the experimental Reynolds number of 2.47.
11
12
13
14
15
16
17

18 [Figure 8 about here.]
19

20 [Figure 9 about here.]
21
22
23

24 4.3. Rising drop in a periodically constricted capillary tube

25 Third, in order to validate the anti-diffusion correction algorithm for non-
26 Cartesian meshes, a case of a drop rising in a periodically constricted capillary
27 tube is considered. The case is based on the experiment carried out by Hemmat
28 and Borhan [31] and simulated previously by Muradoglu and Kayaalp [32]. The
29 parameters used in this numerical example are
30
31
32
33
34
35

$$36 \rho_o = 1160 \text{ kg/m}^3, \rho_d = 966 \text{ kg/m}^3,$$

$$37 \mu_o = 87 \text{ mPa s}, \mu_d = 115 \text{ mPa s},$$

$$38 \sigma = 0.0042 \text{ N/m},$$

39 where the subscripts o and d denote the ambient fluid and the drop fluid respec-
40 tively. Similarly as [32], a portion of 26 cm of a periodically constricted capillary
41 tube is selected as the computational domain, which is shown in Fig. 10. The
42 constricted capillary tube has the following geometric parameters: average radius,
43 $R = 0.5 \text{ cm}$, wavelength of corrugations, $h = 4 \text{ cm}$, and amplitude of corrugations
44 $A = 0.07 \text{ cm}$. The grid resolution of the computational domain is 32 x 1664. The
45 size of the drop is measured by κ which is defined as the ratio of the equivalent
46
47
48
49
50
51
52
53
54
55

1
2
3 spherical drop radius to R . A drop of $\kappa = 0.92$ is initialized at the height of 0.01 cm
4 along the centreline. Simulation time is non-dimensionalized by
5
6

$$7 \quad t_{ref} = \frac{\mu_o}{\Delta\rho g_y R} \quad , \quad (37)$$

8
9 where t_{ref} is the reference time, $\Delta\rho = \rho_o - \rho_d$ and g_y is the gravitational
10 acceleration.
11
12

13
14 [Figure 10 about here.]
15

16
17 First, a 2-dimensional planar case is simulated where the boundary condition
18 of the top, bottom and right boundaries is no-slip wall, and at the left boundary
19 a symmetry condition is imposed. Second, an axisymmetric 3-dimensional case is
20 simulated.
21
22

23 The volume-fraction contours 0.05, 0.5 and 0.95 at different time instants are
24 shown in Fig. 11 for the 2-dimensional planar case and the axisymmetric case. A
25 sharp interface is obtained on the given non-Cartesian meshes for both cases. In
26 addition it can be seen that the drop shape is periodic with respect to the periodic
27 corrugation. This is in good agreement with the experiment [31] where it was
28 stated that the drop deformation parameter was found to be periodic for all drop
29 sizes without drop breakup. Note that [32] does not recover this behavior. In the
30 2-dimensional planar case the drop area corresponds to the drop mass, allowing to
31 assess the discrete conservation of mass. A comparison of the drop area at different
32 time steps with the simulation by Muradoglu and Kayaalp [32] suggests that the
33 drop mass is better conserved by our simulations.
34
35
36
37
38
39
40
41
42
43
44
45
46
47

48 [Figure 11 about here.]
49
50

51 Further comparisons of the drop shape in the 2-dimensional planar case and
52 the axisymmetric case, as shown in Fig. 12, reveal that the necking of the rising
53
54
55

1
2 drop after the corrugation is more pronounced in the axisymmetric case. This
3 agrees with the observation that periodic constrictions lead to stronger cross-
4 section reduction in 3 dimensions than in 2 dimensions, resulting in a stronger
5 drop deformation.
6
7
8
9

10 [Figure 12 about here.]
11
12
13

14 5. Concluding remarks 15

16
17 In this paper we have proposed an interface steepening method by an anti-
18 diffusion correction for two-phase incompressible flows based on the VOF interface-
19 capturing method. The method possesses the following properties: (i) no interface
20 reconstruction is required for the volume-fraction advection, (ii) a sharp phase-
21 interface is maintained throughout the simulation, and (iii) a desired interface
22 sharpness can be attained based on a case- and grid-resolution -independent inter-
23 face sharpness measurement.
24
25
26
27
28
29

30 First, in a 1-dimensional setting an anti-diffusion equation is formulated, and
31 the equation is solved after each volume-fraction advection. A *minmod* function is
32 employed to regularize the anti-diffusion equation for a numerical solution. Modifi-
33 cations to the *minmod* function, which allow for an extension to multiple dimension
34 and unstructured meshes, are presented. 1-dimensional results show that the in-
35 terface can be steepened and the boundedness of the volume-fraction is preserved.
36 A suitable interface sharpness measurement is developed that does not require the
37 evaluation of additional terms as it is based on the flux difference directly.
38
39
40
41
42
43
44
45

46 The anti-diffusion method is extended to multiple dimensions and unstructured
47 meshes, and coupled with the volume-fraction advection and the Navier-Stokes
48 equation for realistic flow simulations. The anti-diffusion equation is reformulated
49 by taking into account the interface normal direction and by adopting the veloc-
50 ity magnitude as the diffusion coefficient, following modified-differential equation
51
52
53
54
55

1
2
3 analysis of the numerical diffusion due to advection. A limiter based on the di-
4 rectional derivative, which follows the 1-dimensional modified *minmod* function,
5 is proposed and allows for stable solution of the anti-diffusion equation. The anti-
6 diffusion correction can be carried out repeatedly to attain a desired interface
7 sharpness. A stopping criterion for the anti-diffusion iteration, which is based on
8 the interface sharpness measurement, is developed.
9

