

RODRIGO LISITA RIBERA

**UNDERSTANDING THE DYNAMICS OF
GAS-LIQUID-SOLID CONTACT POINTS**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2015

RODRIGO LISITA RIBERA

**UNDERSTANDING THE DYNAMICS OF
GAS-LIQUID-SOLID CONTACT POINTS**

Tese apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **DOUTOR EM ENGENHARIA MECÂNICA**.

Área de concentração: Transferência de Calor e Mecânica dos Fluidos.

Orientador: Prof. Dr. Aristeu da Silveira Neto

Co-Orientador: Berend van Wachem (Imperial College London)

Uberlândia - MG

2015

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

R484u Ribera, Rodrigo Lisita, 1981-
2015 Understanding the dynamics of gas-liquid-solid contact points /
Rodrigo Lisita Ribera. - 2015.
228 f. : il.

Orientador: Aristeu da Silveira Neto.

Coorientador: Berend van Wachem.

Tese (doutorado) - Universidade Federal de Uberlândia, Programa
de Pós-Graduação em Engenharia Mecânica.

Inclui bibliografia.

1. Engenharia mecânica - Teses. 2. Escoamento multifásico - Teses.
3. Tensão superficial - Teses. I. Silveira Neto, Aristeu da, 1955- II.
Wachem, Berend van. III. Universidade Federal de Uberlândia.
Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.

CDU: 621

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my supervisor Dr. Aristeu da Silveira Neto for his continuous support and guidance. His efforts over the years allowed all his students to have an excellent research environment.

I would also like to thank my co-supervisor, Dr. Berend van Wachem, for his guidance and collaboration, for receiving me in his research group and always pushing me a bit more, both in my research and in my running.

I would like to thank Millena Villar for the intense collaboration, Ricardo Serfaty, from PETROBRAS, for his support and suggestions along these years, as well as Dr. Alexandre Roma, from IME-USP, Dr. Castello and his group at USP-São Carlos, for their suggestions.

I would like to thank Dr. Fabian Denner at Imperial College London, for the excellent research and cooperation, and for waiting for me in a Friday evening with my flat keys when I arrived in London.

A special thanks goes to John Pennefather and Martin, who helped me to improve my spoken English and for sharing uncountable sugar-free double-essposos, as well as all my other colleagues whom I shortly shared *Room 600* at Imperial College. All of them were always friendly and supportive.

I cannot forget all those who shared *Room 304* with me and also some years of our lives, Damasceno, Denise, Diego, João, Mariana, Renato, Renato, and those who also shared a flat with me, Leonardo, Pivello and Sigeo. They have all become great friends.

I would like to wholeheartedly thank my family, specially my in-laws, for their support, my parents and my sisters, for their unconditional love. They have been an invaluable part of my life and I will always be indebted to them.

Finally, I would like to specially thank my wife, Mariana, for her love, support and patience, and Mateus, who had to wait for some years to born due to my Phd.

To the Federal University of Uberlândia and the School of Mechanical Engineering, and Imperial College London, for providing the computational support for the fulfilling of this work;

To CAPES and PETROBRAS for the financial support.

RIBERA, R. L., **Understanding the dynamics of gas-liquid-solid contact points.** 2015. pages 228. Tese de Doutorado, Universidade Federal de Uberlândia, Uberlândia.

RESUMO

Este trabalho objetiva o estudo, implementação numérica e simulação computacional de escoamentos multifásicos, com o método euleriano VOF, e estudo do contato fluído-líquido-sólido, conhecido como ponto de contato triplo, em um código com refinamento por malha adaptativa estruturada (AMR3D). O trabalho foca no desenvolvimento de modelos matemáticos, técnicas computacionais e validação destas, para modelar eficientemente e com acuidade a dinâmica do contato fluído-líquido-sólido. O principal objetivo do trabalho é o de desenvolver um modelo para a captura da dinâmica de escoamentos multifásicos no contato fluído-líquido-sólido. Estes modelos devem ser: (i) precisos, (ii) robustos e (iii) mais genéricos possíveis. As previsões obtidas do modelo são validadas com resultados de experimentos ou da literatura. São realizados desenvolvimentos sobre um código tridimensional para a simulação de escoamentos multifásicos, com malha de refinamento adaptativo e um modelo para o ponto de contato entre as fases fluída e sólida, genérico e robusto o suficiente para simular paredes que não sejam planas, impostas através do método da Fronteira Imersa, e que seja capaz de representar a física do problema. Um melhor entendimento da dinâmica de gotas em contato com as paredes é obtido, com o estudo dos parâmetros relevantes que afetam a física da linha de contato.

Palavras-chave: Escoamentos Multifásicos, método VOF, Ângulo de Contato, Ponto de contato, Força de Tensão Superficial, método da Fronteira Imersa

RIBERA, R. L., **Understanding the dynamics of gas-liquid-solid contact points** 2015. pages 228. Doctor Thesis, Universidade Federal de Uberlândia, Uberlândia.

ABSTRACT

The objective of this work is to study, implement and numerical simulate the VOF multiphase flow model and to study the contact between the fluid-liquid-solid, know as the triple point region, in a structured adaptative mesh refinement code (AMR3D). The work focus on developing mathematical models, computational techniques and validation of these, to efficiently and accurately model the fluid-liquid-solid contact dynamics. The main objective of the project is to construct a model capturing the fluid-liquid-solids contact dynamics in multiphase flows. This model should be: (i) accurate, (ii) robust and (iii) as generic as possible. It is also aimed to validate the predictions of the model with results obtained from experiments or from the literature. Developments are made in a three-dimensional computational code for the simulation of multiphase flows, with adaptative mesh refinement and a model for the contact point between the fluid phases and the solid that is general and robust enough to simulate walls other than flat surfaces, imposed through the Immersed Boundary Method, and that is capable of representing the physics of the problem. A better understanding of the dynamic of droplets at walls is obtained, with the study of the relevant parameters that affects the physics of the contact line.

Keywords: Multiphase Flows, VOF method, Contact Angle, Contact Point, Surface Tension force, Immersed Boundary method

List of Figures

1.1	Test section photographs of upward air-water flow exhibiting the visual flow pattern features (from left to right): bubbly; spherical cap; stable slug; unstable slug; semi annular; annular. Source: Rosa, Flora and Souza (2012) . . .	2
1.2	Flow patterns in a horizontal pipe. Source: Loilier (2006)	3
2.1	The MAC method. Particles are used to identify full cells and interface cells. Source: Harlow and Welch (1965)	7
2.2	Multi-fluid cell flagging example: (a) classification for fluids 1 and 2; (b) classification only for fluid 1; (c) classification only for fluid 2. Source: McKee et al. (2008)	7
2.3	Steps required to perform surface merging. Source: McKee et al. (2008) . . .	7
2.4	Steps required to perform surface splitting. Source: McKee et al. (2008) . . .	7
2.5	VOF representation of an interface. The information from a known geometry (a) should be transformed in a scalar in each eulerian computational cell (b), which is later reconstructed by a numerical scheme (c-d). Source: Pilliod and Puckett (2004)	9
2.6	Surface tension is an intermolecular force which contract the surface, being responsible for the shape of liquid droplets. Source: www.butane.chem.uiuc.edu	11
2.7	The molecules exposed at the surface are pulled inward, as they do not have neighboring molecules in all directions to provide a balanced net force. Source: www.fusedglass.org	11
2.8	Three different possible wetting states. Source: Bonn et al. (2009)	12
2.9	Illustration of contact angles formed by sessile liquid drops on a smooth homogeneous solid surface. Source: www.biolinscientific.com	13
2.10	The degree to which wetting occurs (i.e., wettability) is determined by the cohesive forces of the liquid molecules among themselves and the adhesive forces that result from the molecular interactions between the liquid and the solid as illustrated in the figure. (In real life, the molecules are not so neatly organized). Source: www.ramehart.com	14
2.11	The equilibrium between the solid, liquid and vapor phase interfaces is expressed by the Young equation. Source: Burris (1999)	14