10
11
12 Validation computations are performed for 2- and 3- dimensional rising-bubble
13 and rising-drop configurations, and for Cartesian and non-Cartesian meshes. The
14 results agree well with experiments and show the advantages of the current method
15 in comparison with previous simulations by other methods. In particular, the
16 method is more accurate at low grid resolutions and the grid convergence is faster
17 as compared, e.g., to reference [13]. The simulation of a drop rising in a periodically
18 constricted capillary tube by our method reproduces the experimental observation
19 of periodic drop-deformation [31], which suggests that the current method is a
20 significant improvement as compared to reference [32].
21
22

23
24
25 Though originally being proposed for solving the anti-diffusion equation, two
26 aspects of the present method may also be applied to other computational modeling
27 problems. First, as the proposed modification of the *minmod* function can be em-
28 ployed as a general slope limiter, it may also be applied to more general equations
29 on unstructured meshes than has been done in this paper. Second, as the proposed
30 interface-sharpness measurement gives a general assessment of a VOF interface rep-
31 resentation, which is independent of the specific interface-steeping method, it may
32 also be used within other methods such as the artificial-compression steepening
33 [13].
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1
2
3 **References**
4

- 5 [1] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incom-
6 pressible, multi-fluid flows, *Journal of Computational Physics* 100 (1) (1992)
7 25-37.
8
9
10
11 [2] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber,
12 J. Han, S. Nas, Y. J. Jan, A front-tracking method for the computations of
13 multiphase flow, *Journal of Computational Physics* 169 (2) (2001) 708-759.
14
15
16
17 [3] D. J. Torres, J. U. Brackbill, The point-set method: Front-tracking without
18 connectivity, *Journal of Computational Physics* 165 (2) (2000) 620-644.
19
20
21
22 [4] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and in-
23 terfacial flow, *Annual Review of Fluid Mechanics* 31 (1999) 567-603.
24
25
26 [5] S. Osher, R. P. Fedkiw, Level set methods: An overview and some recent
27 results, *Journal of Computational Physics* 169 (2) (2001) 463-502.
28
29
30
31 [6] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, *Annual Review*
32 *of Fluid Mechanics* 35 (2003) 341-372.
33
34
35 [7] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, *Journal of Com-*
36 *putational Physics* 141 (2) (1998) 112-152.
37
38
39 [8] W. Noh, P. Woodward, *Lecture Notes in Physics*, Springer Berlin, 1976.
40
41
42 [9] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of
43 free boundaries, *Journal of Computational Physics* 39 (1) (1981) 201-225.
44
45
46 [10] H. Rusche, *Computational fluid dynamics of dispersed two-phase flows at*
47 *high phase fractions*, Ph.D. thesis, Imperial College of Science, Technology &
48 *Medicine* (2002).
49
50
51
52
53
54
55

- 1
2
3 [11] O. Ubbink, R. I. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, *Journal of Computational Physics* 153 (1) (1999) 26-50.
4
5
6
7 [12] Z. Xu, C.-W. Shu, Anti-diffusive flux corrections for high order finite difference
8
9
10
11
12 [13] E. Olsson, G. Kreiss, A conservative level set method for two phase flow,
13
14
15
16
17 [14] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase
18
19
20
21
22 [15] Openfoam, <http://www.opencfd.co.uk/openfoam/>, version 1.5.
23
24 [16] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling
25
26
27
28
29 [17] H. Jasak, H. G. Weller, A. D. Gosman, High resolution NVD differencing
30
31
32
33
34
35
36 [18] R. I. Issa, Solution of the implicitly discretised fluid flow equations by
37
38
39
40
41 [19] O. Ubbink, Numerical prediction of two fluid systems with sharp interfaces,
42
43
44
45
46 [20] J. P. Boris, D. L. Book, Flux-corrected transport. I. SHASTA, a fluid trans-
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

- 1
2
3 [21] I. Pollak, A. S. Willsky, H. Krim, Image segmentation and edge enhance-
4 ment with stabilized inverse diffusion equations, IEEE Transactions on Image
5 Processing 9 (2) (2000) 256-266.
6
7
8
9 [22] G. Gilboa, N. Sochen, Y. Y. Zeevi, Forward-and-backward diffusion processes
10 for adaptive image enhancement and denoising, IEEE Transactions on Image
11 Processing 11 (7) (2002) 689-703.
12
13
14 [23] D. S. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-
15 oscillatory schemes with increasingly high order of accuracy, Journal of Compu-
16 tational Physics 160 (2) (2000) 405-452.
17
18
19 [24] W. J. Rider, L. G. Margolin, Simple modifications of monotonicity-preserving
20 limiter, Journal of Computational Physics 174 (1) (2001) 473-488.
21
22
23 [25] S. Osher, L. I. Rudin, Shocks and other nonlinear filtering applied to image
24 processing, Vol. 1567, SPIE, 1991, pp. 414-431.
25
26
27 [26] M. Breuß, T. Brox, T. Sonar, J. Weickert, Stabilized nonlinear inverse dif-
28 fusion for approximating hyperbolic PDEs, Scale Space and PDE Methods in
29 Computer Vision, Proceedings 3459 (2005) 536-547.
30
31
32 [27] M. Breuß, M. Welk, Staircasing in semidiscrete stabilized inverse linear dif-
33 fusion algorithms, Journal of Computational and Applied Mathematics 206 (1)
34 (2007) 520-533.
35
36
37 [28] R. J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge
38 University Press, 2002.
39
40
41 [29] B. van Leer, Towards the ultimate conservative difference scheme. II. mono-
42 tonicity and conservation combined in a second-order scheme, Journal of Com-
43 putational Physics 14 (4) (1974) 361-370.
44
45
46
47
48
49
50
51
52
53
54
55

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

[30] D. Bhaga, M. E. Weber, Bubbles in viscous-liquid shapes, wakes and velocities, *Journal of Fluid Mechanics* 105 (Apr) (1981) 61-85.