2.12	Wetting states defined by: a) Wenzel model; b) Cassie-Baxter model; c) transition from Cassie-Baxter to Wenzel; and d) <i>hemi-whicking</i> . Source: www.commons.wikimedia.org/wiki/User:Spinal83	16
2.13	If water is carefully added to the droplet with a syringe, the droplet volume and contact angle increase, and the same contact area is maintained until the droplet begins to advance, at a constant advancing contact angle (left); if water is carefully withdrawn from the droplet with a syringe, the droplet decreases in volume and contact angle, maintaining the same contact area with the surface until it begins to recede, with a constant contact angle θ_R (right). Source: www.ramehart.com	18
2.14	Advancing and receding contact angles captured by tilting base method. Source: www.ramehart.com	19
2.15	Schematic representation of the velocity dependence of the contact angle. source: Blake (2006)	20
2.16	Interpretation of slip length. Source: Lauga, Brenner and Stone (2005)	22
3.1	Example of mesh refinement in a two-phase flow. source: N3s (2007)	32
3.2	Computational cell; A) position of the scalar variables; B) position of the vector variables in the x direction; C) position of the vector variables in the y direction; D) position of the vector variables in the z direction. source: N3s (2007)	33
3.3	Computational mesh with ghost cells represented by the dashed lines and staggered variables represented by the arrows at the cell faces. source: Villar (2007)	33
3.4	An example of block structured adaptive mesh refinement properly nested. Source: Villar (2007).	35
3.5	An example of block structured adaptive mesh refinement violating the nesting rules: the fine block in grid (a) does not lie on vertices of the coarse level block and the block in the finest level in grid (b) touches the boundary of a block in the next coarse level which is not part of the domain boundary. Source: Villar (2007).	36
3.6	Example of adaptive mesh refinement based on the vorticity magnitude. Source: Pivello (2012)	36
3.7	Interpolation stencil. * value at the finer level, • value at the coarse level, Δ extrapolated value from the finer level, \square interpolated value from the coarse level, \circ final value at the ghost cell. source: Villar (2007)	37
3.8	A 3D cube (left) and the right upper front octant (right) used for the 27 Cell discretization method. Source: Lam (2009)	45
3.9	The 8 normal components in i - direction, where the big arrows are the average of 4 corresponding small ones. Source: Lam (2009)	47

3.10	125 Cells discretization in a) i, j -planes and b) i, k - planes. Source: Lam (2009)	48
3.11	The 7x3 fluid stencil used to calculate fluid height. Source: Afkhami and Bussmann (2007)	50
3.12	A general plane	58
3.13	Volume under a general plane cutting an Eulerian cell	61
3.14	Volumes that should be subtracted (yellow) in Eq.(3.104), as they are outside the Eulerian cell	62
3.15	Volumes that should be added back (blue) in Eq.(3.104), if they are subtracted twice	62
3.16	Volumes that should be subtracted (gray) in Eq.(3.105), as they are outside the Eulerian cell	63
3.17	Volume represented by Eq.(3.106), considering the normal component at $x - direction$ different than zero	63
3.18	(a) Change in the normal component due to the velocity flow field: $n_{1*} > n_1$ (b) Due to the advection of the normal component, the interface changes its inclination, and consequently the volume fraction inside the Eulerian cell. . .	66
3.19	left face analysis (a) before movement (b) after movement to right (c) after movement to left	67
3.20	(a) Volume of the cell (b) Volume transfered to left	68
3.21	Right face analysis (a) before movement (b) after movement to right (c) after movement to left	69
3.22	$U_R > 0$, but no volume is transfered to right	69
3.23	$U_L > 0$ and $U_R > 0$ (a) Initial Volume (b) Volume to right (c) Volume that remains in the cell	71
3.24	$U_L < 0$ and $U_R < 0$ (a)Initial Volume (b)Volume to left (c)Volume that remains in the cell	72
3.25	$U_L > 0$ and $U_R < 0$ (a)Initial Volume (b)Volume that remains in the cell . .	73
3.26	A contact point cell (cicle) and an adjacent cell (squares): a) $\theta < 90^\circ$ and b) $theta > 90^\circ$ Identification of contact point, contact cells and adjacent ones. Source: Afkhami and Bussmann (2009)	74
3.27	The ghost cells heights $h_{i,j-1,0}$, $h_{i,j,0}$ and $h_{i,j+1,0}$ are determined so that the interface orientation at the contact point corresponds to the contact point normal. Source: Afkhami and Bussmann (2009)	75
3.28	The height $h_{j=0}$ in the ghost cell is determined so that the interface orientation at the contact point corresponds to \mathbf{n}_{cl} . Source: Afkhami and Bussmann (2009)	75
3.29	\mathbf{n}_{cl-xy} is the projection of the contact point normal \mathbf{n}_{cl} onto the $x - y$ plane, defined by an outward normal \mathbf{n}_s . Source: Afkhami and Bussmann (2009) .	76

4.1	Volume fraction initialization errors as a function of grid resolution for a sphere of radius $r = 2$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$	86
4.2	Initialization time as a function of grid resolution for a sphere of radius $r = 2$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$	87
4.3	Color function for the sphere with radius $r = 2[m]$, centered in a domain of size $\Omega = 8^3[m^3]$, initialized in the AMR3D code with the López et al. (2009) method, for maximum mesh resolution ranging from $\Delta = 2$ to $\Delta = 1/16$	88
4.4	Volume fraction initialization errors as a function of grid resolution for a sphere of radius $r = 0.25[m]$, centered in a domain of size $\Omega = 1^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$	89
4.5	Initialization time as a function of grid resolution for a sphere of radius $r = 0.25[m]$, centered in a domain of size $\Omega = 1^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$	90
4.6	Volume fraction initialization errors as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$	91
4.7	Initialization time as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$	92
4.8	Color function of the ellipsoid with semi-axes $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$, initialized in the AMR3D code with the Points method, for maximum mesh resolution ranging from $\Delta = 2$ to $\Delta = 1/16$	93

4.9 Volume fraction initialization errors as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1.0^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$ 94

4.10 Initialization time as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$ 95

4.11 Volume fraction initialization errors as a function of grid resolution for a Torus of parameters $R = 2[m]$, $r = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 50^3$ 96

4.12 Initialization time as a function of grid resolution for a Torus of parameters $R = 2[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 50^3$ 97

4.13 Color function of the Torus with parameters $R = 2[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m]$, initialized in the AMR3D code with the Points method, for maximum mesh resolution ranging from $\Delta = 2$ to $\Delta = 1/16$ 98

4.14 Three dimensional Shear flow test performed with the AMR3D code 99

4.15 L_2 error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method. 103

4.16 L_2 error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method and the smoothed color function. 103

4.17 L_∞ error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method. 104

4.18 L_∞ error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method and the smoothed color function. 104

4.19	L_2 error norm as a function of the mesh refinement for the normal vector of a sphere with $r = 0.25[m]$, computed with the Least Squares, the Young and the Shirani methods.	106
4.20	L_∞ error norm as a function of the mesh refinement for the normal vector of a sphere with $r = 0.25[m]$, computed with the Least Squares, the Young and the Shirani methods.	107
4.21	L_2 error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3 [m^3]$. Comparison between the non-smoothed and the smoothed color function.	109
4.22	L_∞ error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3 [m^3]$. Comparison between the non-smoothed and the smoothed color function.	110
4.23	L_2 error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3 [m^3]$. Comparison between the Young, Shirani and Least Squares methods.	111
4.24	L_∞ error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3 [m^3]$. Comparison between the Young, Shirani and Least Squares methods.	112
4.25	L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[m]$, computed with the Least Squares method. Analysis of the relation between the stencil sizes of the normal and the curvature. . .	114
4.26	L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[m]$ computed with the Least Squares method for the cases: LS s3 s5, LS s5 s5, LS s3 s7 and LS s5 s7.	115
4.27	L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[m]$, computed with the Least Squares method. Analysis of the use of the geometrical weighting factor on the case LS s3 s5.	116
4.28	error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[m]$ computed with the Least Squares method. Analysis of the use of the smoothed color function on the case LS s3 s5g1.	117
4.29	L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[m]$ computed with the Least Squares method, computational stencil size $s = 7$ for the normal and $s = 5$ for the curvature, and geometrical weighting factor $0 \leq g \leq 2$	119
4.30	error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[m]$ computed with the Least Squares method and cases LS s5 s7.120	