[31] M. Hemmat, A. Borhan, Buoyancy-driven motion of drops and bubbles in a periodically constricted capillary, *Chemical Engineering Communications* 150 (1996) 363-384.

[32] M. Muradoglu, A. D. Kayaalp, An auxiliary grid method for computations of multiphase flows in complex geometries, *Journal of Computational Physics* 214 (2) (2006) 858-877.

1
2
3 **List of Figures**
4

5 1 1-dimensional profile of volume fraction α from value 0 to 1. i
6 denotes the cell index. 1st AD denotes the profile after the first
7 anti-diffusion correction. 29
8
9 2 Evolution of TOL_1 and TOL_2 corresponding to the anti-diffusion
10 correction in Fig. 1. n denotes the number of anti-diffusion correction. 30
11
12 3 The cell arrangement in unstructured meshes. 31
13
14 4 Volume-fraction contours 0.05, 0.5 and 0.95 of a 3-dimensional dif-
15 fused profile. (a) Initial profile (3-dimensional view). (b) Initial
16 profile (sectional view). (c) Profile after 4 anti-diffusion corrections
17 (3-dimensional view). (d) Profile after 4 anti-diffusion corrections
18 (sectional view). 32
19
20 5 Evolution of TOL_1 and TOL_2 corresponding to the anti-diffusion
21 correction in Fig. 4. n denotes the number of anti-diffusion correction. 33
22
23 6 The volume-fraction contours 0.05, 0.5, 0.95 of the bubble at $t =$
24 0.5 in four different grid resolutions. From left to right: $\Delta x =$
25 $2/25$, $\Delta x = 2/50$, $\Delta x = 2/100$, $\Delta x = 2/200$. (a) Result by anti-
26 diffusion interface sharpening. (b) Result of Olsson and Kreiss [13],
27 reproduced with permission. 34
28
29 7 The rising velocity of the bubble in four different grid resolutions.
30 (a) Result by anti-diffusion interface sharpening. (b) Result of
31 Olsson and Kreiss [13], reproduced with permission. Dotted line:
32 25×50 ; dashed-dotted line: 50×100 ; dashed line: 100×200 ; solid
33 line: 200×400 35
34
35 8 (a) 3-dimensional view of volume-fraction contour 0.5 of the bubble
36 at steady state at $t = 0.55s$. (b) Sectional view of volume-fraction
37 contours 0.05, 0.5 and 0.95 of the bubble at steady state at $t = 0.55s$. 36
38
39 9 The rising velocity u_B of the bubble in the 3-dimensional rising-
40 bubble case. 37
41
42 10 The computational domain for case of 2-dimensional rising drop in
43 a periodically constricted capillary tube. (a) Full domain. (b) Plan
44 view showing dimensional details (not to scale). 38
45
46 11 The volume-fraction contours 0.05, 0.5 and 0.95 of the rising drop.
47 (a) Plots for *2-dimensional planar* simulation from $t = 0$ to $t =$
48 3937.5 at time interval of 437.5 . (b) Plots for *axisymmetric 3-*
49 *dimensional* simulation from $t = 0$ to $t = 4823.1$ at time interval of
50 535.9 39
51
52 12 The volume-fraction contours 0.05, 0.5 and 0.95 of the rising drop.
53 (a) Plot at $t = 1750$ for the *2-dimensional planar* simulation. (b)
54 Plot at $t = 2143.6s$ for the *axisymmetric 3-dimensional* simulation. 40
55
56
57
58
59
60
61
62
63
64
65

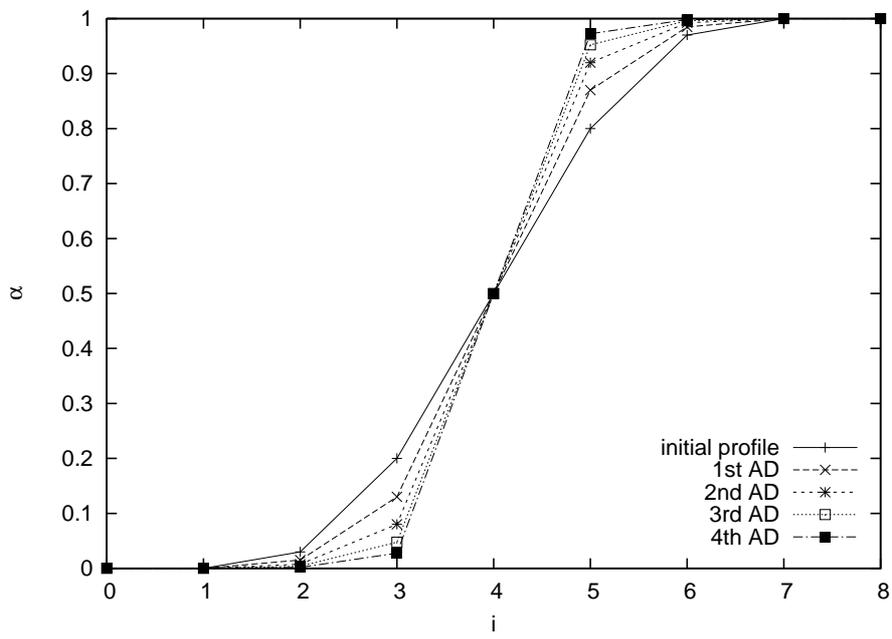


Figure 1: 1-dimensional profile of volume fraction α from value 0 to 1. i denotes the cell index. 1st AD denotes the profile after the first anti-diffusion correction.

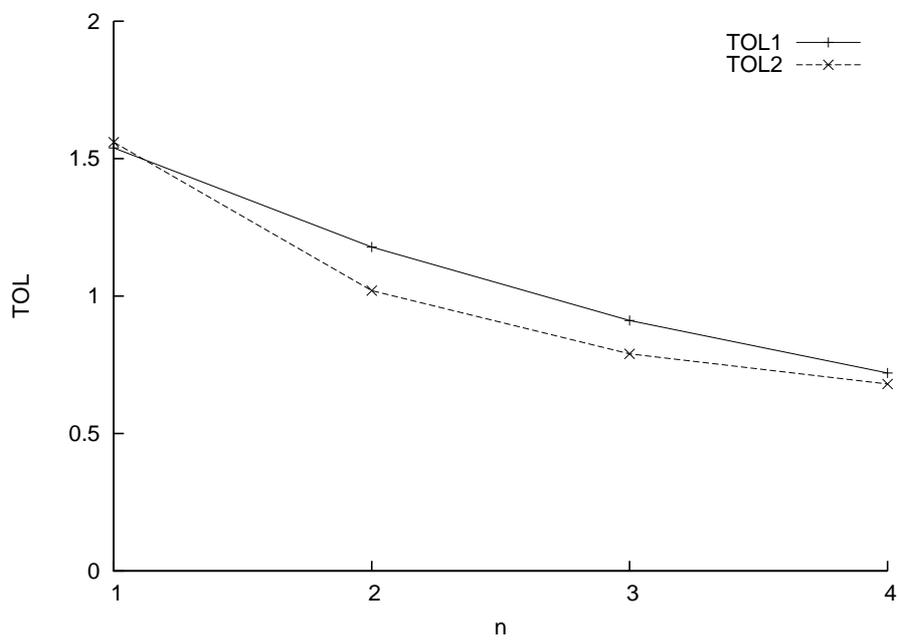


Figure 2: Evolution of TOL_1 and TOL_2 corresponding to the anti-diffusion correction in Fig. 1. n denotes the number of anti-diffusion correction.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

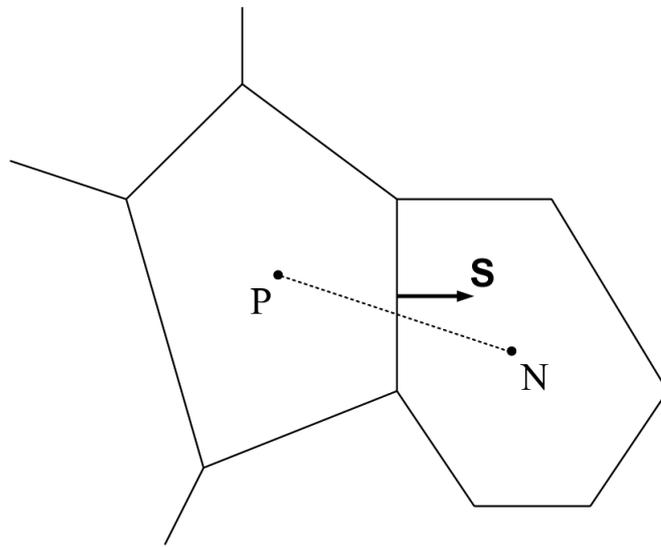
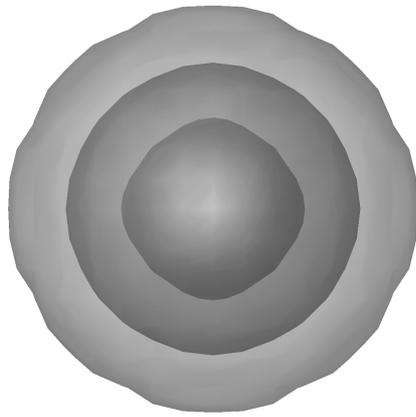
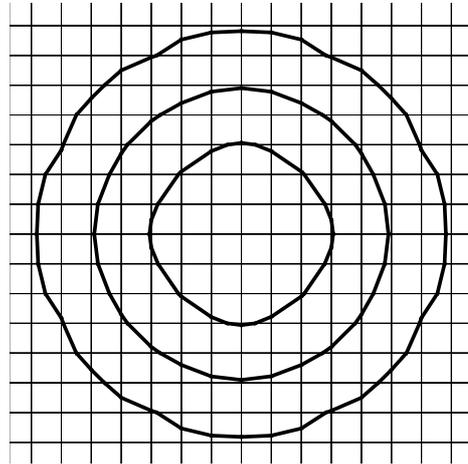


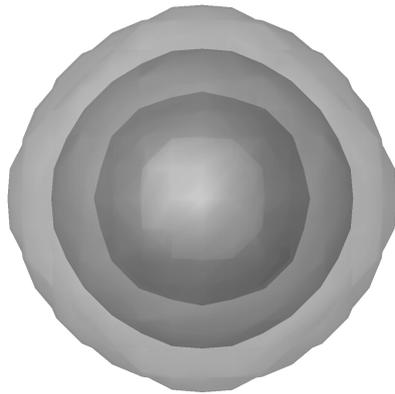
Figure 3: The cell arrangement in unstructured meshes.