4.31	error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$ computed with the Least Squares method. Comparison between cases LS s3;s5 g1 (smoothed), LS s5;s7g2 and LS s5g1;s7 g2 (smoothed).	121
4.32	L_2 and L_∞ error norms and mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$ computed with the Paraboloid method, computational stencil sizes 3, 5 and 7, and smoothed color function for the stencil size 5.	123
4.33	L_2 and L_∞ error norms and mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$ computed with the Shirani and 27 Cells methods with and without the smoothed color function.	125
4.34	L_2 and L_∞ error norms and mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$ computed with the Height Function method.	127
4.35	L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$. Comparison between different curvature computation methods.	128
4.36	L_∞ error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$. Comparison between different curvature computation methods.	129
4.37	Mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$. Comparison between different curvature computation methods.	129
4.38	a) L_2 error norm x mesh refinement and b) L_∞ error norm x mesh refinement of the curvature for an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$. Comparison between the Height Function coupled with different methods	130
4.39	L_2 error norm as a function of the mesh refinement for the curvature computation of an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$. Comparison between different curvature computation methods	131
4.40	L_∞ error norm as a function of the mesh refinement for the curvature computation of an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$. Comparison between different curvature computation methods	132
4.41	Effect of the surface tension coefficient on the error of the maximum velocity $ \mathbf{u} _{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. Data from Tab. 4.23	137
4.42	Effect of the time step magnitude on the error of the maximum velocity $ \mathbf{u} _{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. Data from Tab. 4.24	138

4.43	Evolution of the RMS velocity around a circular droplet in theoretical equilibrium for $La = 120$. a) Result comparison with different curvature computation methods b) Result comparison excluding the Paraboloid method c) Result comparison between the Height function coupled with the Shirani and coupled with the Paraboloid method. Time and velocity are made non-dimensional using T_μ and U_σ as reference scales, respectively.	140
4.44	Evolution of the RMS velocity around a circular droplet in theoretical equilibrium for $La = 1200$. a) Result comparison with different curvature computation methods b) Result comparison excluding the Paraboloid method c) Result comparison between the Height function coupled with the Shirani and coupled with the Paraboloid method. Time and velocity are made non-dimensional using T_μ and U_σ as reference scales, respectively.	141
4.45	Evolution of the RMS velocity around a circular droplet in theoretical equilibrium for $La = 12000$. a) Result comparison with different curvature computation methods b) Result comparison excluding the Paraboloid method c) Result comparison between the Height function coupled with the Shirani and coupled with the Paraboloid method. Time and velocity are made non-dimensional using T_μ and U_σ as reference scales, respectively.	142
4.46	Observed bubble terminal shapes for the rising bubble experiment. Source: Hua and Lou (2007)	143
4.47	Time evolution of the Reynolds number for simulations with the Front-Tracking scheme from Pivello (2012) and with the VOF method with the Height Function and the Least Squares curvature computation methods.	146
4.48	Color function. $x - z$ slice at $y = 1.5[m]$	147
4.49	Normal x component. $x - z$ slice at $y = 1.5[m]$	148
4.50	Normal x. Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; <i>Dashed line: Conventional wall; Continuous line: IB wall</i>	148
4.51	Normal z. $x - z$ slice at $y = 1.5[m]$	149
4.52	Normal z. Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; <i>Dashed line: Conventional wall; Continuous line: IB wall</i>	149
4.53	Curvature. <i>Conventional wall. $x - z$ slice at $y = 1.5[m]$</i>	150
4.54	Curvature. <i>IB wall. $x - z$ slice at $y = 1.5[m]$</i>	150
4.55	Curvature. Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; <i>Dashed line: Conventional wall; Continuous line: IB wall</i>	150
4.56	<i>Conventional wall.</i>	151
4.57	<i>IB wall.</i>	151
4.58	Force x. $x - z$ slice at $y = 1.5[m]$	151

4.59	Force x. Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; <i>Dashed line: Conventional wall; Continuous line: IB wall</i>	151
4.60	<i>Conventional wall.</i>	151
4.61	<i>IB wall.</i>	151
4.62	Force z. $x - z$ slice at $y = 1.5[m]$	151
4.63	Force z. Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5[m]$; <i>Dashed line: Conventional wall; Continuous line: IB wall</i>	152
4.64	x-z plane view ($y = 1.5[m]$) of the final position of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution: a) $\Delta = \frac{1}{8}$; b) $\Delta = \frac{1}{16}$; c) $\Delta = \frac{1}{32}$; d) $\Delta = \frac{1}{64}$; e) $\Delta = \frac{1}{128}$	155
4.65	x-z plane view ($y = 1.5[m]$) of the final position of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution: a) $\Delta = \frac{1}{8}$; b) $\Delta = \frac{1}{16}$; c) $\Delta = \frac{1}{32}$; d) $\Delta = \frac{1}{64}$; e) $\Delta = \frac{1}{128}$	156
4.66	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i>	157
4.67	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for mesh with maximum resolution $\Delta = \frac{1}{128}$	158
4.68	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. No-slip boundary condition.	159
4.69	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. Free-slip boundary condition.	159
4.70	Side View of the time evolution of the interface with No-Slip and Free-Slip boundary conditions	160
4.71	Top view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions	161
4.72	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. No-Slip boundary condition. . .	162
4.73	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. Free-Slip boundary condition. .	162

4.74	contact angle measured at the end of the simulation for $\theta_E = 60^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{128}$	163
4.75	contact angle measured at the end of the simulation for $\theta_E = 60^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{128}$ (DropSnake measurement).	163
4.76	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for mesh with maximum resolution $\Delta = \frac{1}{64}$	164
4.77	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-slip boundary condition.	165
4.78	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-slip boundary condition.	165
4.79	Side View of the time evolution of the interface with No-Slip and Free-Slip boundary conditions	166
4.80	Top view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions	167
4.81	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-Slip boundary condition. . .	168
4.82	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-Slip boundary condition. .	168
4.83	contact angle measured at the end of the simulation for $\theta_E = 45^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$	169
4.84	contact angle measured at the end of the simulation for $\theta_E = 45^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$ (DropSnake measurement).	169
4.85	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for mesh with maximum resolution $\Delta = \frac{1}{64}$	170
4.86	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 148^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-slip boundary condition.	171
4.87	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 148^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-slip boundary condition.	171

4.88	Side view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions	172
4.89	Top view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions	173
4.90	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-Slip boundary condition. . .	174
4.91	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 148^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-Slip boundary condition. .	174
4.92	contact angle measured at the end of the simulation for $\theta_E = 148^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$	175
4.93	contact angle measured at the end of the simulation for $\theta_E = 148^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$ (DropSnake measurement).	175
4.94	x-z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 30^\circ$	177
4.95	Top view (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)	178
4.96	x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 30^\circ$	179
4.97	$\theta_N = 32.0^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 30^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$	180
4.98	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for $\theta = 30^\circ$	180
4.99	x-z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$	181
4.100	Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)	182
4.101	x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$	183
4.102	$\theta_N = 49.6^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 45^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$	184
4.103	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for $\theta = 45^\circ$	184

4.104	x-z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$	185
4.105	Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)	186
4.106	x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$	187
4.107	$\theta_N = 61.6^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 60^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$	188
4.108	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for $\theta = 60^\circ$	188
4.109	x-z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 120^\circ$	189
4.110	Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)	190
4.111	x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 120^\circ$	191
4.112	$\theta_N = 121.1^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 120^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$	192
4.113	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for $\theta = 120^\circ$	192
4.114	x-z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 135^\circ$	193
4.115	Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)	194
4.116	x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 135^\circ$	195
4.117	$\theta_N = 133.5^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 135^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$	196
4.118	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for $\theta = 135^\circ$	196
4.119	x-z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 150^\circ$	197
4.120	Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)	198

4.121	x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 150^\circ$.	199
4.122	$\theta_N = 149.4^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 150^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$.	200
4.123	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for $\theta = 150^\circ$.	200
4.124	Eulerian domain with the mesh used for the simulation, the immersed boundary and the tag of the Eulerian cells: 0 = inside cells; 1 = interface cells; 2 = outside cells.	201
4.125	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 30^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.	202
4.126	x-y plane view ($z = 1.0[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 30^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.	203
4.127	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 30^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.	204
4.128	$\theta_N = 32.6^\circ$ measured at $t = 5.00[s]$ for mesh with maximum resolution $\Delta = \frac{1}{64}$.	205
4.129	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for mesh with maximum resolution $\Delta = \frac{1}{64}$.	205
4.130	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.	206
4.131	x-y plane view ($z = 1.0[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.	207
4.132	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.	208
4.133	$\theta_N = 60.3^\circ$ measured at $t = 5.00[s]$ for mesh with maximum resolution $\Delta = \frac{1}{64}$.	209
4.134	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for mesh with maximum resolution $\Delta = \frac{1}{64}$.	209
4.135	x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 135^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.	210

4.136	x-y plane view ($z = 1.0[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 135^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$	211
4.137	x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 135^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. . .	212
4.138	$\theta_N = 60.3^\circ$ measured at $t = 5.00[s]$ for mesh with maximum resolution $\Delta = \frac{1}{64}$	213
4.139	log of the error in volume conservation (ϵ_{vol}) vs <i>time</i> for mesh with maximum resolution $\Delta = \frac{1}{64}$	213