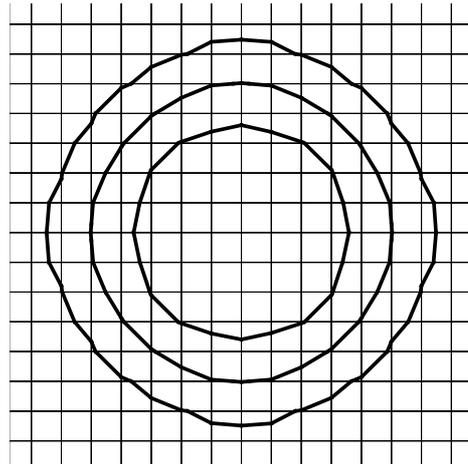
(a)



(b)



(c)



(d)

Figure 4: Volume-fraction contours 0.05, 0.5 and 0.95 of a 3-dimensional diffused profile. (a) Initial profile (3-dimensional view). (b) Initial profile (sectional view). (c) Profile after 4 anti-diffusion corrections (3-dimensional view). (d) Profile after 4 anti-diffusion corrections (sectional view).

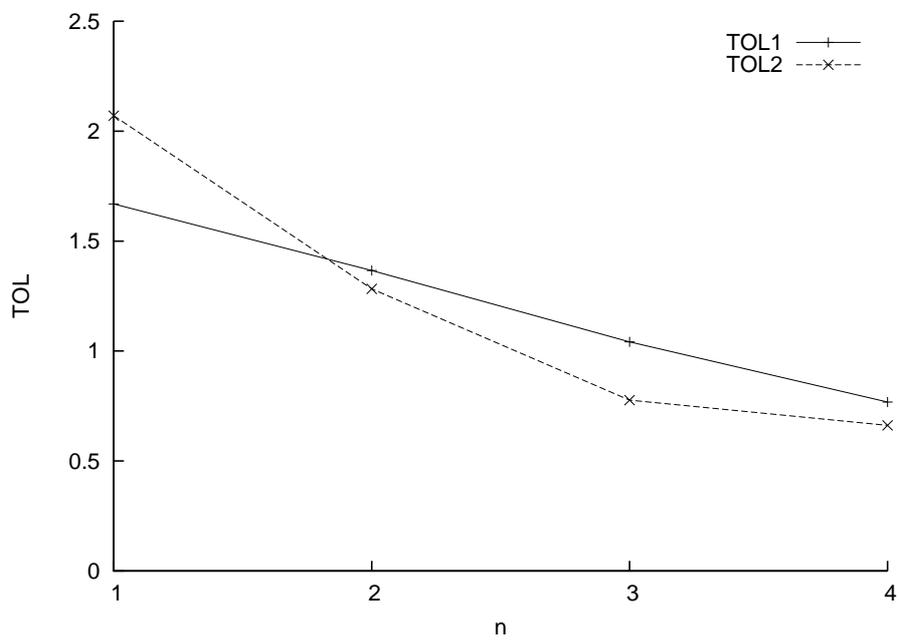


Figure 5: Evolution of TOL_1 and TOL_2 corresponding to the anti-diffusion correction in Fig. 4. n denotes the number of anti-diffusion correction.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