List of Tables

2.1	Physical parameters estimated for water at room temperature for the Interface Formation Model (SIBLEY; SAVVA; KALLIADASIS, 2012)	26
3.1	Some δ approximations presented in the literature	40
3.2	Coordinates of the vertices of the triangles from the PLIC reconstruction method	54
4.1	Input parameters for the initialization analysis of a sphere with radius $r = 2[m]$ and $r = 0.25[m]$, an ellipsoid with semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$ and $a = 0.30[m]$, $b = 0.25[m]$ and $c = 0.20[m]$ and a torus with $R = 2.0$ and $r = 1.0$, all centered in a domain of size $\Omega = 8.0^3[m^3]$	85
4.2	Convergence ratio for the error in the color function initialization of a sphere with radius $r = 2.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$	87
4.3	Convergence ratio for the error in the color function initialization of a sphere with radius $r = 0.25[m]$, centered in a domain of size $\Omega = 1^3[m^3]$	90
4.4	Convergence ratio for the error in the color function initialization of an ellipsoid with semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$	92
4.5	Convergence ratio for the error in the color function initialization of an ellipsoid with semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1.0^3[m^3]$	95
4.6	Convergence ratio for the error in the color function initialization Torus with parameters $R = 2.0[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = [8]^3[m^3]$	97
4.7	Mass loss for the VOF method in a 3D Shear Flow test. $T = 3.0[s]$ and $\Delta t = \frac{\Delta x}{2}[s]$	99
4.8	Convergence ratio for normal vector of a sphere with radius $r = 0.25[m]$, computed with the Least Squares method.	105
4.9	Convergence ratio for the normal vector of a sphere computed with the Least Squares, the Young and the Shirani methods.	108
4.10	Convergence ratio for normal vector of an ellipsoid, computed with the Least Squares, the Young and the Shirani methods.	108

4.11	Test cases relating the stencil size for the normal and for the curvature, for the curvature computation of a sphere with $r = 0.25[m]$ with the Least Squares method.	113
4.12	Test cases for Least Squares curvature computation of a sphere with radius $r = 0.25[m]$, with stencil size 5 for the normal and 3 for the curvature, with and without geometrical weighting factor.	115
4.13	Test cases for Least Squares curvature computation of a sphere with radius $r = 0.25[m]$, with stencil size 7 for the normal and 5 for the curvature, with and without geometrical weighting factor.	118
4.14	Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Least Squares method and stencil sizes LS s3;s5 and LS s5;s7	122
4.15	Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Paraboloid method and stencil sizes 3, 5 and 7.	122
4.16	Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Shirani and the 27 Cells methods with and without the smoothed color function	124
4.17	Quantity of inconsistent points in relation to the mesh size for the Height Function method in a sphere with $r = 0.25[m]$	126
4.18	Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Height Function method coupled with the Shirani, the Paraboloid and the 27 Cells methods.	126
4.19	Quantity of inconsistent points as a function of the mesh refinement for the Height Function method. Curvature computation on an ellipsoid	131
4.20	Convergence ratio for the L_2 and the L_∞ error norms for the curvature of an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$	132
4.21	Error in maximum velocity $ \mathbf{u} _{max}$ and relative pressure jump $E(\Delta P)_{max}$ after one and 50 time steps for a 3D inviscid static drop in equilibrium with different curvature estimates.(Uniform mesh)	134
4.22	Effect of the viscosity ratio on the error of the maximum velocity $ \mathbf{u} _{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 and 100 time steps for the viscous drop in equilibrium. The fluid density ration is 10^3 , with density inside ($\rho_1 = 1[kg/m^3]$) and density outside ($\rho_2 = 10^{-3}[kg/m^3]$) and $\sigma = 73.0[N/m]$. The drop has radius $R = 2[m]$ and is centered in a domain $\Omega = [8]^3[m^3]$ with mesh 40^3 ($R/h = 10$). The time step is constant and equal to $10^{-6}[s]$. The fluid viscosity inside the drop ($\mu_1 = 10^{-2}[Pa.s]$) and outside the viscosity is allowed to vary. (Uniform mesh)	135

- 4.23 Effect of the surface tension coefficient on the error of the maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. The fluid density ration is 10^3 , with density inside ($\rho_1 = 1[kg/m^3]$) and density outside ($\rho_2 = 10^{-3}[kg/m^3]$). The drop has radius $R = 2[m]$ and is centered in a domain $\Omega = [8]^3[m^3]$ with mesh 40^3 ($R/h = 10$). The time step is constant and equal to $10^{-6}[s]$. The viscosity ratio is 10, with $\mu_1 = 10^{-2}[Pa.s]$ and $\mu_2 = 10^{-3}[Pa.s]$. (Uniform mesh) 136
- 4.24 Effect of the time step magnitude on the error of the maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. The fluid density ration is 10^3 , with density inside ($\rho_1 = 1[kg/m^3]$) and density outside ($\rho_2 = 10^{-3}[kg/m^3]$). The drop has radius $R = 2[m]$ and is centered in a domain $\Omega = [8]^3[m^3]$ with mesh 40^3 ($R/h = 10$). The viscosity ratio is 10, with $\mu_1 = 10^{-2}[Pa.s]$ and $\mu_2 = 10^{-3}[Pa.s]$ and $\sigma = 730.0[N/m]$. (Uniform mesh) 136
- 4.25 Comparison of terminal shapes and Reynolds number reported by Hua and Lou (2007), Pivello (2012) and the present work. 145
- 4.26 Mesh resolution, time step, error in the initialized vof volume and contact angle error for $\theta = 60^\circ$ and $1/8 \leq \Delta \leq 1/128$. (No-Slip boundary condition.) . 154

List of Symbols

Vectors and tensors are either denoted in **bold** or in tensor notation by subscripts i (vectors) or ij tensors, dependent on the context. All units are SI units.

Roman letters

A_1, A_2	Total area of solid-liquid interface and liquid-air interface, respectively
A_{ij}, \mathbf{A}	Coefficient matrix of an equation system
b_i, \mathbf{b}	Right-hand side of an equation system
B_{ij}, \mathbf{B}	Coefficient submatrix of an equation system
d	Distance between two points
F	Scalar representing the volume fraction of one phase inside a computational cell
\tilde{F}	Smoothed color function
f_i, \mathbf{f}	Body force per unit volume
\mathbf{F}_σ	Surface tension force
g	Gravitational acceleration
\mathbf{g}	Gravitational vector
G	Geometrical weighting factor
H	Mean curvature
H	Heaviside function
h_i	First derivative of fluid height
h_{ij}	Second derivative of fluid height
I	Kronecker tensor
l	Level of refinement
l_λ	Capillary length
\mathbf{n}	Normal vector
p	Pressure
\mathbf{P}	Generic point on a mesh
r	Roughness factor
r	Refinement ratio
R	Characteristic length scale
R	Radius of the droplet
R_1, R_2	Principal curvature radii

t	time
\mathbf{t}	Tangent vector
\mathbf{u}	Velocity vector in the Eulerian framework
\mathbf{U}	Velocity vector in the Lagrangian framework
\mathbf{U}_{CL}	Contact line velocity vector
u, v, w	Velocity components in x-, y- and z-directions, respectively, in the Eulerian framework

Greek letters

α	Response of the interface to surface tension gradients
α_i	Time discretization coefficients, $i = 0, 1, 2$
β_i	Time discretization coefficients, $i = 0, 1$
β_i	Constant in the temporal discretization method
Δ	Maximum grid resolution
Δt	Time step
Δx	Grid spacing in the x-direction
Δy	Grid spacing in the y-direction
Δz	Grid spacing in the z-direction
ϵ	Relative error between two variables
ϵ_{vol}	Error in the volume conservation
ϵ_θ	Error between the numerical and the theoretical contact angles
\mathbf{e}_{rr}	Error between theoretical and computed normal vectors
γ	Inverse surface layer compressibility
γ	Surface tension coefficient
γ	Over relaxation factor
γ	Angle between the rotated normal and velocity vector
κ	Interface curvature
λ	Slip length
μ	Viscosity
∇	Gradient operator
∇^2	Laplacian operator
ν	Kinematic viscosity
ω	Ratio between two consecutive time steps
Ω	Size of the domain
ϕ	Generic scalar
Φ	Generic scalar
$\phi(x, y, z)$	Implicit function
ϕ_s	fraction of the solid/liquid interface below the drop
ρ	Density

σ	Surface tension coefficient
θ	Contact angle
θ_A	Advancing contact angle
$\theta_{A,S}$	Static advancing contact angle
$\theta_{A,R}$	Static receding contact angle
θ_c	Cassie apparent contact angle
θ_D	Dynamic contact angle
θ_E	Equilibrium contact angle
θ_i	Constant in the temporal discretization model
θ_{mda}	Maximum advancing contact angle
θ_{mdr}	Minimum receding contact angle
θ_N	Numerical contact angle
θ_r	Wenzel's apparent contact angle
θ_R	Receding contact angle
θ_S	Static contact angle
θ_i	Time discretization coefficients, $i = 0, 1, 2$

Subscripts

<i>adv</i>	Index referring to the advective term
<i>C</i>	Index referring to the continuous phase
<i>cap</i>	Index referring to the capillary term
<i>cur</i>	Index referring to the value of the variable at the current time step
<i>D</i>	Index referring to the dispersed phase
<i>diff</i>	Index referring to the diffusive term
<i>LS</i>	Index referring to the liquid-solid interface
<i>LV</i>	Index referring to the liquid-vapor interface
<i>max</i>	Index referring to the maximum value of the variable
<i>min</i>	Index referring to the minimum value of the variable
<i>prev</i>	Index referring to the value of the variable at the previous time step
<i>VS</i>	Index referring to the vapor-solid interface

Superscripts

n	Present iteration
$n - 1$	Previous iteration
$n + 1$	Next iteration

Non-dimensional numbers

<i>Bo</i>	Bond number
<i>Ca</i>	Capillary number

Eo	Eötvös number
La	Laplace number
M	Morton number
Re	Reynolds Number

Functions

$ \cdot $	Absolute value.
$\ \cdot\ _2$	Euclidean norm, L_2 -norm.
$\ \cdot\ _{max}$	Infinity norm, L_∞ .
$\delta(x)$	Dirac delta function
δ_{ij}	Kronecker delta
$f()$	General function
$\phi()$	General implicit function

Mathematical Operators and Symbols

\int	Integral
$O()$	Order of the truncation error
∇	Nabla operator
Σ	Summation operator

Abbreviations

AMR	Adaptive Mesh refinement
ALE	Arbitrary-Lagrangian-Eulerian
BC	Boundary condition
CDS	Central difference scheme
CFD	Computational Fluid Dynamics
$CLSVOF$	Coupled Level Set and Volume of Fluid
$CNAB$	Crank-Nicholson Adams-Bashforth
$CNLF$	Crank-Nicholson Leap-Frog
CSF	Continuum Surface Force
CSS	Continuum Surface Stress
FC	Front Capturing
FT	Front Tracking
HF	Height Function
IB	Immersed Boundary
IFM	Interface Formation Model
LS	Level-Set
LS	Lest Squares
MAC	Marker and Cell

<i>MCL</i>	Moving Contact Line
<i>MCNAB</i>	Modified Crank-Nicholson Adams-Bashforth
<i>MPM</i>	Material Point method
<i>np</i>	Number of points
<i>nsc</i>	Number of subdomains
<i>PAF</i>	Particle and Forcel
<i>PLIC</i>	Piecewise Linear Interface Calculation
<i>PIC</i>	Particle-in-Cell
<i>RHS</i>	Right hand side
<i>RM</i>	Rotation matrix
<i>RMS</i>	Root mean square
<i>SAMR</i>	Structured Adaptative Mesh Refinement
<i>SBDF</i>	Semi-Backward Difference
<i>SMAC</i>	Simplified Marker and Cell
<i>WLS</i>	Weighted Leasts Squares
<i>VOF</i>	Volume of Fluid

CONTENTS

LIST OF FIGURES	xiii
LIST OF TABLES	xxvi
LIST OF SYMBOLS	xxxi
1 INTRODUCTION	1
1.1 Objectives	1
1.2 Justification	2
1.3 International cooperation	3
1.4 Thesis organization	4
2 Background	5
2.1 Review of two-phase flows and the VOF method	5
2.2 Surface tension	10
2.3 Wetting, contact angle and contact point	12
2.4 Moving Contact Line (MCL)	19
2.5 Dynamic contact angle - empirical models	23
2.6 Dynamic contact angle: mathematical model	24
3 METHODOLOGY	27
3.1 Mathematical Modeling	27
3.1.1 Eulerian formulation	27
3.1.2 Eulerian - Lagrangian coupling	28
3.2 Numerical method: temporal discretization	29
3.3 Numerical method: spatial discretization	31
3.3.1 Structured Adaptive Mesh Refinement	34
3.3.2 Ghost cells	36
3.4 Fractional step method	38
3.5 Surface tension force	39
3.6 Color function initialization	42
3.6.1 Initialization with implicit functions	42

3.6.1.1	The López et al. (2009) initialization method	43
3.6.1.2	Initialization method with points inside the computational cell	43
3.7	Normal and curvature estimation	43
3.7.1	Parker and Young's method	44
3.7.2	27 Cells discretization	45
3.7.3	125 Cell discretization based on Shirani, Ashgriz and Mostaghimi (2005)	48
3.7.4	The Height function method	50
3.7.5	The Paraboloid method	52
3.7.6	Least Squares method	53
3.8	VOF PLIC method	57
3.8.1	Reconstruction Step of the VOF PLIC method	58
3.8.2	The interface reconstruction in the VOF/PLIC method	59
3.8.3	The volume equation	60
3.8.4	The reconstruction step	63
3.9	Lagrangian Propagation of the interface (VOF advection)	64
3.9.1	First Contribution for the change in the interface equation - movement of the flow	64
3.9.2	Second Contribution for the change in the interface equation - movement of the rectangular sides of the volume	66
3.9.2.1	Left face analysis	66
3.9.2.2	Right face analysis	68
3.9.2.3	A combination of the left and right face movement's: $U_L > 0$ and $U_R > 0$	70
3.9.2.4	A combination of the left and right face movement's: $U_L < 0$ and $U_R < 0$	71
3.9.2.5	A combination of the left and right face movement's: $U_L > 0$ and $U_R < 0$	72
3.9.3	Performing the lagrangian propagation of the interface	72
3.10	Numerical procedure to apply the contact angle	73
3.11	Applying the contact angle in the AMR3D Code	76
3.11.1	Determining the direction of displacement of the interface	77
3.12	Changes in the Least Squares system to account for the contact point	78
3.12.1	Proposed change 1 - Use the values of the known normals at the walls	78
3.12.2	Proposed change 2 - Use only internal cells information	79
3.13	A direct forcing based scheme to account for the contact point force	80
4	RESULTS	83
4.1	Verification	84
4.1.1	Analysis of the initialization methods	84

4.1.1.1	Test initialization for a sphere with radius $r = 2.0[m]$ centered in a domain of size $\Omega = 8^3[m^3]$	85
4.1.1.2	Test initialization for a sphere with radius $r = 0.25[m]$ centered in a domain of size $\Omega = 1.0^3[m^3]$	89
4.1.1.3	Test initialization for an ellipsoid with semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8.0^3[m^3]$	91
4.1.1.4	Test initialization for an ellipsoid with semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1.0^3[m^3]$	94
4.1.1.5	Test initialization for a Torus with parameters $R = 2.0[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = [8]^3[m^3]$	96
4.1.2	Analysis of the reconstruction and advection step	99
4.1.3	Normal and Curvature computation - Simulation set-up	100
4.1.4	Normal and Curvature simulation results	101
4.1.4.1	Normal analysis: sphere. Least Squares Method	101
4.1.4.2	Normal analysis: sphere. Comparison between the Least Squares, the Young and the Shirani methods	105
4.1.4.3	Normal analysis: ellipsoid. Comparison between the Least Squares, the Young and the Shirani methods.	108
4.1.4.4	Curvature analysis: sphere. Least Squares Method	112
4.1.4.5	Curvature analysis: sphere. Paraboloid method	122
4.1.4.6	Curvature analysis: sphere. Shirani and 27 Cells	124
4.1.4.7	Curvature analysis: sphere. Height function method	126
4.1.4.8	Curvature analysis: sphere. Comparison between different curvature computation methods	128
4.1.4.9	Curvature analysis: ellipsoid	130
4.1.5	Analysis of a 3D static drop in equilibrium	132
4.1.5.1	Effect of fluid properties and integration time step	134
4.1.6	Parasitic current analysis	139
4.1.7	Ascending bubble	143
4.1.8	Analysis of normal, curvature and surface tension force at the contact point	146
4.1.8.1	Color function	147
4.1.8.2	Normal component parallel to the wall	148
4.1.8.3	Normal component perpendicular to the wall	148
4.1.8.4	Curvature	149
4.1.8.5	Force components	150
4.2	Applications	152