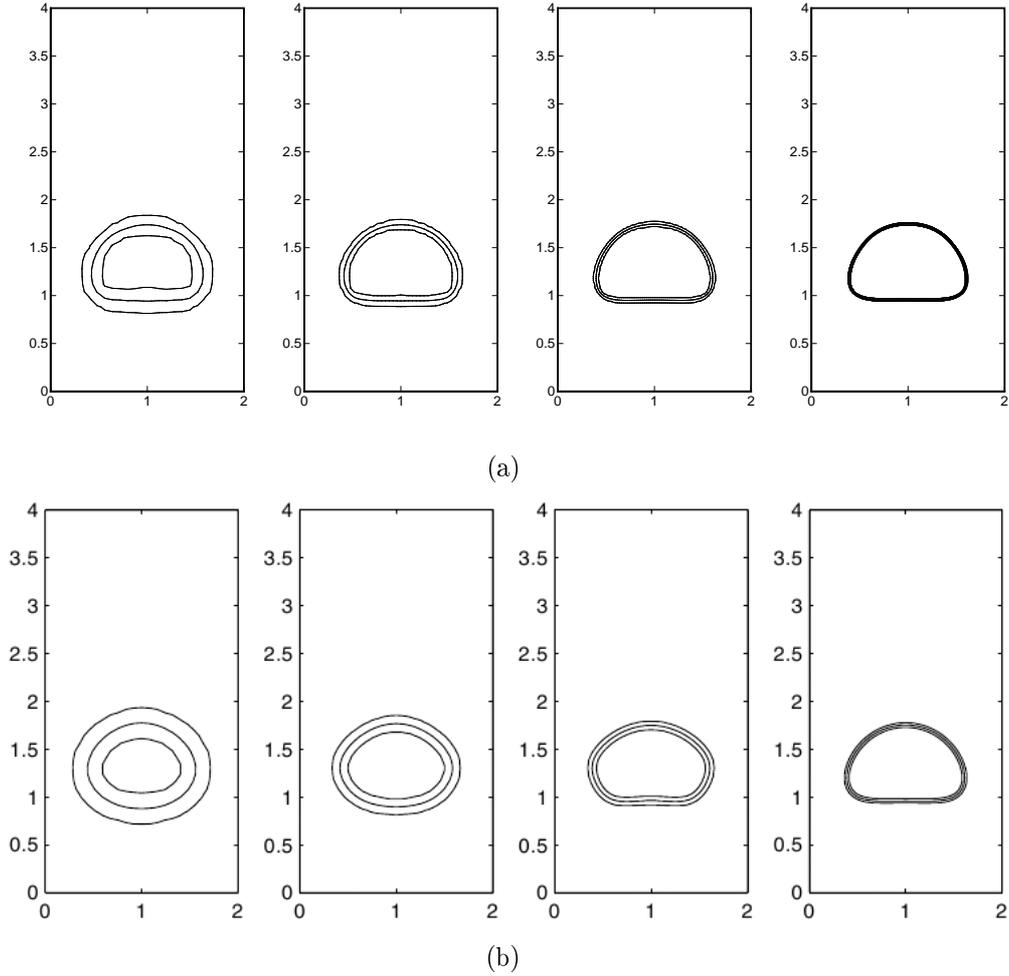


Figure 6: The volume-fraction contours 0.05, 0.5, 0.95 of the bubble at $t = 0.5$ in four different grid resolutions. From left to right: $\Delta x = 2/25$, $\Delta x = 2/50$, $\Delta x = 2/100$, $\Delta x = 2/200$. (a) Result by anti-diffusion interface sharpening. (b) Result of Olsson and Kreiss [13], reproduced with permission.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

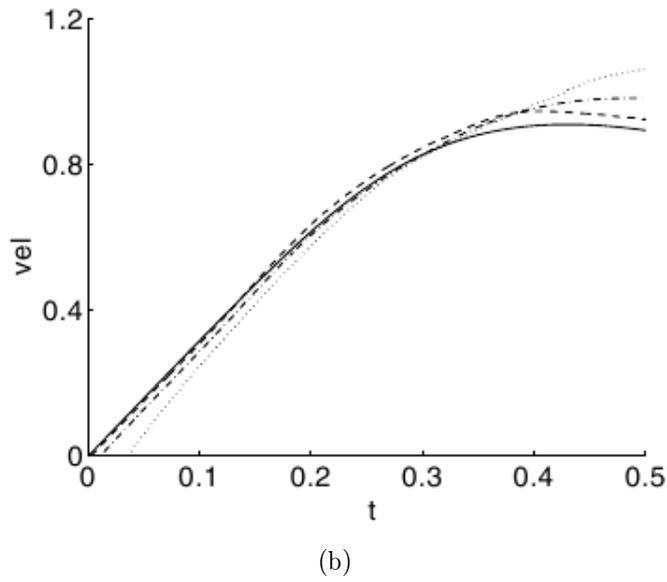
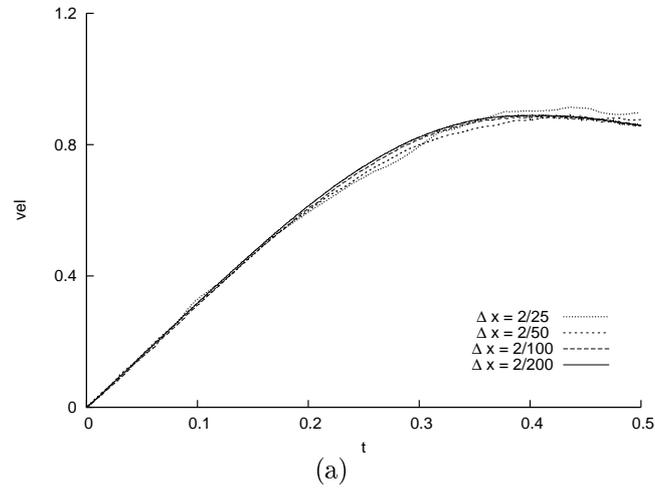


Figure 7: The rising velocity of the bubble in four different grid resolutions. (a) Result by anti-diffusion interface sharpening. (b) Result of Olsson and Kreiss [13], reproduced with permission. Dotted line: 25×50 ; dashed-dotted line: 50×100 ; dashed line: 100×200 ; solid line: 200×400 .