4.2.1	3D contact point study with density ratio $\rho_l/\rho_g = 1.0$ and viscosity ratio $\mu_l/\mu_g = 1.0$, based on the 2D work of Lai, Tseng and Huang (2010)	152
4.2.1.1	Hydrophilic case: $\theta = 60^\circ$, base mesh [12 x 12 x 4] and 2 to 6 levels of refinement. No-Slip boundary condition.	154
4.2.1.2	Hydrophilic case: $\theta = 60^\circ$, base mesh [12 x 12 x 4] and 6 levels of refinement. Comparison between Free-Slip and No-Slip boundary conditions	157
4.2.1.3	Hydrophilic case: $\theta = 45^\circ$, base mesh [12 x 12 x 4] and 5 levels of refinement. Comparison between Free-Slip and No-Slip boundary conditions.	163
4.2.1.4	Hydrophobic case: $\theta = 148^\circ$, base mesh [12 x 12 x 4] and 5 levels of refinement. Comparison between Free-Slip and No-Slip boundary conditions.	169
4.2.2	3D contact point study with density ratio $\rho_l/\rho_g = 800$ and viscosity ratio $\mu_l/\mu_g = 100$, based on the 2D work of Afkhami and Bussmann (2009).	176
4.2.2.1	Hydrophilic case: $\theta = 30^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.	176
4.2.2.2	Hydrophilic case: $\theta = 45^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.	181
4.2.2.3	Hydrophilic case: $\theta = 60^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.	185
4.2.2.4	Hydrophobic case: $\theta = 120^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.	189
4.2.2.5	Hydrophobic case: $\theta = 135^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.	193
4.2.2.6	Hydrophobic case: $\theta = 150^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.	197
4.2.3	3D contact point study with density ratio $\rho_l/\rho_g = 1.0$ and viscosity ratio $\mu_l/\mu_g = 1.0$. Couple with the Immersed boundary method . . .	201
4.2.3.1	$\theta = 30^\circ$	202
4.2.3.2	$\theta = 60^\circ$	206
4.2.3.3	$\theta = 135^\circ$	210

5 CONCLUSIONS **215**
 5.1 Future developments 216
BIBLIOGRAPHY **217**

CHAPTER I

INTRODUCTION

1.1 Objectives

In this thesis the author aims to analyze the dynamics of gas-liquid-solid contacts occurring in two-phase flows. The project focus on developing and validating of computational techniques to model accurately and efficiently the gas-liquid-solid contact dynamics.

The mathematical framework is developed in the 3D computer code previously and jointly developed at FEMEC-UFU and at IME-USP (AMR3D). This code employs a formulation based on the primitive variables modeling a flow (velocity, pressure, and phase indicator function) whose spatial discretization is constructed on block-structured Cartesian meshes including adaptive refinement. The developed modeling framework is aimed at efficiently capturing the physics and should be applicable in any numerical framework.

The main objective of the project is to develop a model capturing the gas-liquid-solids contact dynamics in multiphase flows. This model should be: (i) accurate, (ii) robust, (iii) as generic as possible . It is also aimed to validate, where possible, the predictions of the model with results obtained from experiments or from the literature.

The project is divided in the following steps:

- A) Check the “state of the art” through a literature review, focusing on recent publications involving Front-Tracking (FT), Volume of Fluid(VOF), Front-Capturing (FC), and other potentially new developments and applications.
- B) Inventorize existing frameworks of mathematical models approximating the normals, curvature and surface tension force with application to the gas-liquid interface and triple-wall interactions in fluid-fluid.
- C) To implement and validate a model describing the dynamics of the contact point at the gas-liquid-solid interface, with the following steps:

- [1] Implement interface reconstruction model

[2] Implement models to determine the normals and the curvature of the interface

[3] Implement a model for calculating the interfacial surface tension

[4] Implement a model for calculating the surface force associated with the gas-liquid-solid contact point.

[5] Couple the gas-liquid-solid contact point model with the Immersed Boundary (IB) method.

D) To validate the above model with:

- numerical experiments
- data and test-cases from the literature

E) To study cases which are relevant to the context of contact point problems, including the coupling with the Immersed Boundary method and compare, where possible, with experimental results and other available literature.

1.2 Justification

Annular two-phase flow problems are very common in industry, for instance in the area of extraction and transport of oil mixtures. Figures 1.1 and 1.2 illustrates the most common and known flow regimes of gas-liquid two-phase flows, which are classified either as separated or dispersed flows. The flow regime obtained in a two-phase flow generally depends on the properties, the velocity of the fluids, the volume fraction of the phases, and the inclination and the diameter of the pipe (LOILIER, 2006).

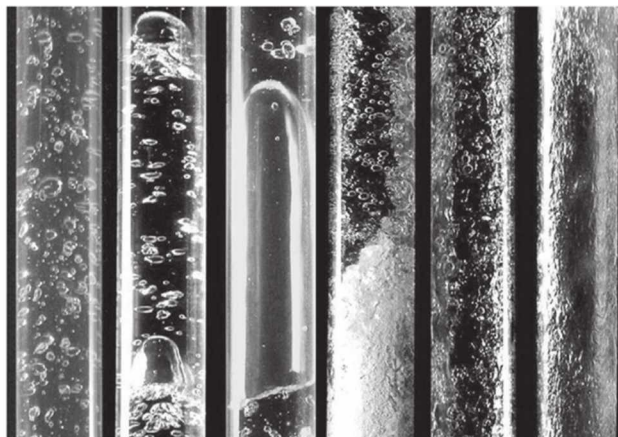


Figure 1.1: Test section photographs of upward air-water flow exhibiting the visual flow pattern features (from left to right): bubbly; spherical cap; stable slug; unstable slug; semi annular; annular. Source: Rosa, Flora and Souza (2012)

Of particular industrial interest are the stratified annular flow regimes as illustrated in Figs. 1.1 and 1.2. This regime is commonly found, for example, in the process of injecting

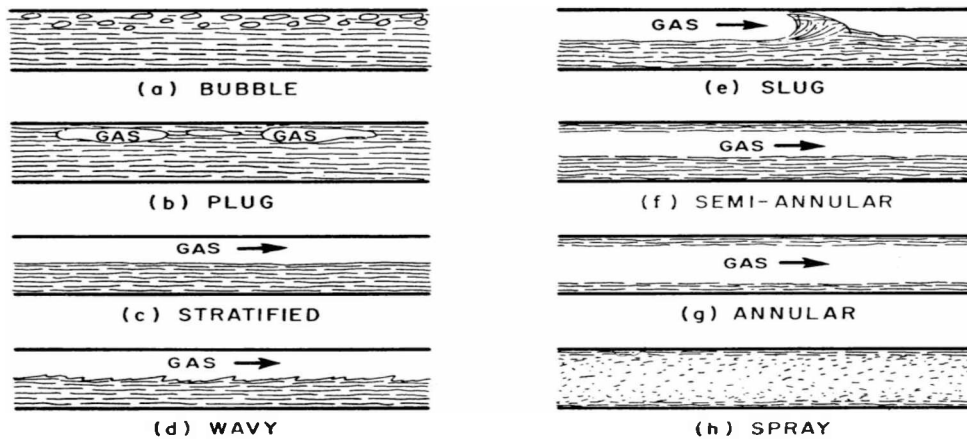


Figure 1.2: Flow patterns in a horizontal pipe. Source: Loilier (2006)

water into the transport of heavy oils, known as Core-Flow process. If the fluid near the wall has a low viscosity, the pressure drop of the total flow is less than if there were only a core of fluid. Literature data shows that the reduction in pressure drop can be orders of magnitude. When using this mechanism for reducing the viscous friction the pumping power associated with moving the fluid can be drastically reduced (CHARLES; GOVIER; HODGSON, 1961; RUSSELL; CHARLES, 1959)

Until a specific flow velocity, the flow remains stable. If the velocity is increased sufficiently, the difference in velocity between the two phases increases and the flow mixture becomes unstable, as shown in Fig. 1.1. When the instability grows sufficiently, the annular flow regime becomes highly unstable and the pressure drop increases dramatically.

Instabilities may occur via two mechanisms. The first is the Rayleigh-Taylor instability, arising due to the combined effect of interfacial tension and/or density gradients in both fluids. The second mechanism is the Kelvin-Helmholtz instability, arising due to the action of another surrounding liquid (OOMS, 1972).

Thus, the detailed study of the formation and transport of these instabilities is of fundamental importance for understanding these phenomena, to reduce the loss of pressure and thereby the power required for pumping. Due to the great difficulty of experimentally studying the details of these phenomena, the approach of mathematical modelling and numerical simulation has great potential for the detailed analysis of these physical phenomena.

1.3 International cooperation

The project was carried out at UFU. However, due to the experience and expertise in the area, Dr. Berend van Wachem from Imperial College London is involved in this project. Dr Berend van Wachem acted as co-supervisor, remaining in close contact with Rodrigo Lisita Ribera and Prof. Aristeu, advising and steering the project and providing supervision and knowledge where applicable.

During his PhD study the student spent 6 months at Imperial College London, under

Berend van Wachem supervision and CAPES scholarship. In this period, discussions about normal and curvature computation methods, an analysis of some of the methods presented on literature and their comparison with a proposed Least Squares method was performed, as well as discussions about the implementation of the contact point model.

1.4 Thesis organization

The work developed is presented in the following chapters:

- Chapter 2 presents a review of the “state of the art” about the two-phase flow and the VOF method. The surface tension, contact angle and moving contact point are also presented.
- Chapter 3 presents the mathematical modeling, the surface tension force model, a study of several methods to compute the normal and curvature, the VOF PLIC Interface reconstruction method and the surface force associated with the gas-liquid-solid contact point.
- Chapter 4 presents the verification of the models implemented and applications to contact point simulations. Results for the interface reconstruction and advection, normal and curvature computation, pressure jump, parasitic current analysis and simulations of a rising bubble in different flow regimes, with analysis of its terminal Re and shape are performed, as well as the study of contact point problems and its couple with the Immersed Boundary method.
- Chapter 5 presents the conclusion and future developments.

CHAPTER II

Background

2.1 Review of two-phase flows and the VOF method

The presence of liquid interphases plays a fundamental role in a large variety of phenomena, both in nature, such as cellular phagocytosis and insect flotation on pool water and in industrial processes, such as soldering and steam condensation (NAVASCUES, 1979). Processes such as extraction, chemical reaction, mass-transfer, separation, etc., also involve interfacial flows (GOPALA; WACHEM, 2008): gas-liquid bubble columns, frequently employed in the chemical, petrochemical, and wastewater industries (WACHEM; SCHOUTEN, 2002) and droplet clouds for combustion problems with liquid and gas reagents (??) are some examples.

The numerical simulation of interfacial problems dates back to the arrival of large computers on the research groups in the 1950s (HARLOW, 2004). Pasta and Ulam (apud HARLOW, 1963) showed that many features of the dynamics of fluids could be represented by a calculation of the trajectories of interacting particles, each representing a macroscopic fluid element.

This leads to a series of particle-based methods, one of the first ones being the Particle-in-Cell (PIC) method. Harlow (1988) describes the PIC method as an hybrid one, combining an Eulerian mesh of computational cells with a Lagrangian mesh of marker particles. He also presents a review about the method and a good list of references about its development and applications.

The method itself was presented by Harlow (1956) in one-dimensional form, with the aim of treating boundaries between materials, especially with large slips or distortions. The fluid dynamics is represented by a system of discrete particles with equal mass for each material. The particles in each cell have the same velocity \mathbf{u} , even if they are of different materials. He presented result for a steady-state shock for one material, shock reflection and transmission for particles of two different masses, and a fractured diaphragm problem in one

and in two dimensions.

One year later Harlow and Evans (1957) coined the name Particle-in-Cell (PIC). They also presented an explanation of the solution procedure, the use of several materials, the extension of the method for two-dimensional problems in cartesian and cylindrical coordinates and results for one and two-dimensional simulations, with explanations of two codes aimed to solve two-dimensional problems, the SUNBEAM code and the KAREN code.

Harlow et al. (1959) considers that, “as applied to two-dimensional calculations, the method was incomplete and should be modified somewhat”. They present a more detailed explanation of the method in two-dimensions and results for: 1) Shock-Wave refraction at a gaseous interface, 2) Shock passage through a discontinuously enlarged channel, 3) Interaction of a shock with a deformable object, 4) Hypersonic shear flow with perturbed interface, 5) Taylor instability and 6) Viscous-Flow calculations.

According to Brackbill (2002), PIC was “the first method to model high-speed and free-surface flow in two dimensions, and the first to model collisionless plasmas in two space dimensions”. In its latest versions the method is called Material Point Method (MPM) (ZHANG et al., 2008).

Departed from PIC and based on the work of Pasta and Ulam (1959), Harlow and Meixner (1961), Harlow (1963), Daly et al. (1964) presents the Particle and Force (PAF) method, a particle method with no Eulerian mesh. “It is based on a representation of the fluid by a set of mass points which are accelerated by mutual forces and whose consequent motions represent that of a fluid”. The purpose is to solve multidimensional compressible-fluid dynamics problems (HARLOW; MEIXNER, 1961).

The “technique was moderately successful in applications to some two-dimensional test problems, but suffered from fluctuations that especially manifested themselves in ragged behavior of material interfaces” (HARLOW, 1988). Harlow (1963) considered the PAF much more likely the original procedure of Pasta and Ulam (1959), while the PIC resembles the Eulerian method more closely than any other particle method.

The Arbitrary-Lagrangian-Eulerian (ALE) method (HIRT; AMSDEN; COOK, 1974) also departs from the PIC method. Accordingly to Harlow (2004), one way of viewing the Particle in Cell method is to consider the mesh of cells as Lagrangian, so that the first part of the calculation advances all the variables, and the second part maps the mesh back into its original configuration. In the ALE method, only a partial mapping is performed in the mesh in each cycle.

Zhang, Vanderheyden and Zou (2007) couples the ALE method with the Material Point method (MPM), with the aim of simulating, among other things, multiphase flows, interfacial flows and free surface flows. A numerical method for the computation of compressible multi-material flow problems, with the aim of modeling underwater explosions, is developed by Luo, Baum and Löhner (2004).

Following the particle-based concept, Harlow, Shannon and Welch (1965) develop a method to study waves and other phenomena associated with the motion of an incompressible

fluid with free surface. They use a finite-difference approximation to the full, nonlinear, Navier-Stokes equation for a viscous, incompressible fluid, related to an Eulerian mesh. Harlow and Welch (1965) designated it as the Marker and Cell (MAC) method (Fig. 2.1).

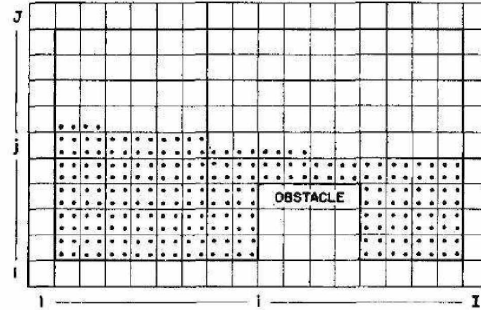


Figure 2.1: The MAC method. Particles are used to identify full cells and interface cells. Source: Harlow and Welch (1965)

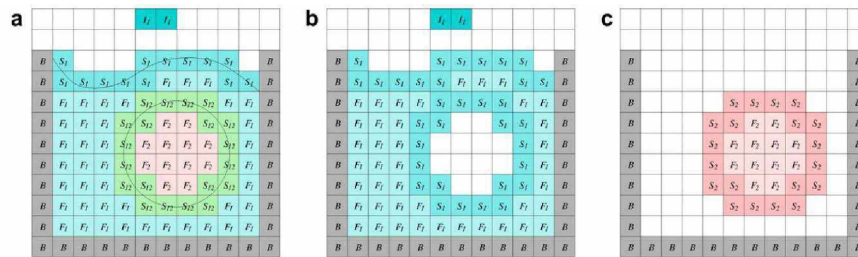


Figure 2.2: Multi-fluid cell flagging example: (a) classification for fluids 1 and 2; (b) classification only for fluid 1; (c) classification only for fluid 2. Source: McKee et al. (2008)

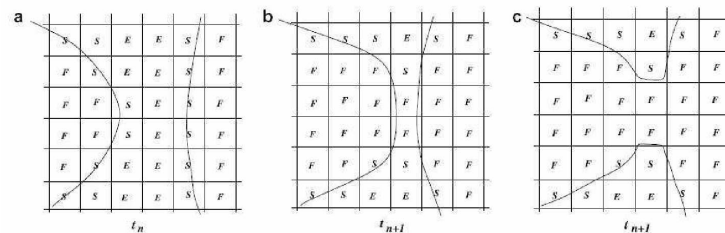


Figure 2.3: Steps required to perform surface merging. Source: McKee et al. (2008)

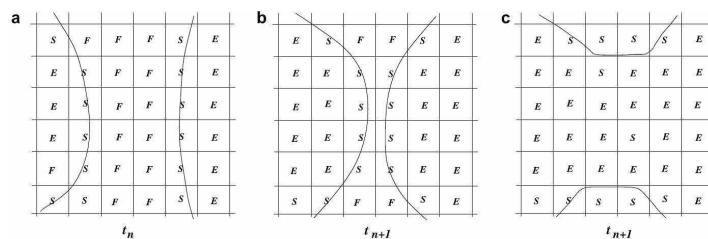


Figure 2.4: Steps required to perform surface splitting. Source: McKee et al. (2008)

On the MAC method, the only purpose of the particles is to indicate fluid configuration, showing which cells contain fluid and which lie along the free surface. Unlike the previous

methods, the particles do not participate on the computation of the flow. “A cell with no marker particles is considered to contain no fluid. A cell with marker particles, lying adjacent to an empty cell, is called a surface cell. All other cells with particles are considered to be filled with fluid” (HARLOW; WELCH, 1965). Fig. 2.2 illustrates this marking process, Fig. 2.3 illustrates the process of Merging and 2.4 the process of Splitting.