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

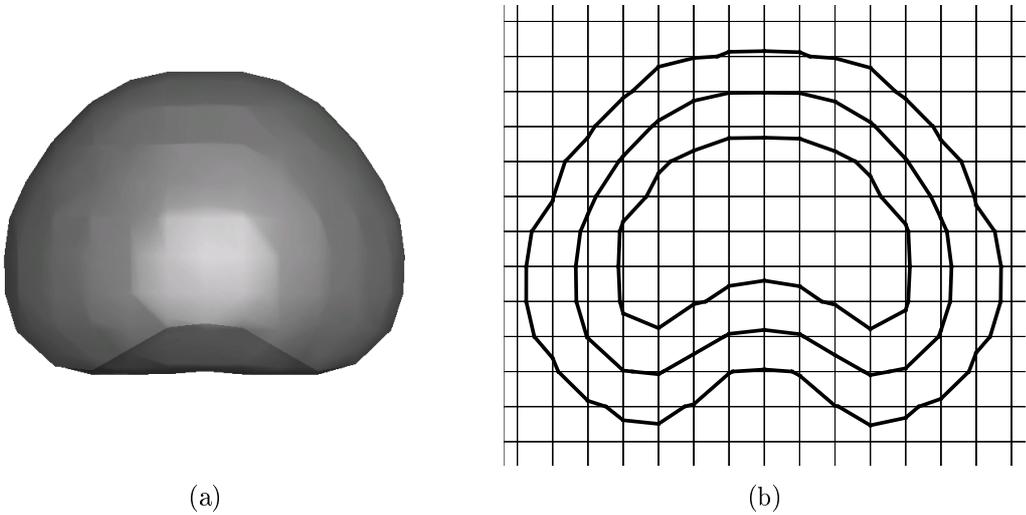


Figure 8: (a) 3-dimensional view of volume-fraction contour 0.5 of the bubble at steady state at $t = 0.55s$. (b) Sectional view of volume-fraction contours 0.05, 0.5 and 0.95 of the bubble at steady state at $t = 0.55s$.

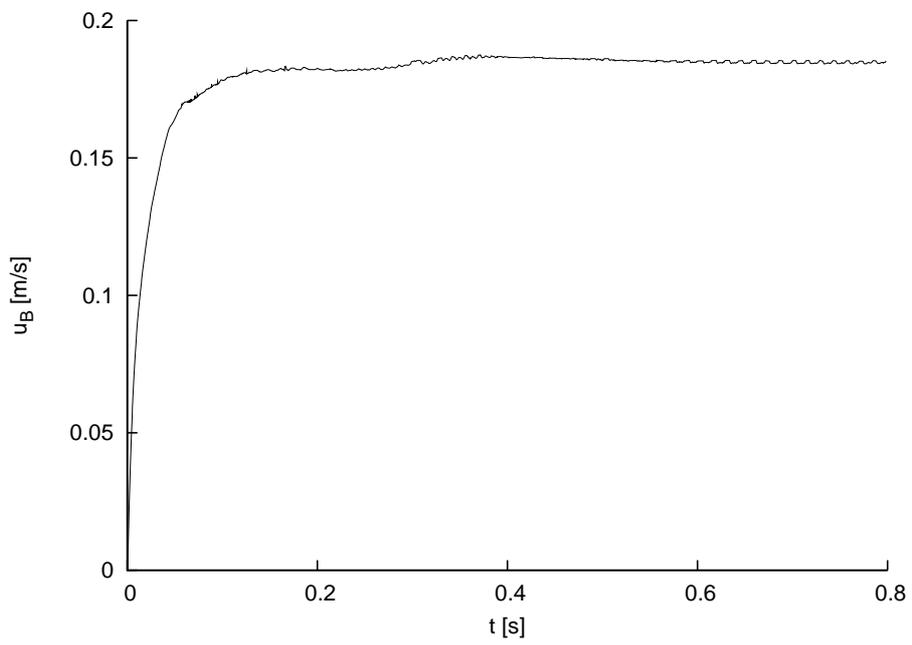


Figure 9: The rising velocity u_B of the bubble in the 3-dimensional rising-bubble case.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

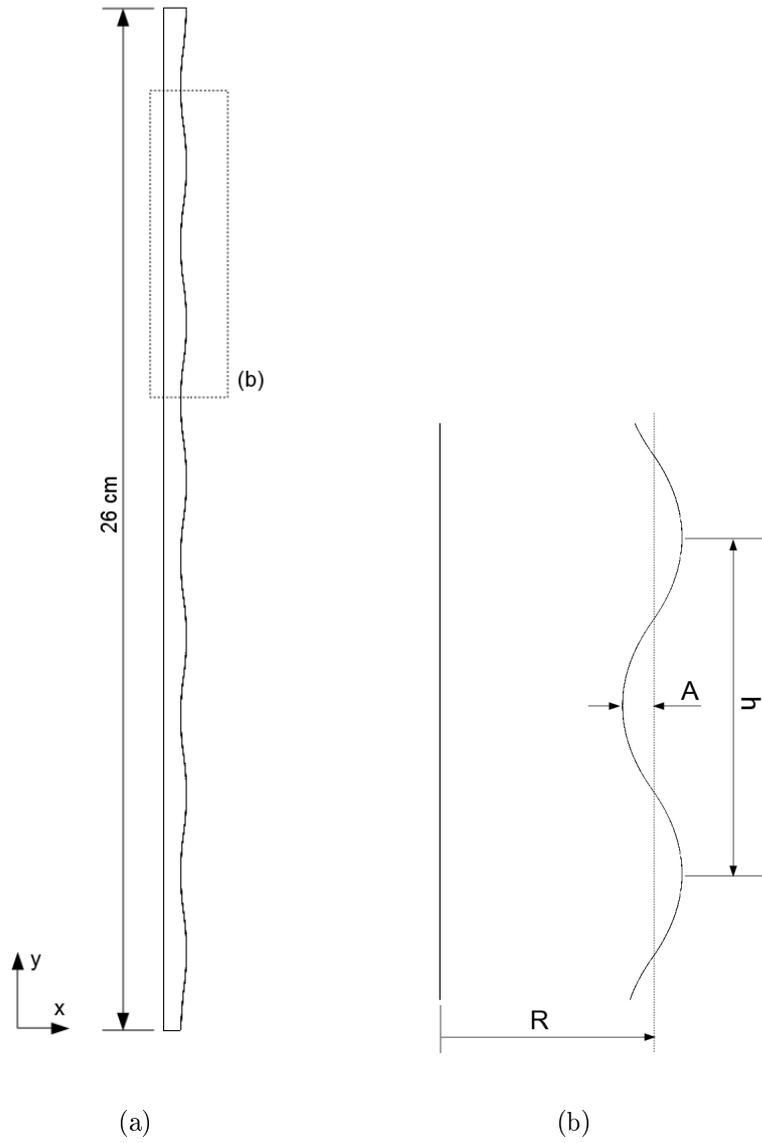


Figure 10: The computational domain for case of 2-dimensional rising drop in a periodically constricted capillary tube. (a) Full domain. (b) Plan view showing dimensional details (not to scale).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

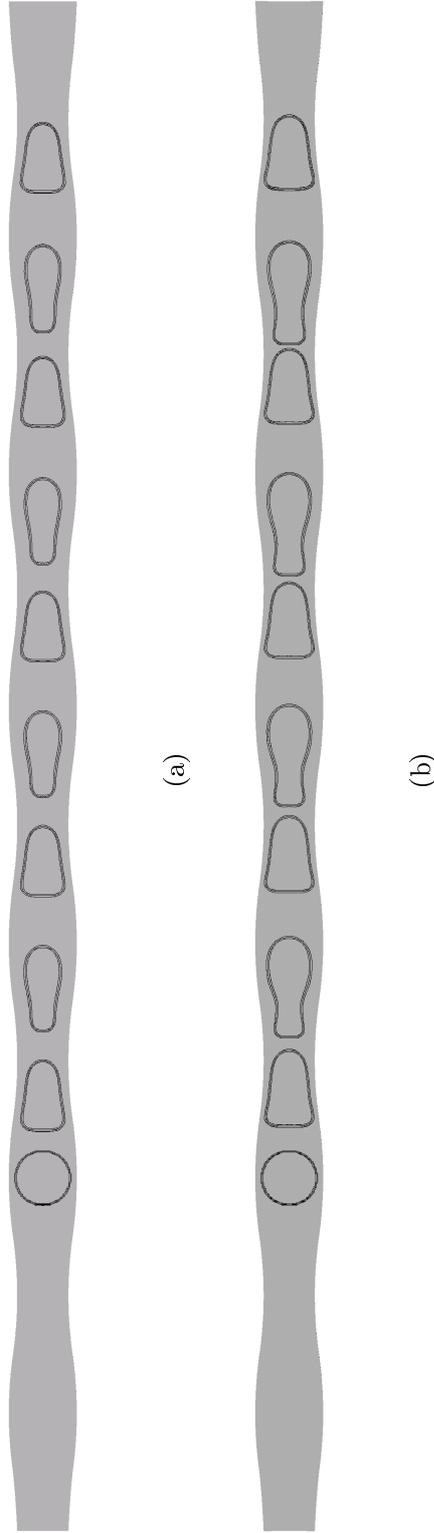


Figure 11: The volume-fraction contours 0.05, 0.5 and 0.95 of the rising drop. (a) Plots for 2-dimensional planar simulation from $t = 0$ to $t = 3937.5$ at time interval of 437.5. (b) Plots for axisymmetric 3-dimensional simulation from $t = 0$ to $t = 4823.1$ at time interval of 535.9.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

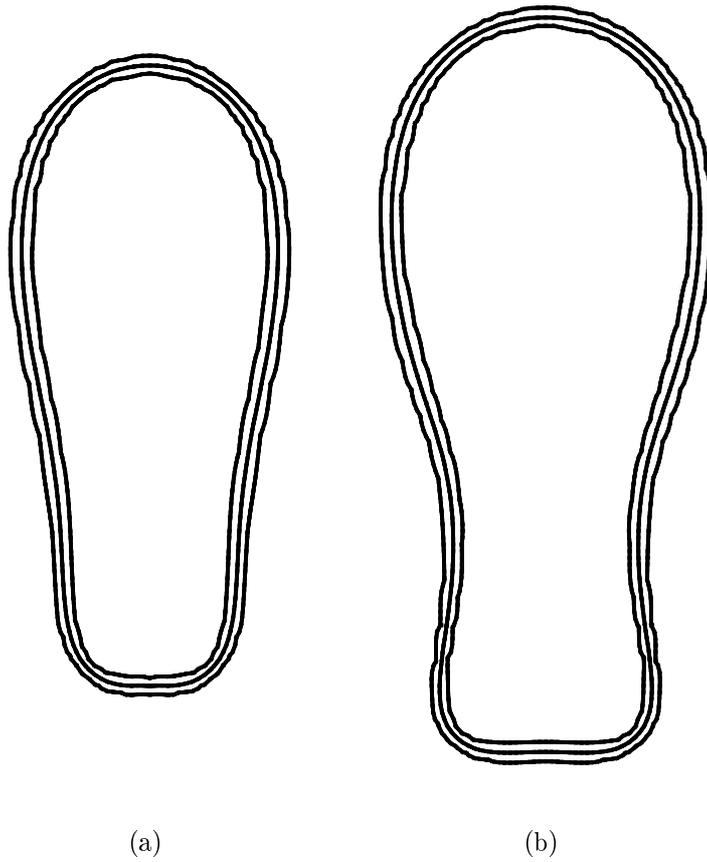


Figure 12: The volume-fraction contours 0.05, 0.5 and 0.95 of the rising drop. (a) Plot at $t = 1750$ for the *2-dimensional planar* simulation. (b) Plot at $t = 2143.6s$ for the *axisymmetric 3-dimensional* simulation.