A complete description of the MAC method, from its developers, can be found at Welch et al. (1966). Free surface motions is studied by Harlow and Welch (1965), and the splash of liquid drops by Harlow and Shannon (1967). Later, Amsden and Harlow (1970) develop a simplified MAC method (SMAC) and Hirt, Nichols and Romero (1975) develops a simplified MAC code, SOLA, with no marker particles. The SOLA family of codes were the first ideas distributed internationally (MCKEE et al., 2008).

A review of recent developments of the method, and numerical simulations of a three-dimensional cylindrical and planar jet buckling, the splash of a liquid drop, circular hydraulic jump, filling process of a trapezoidal container, Newtonian and Viscoelastic jet impinging onto a rigid plate and the extrudate swell are presented by McKee et al. (2008).

The MAC method is also used by Daly (1969) but at each cycle of calculation the interface particles are connected to form an interface line, through the application of a spline fit interpolation scheme. This is done in order to obtain the orientation of the fluid interface, which allows to determine the surface tension contribution to the fluid acceleration. Accordingly to Prosperetti and Tryggvason (2007), this was the first use of connected marker points to identify a boundary between two fluids governed by the full Navier-Stokes equation, with further progress made by Peskin (1977), but the first use of (connected) marker points to capture immiscible fluid interfaces in a flow governed by the full Navier-Stokes equations is due to Unverdi and Tryggvason (1992), who compute the flow field in a stationary grid and represents the interface, which is explicitly tracked, by a separate, unstructured grid.

This approach is referred as Front Tracking (FT) method. Advances on it and applications to homogeneous bubbly flows, atomization, flows with variable surface tension, solidification and boiling are presented by Tryggvason et al. (2001).

Instead of a direct representation of the free boundaries, the Volume of Fluid method (VOF) (HIRT; NICHOLS, 1981) (NICHOLS; HIRT; HOTCHKISS, 1980), is an Eulerian method that follows regions of fluid. They are among the most used ones for the simulation of interfacial multiphase flows. Its basic concept is to perform a mass balance of a scalar F within a control volume, which represents the volume fraction of one phase inside a computational cell (Fig. 2.5 (a) and (b)). For a cell completely full of one phase $F = 1$, for a cell completely empty, $F = 0$, and for a cell with an interface, $0 < F < 1$. This semi-discontinuous aspect of the color function facilitates the calculation of the properties of each of the phases and makes it possible to present an accurate numerical scheme for solving the color transport equation (WACHEM; SCHOUTEN, 2002).

VOF methods possess superior mass conservation properties, but the discrete, abruptly-varying volume fractions representation of the interface poses challenges for the accurate

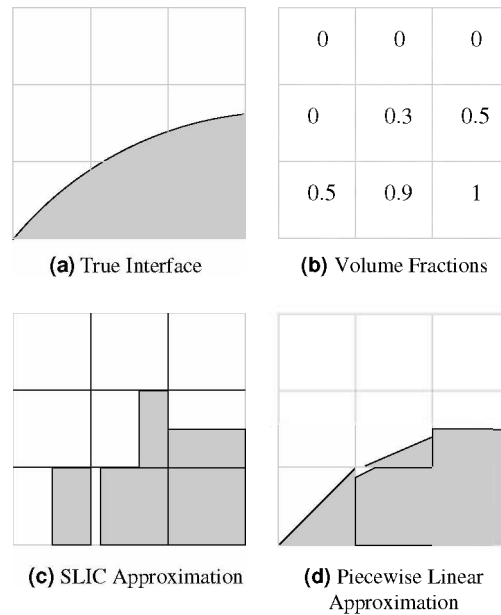


Figure 2.5: VOF representation of an interface. The information from a known geometry (a) should be transformed in a scalar in each eulerian computational cell (b), which is later reconstructed by a numerical scheme (c-d). Source: Pilliod and Puckett (2004)

estimation of interfacial curvature (CUMMINS; FRANCOIS; KOTHE, 2005). This poor curvature estimation may produce unphysical velocities (spurious or parasitic currents) at fluid interface, as well as erroneous pressure drop over the interface, affecting the accuracy of the simulations (LÓPEZ et al., 2009; RAESSI; MOSTAGHIMI; BUSSMANN, 2007).

Another important aspect is the approximation used for the interface normal \mathbf{n} . The solution quality of the VOF and the Continuum Surface Force (CSF) methods are sensitive to any error in it. Its accurate estimation often dictates overall accuracy and performance in these methods (KOTHE et al., 1996).

One of the main alternatives for the volume-of-fluid method for the direct advection of a marker function is the Level-Set (LS) method (PROSPERETTI; TRYGGVASON, 2007). Its development for computing the motion of incompressible two-phase flows is due to Sussman, Smereka and Osher (1994), who points some main advantages of the method: the elimination of the problem of adding/subtracting points to a moving grid, the automatically merging and breaking of the interface and its easy of extension to three-dimensional problems.

The interface is described by the LS function (defined as a signed distance function). Its continuous and smooth aspect make the calculation of normal and curvature very easy (WANG et al., 2009). Some of its drawbacks is the lack of mass conservation, its considerable degrade when interfaces possess high curvature relative to the mesh spacing and the expensive reinitialization scheme needed for maintaining solution quality (KOTHE; RIDER, 1995).

An attempt to explore the advantages of both the VOF method (mass conservation) and the Level Set (easy estimation of geometric properties) is the Coupled Level Set and Volume of Fluid method (CLSVOF) (SUSSMAN; PUCKETT, 2000), where the interface is

reconstructed and advected with the VOF method, and the level set function is re-distanced based on the reconstructed interface.

Despite its advantages, it introduces two different representations of the same interface with the associated complexity, efficiency and consistency issues (POPINET, 2009). Simulations of gas bubble rising in a viscous liquid, water drop impact onto a deep water pool, wave breaking of a steep Stokes wave and plunging wave breaking over a submerged bump are presented by (WANG et al., 2009). Droplet impact on a dry surface with a dynamic contact angle model is study by (YOKOI et al., 2009).

Due to the property of excellent mass conservation, the VOF method is chosen in this work for the numerical two-phase flow simulations. Also, several studies about the contact between the two fluid phases and a solid (the triple line/point) have been carried out with the VOF methods. Some examples are Afkhami and Bussmann (2004), who studies surface tension driven flows, Nichita, Zun and Thome (2010), who study the dynamic wetting and Fang et al. (2008) who performs a 3D numerical simulation of contact angle hysteresis for micro-scale two-phase flows.

2.2 Surface tension

“The water, while in surface, acts in a different way to what it does inside”. It behaves as if the surface were an elastic skin. If the weight of the drop or the force pulling it could be prevented from acting, the drop would only feel the effect of this elastic skin, which would try to pull it into such a form as to make the surface as small as possible. It would rapidly become a perfectly round ball (BOYS, 1920).

By attempting to determine theoretically the form of the surface of a drop of water resting on a horizontal plane, Johann Andreas von Segner introduced the concept of the surface tension of liquids in 1751. On the hypothesis of the attraction of the parts of a fluid for each other, he admitted the *tenacity* of fluids, due to the action of an attractive forces resident in their constituent molecules, whose sphere of activity of each particle is of insensible magnitude. The shape which the drop assumes refers to the action of the superficial particles and their *tenacity*, which counteract the tendency of the drop to spread due to gravity (VARIOUS, 1911; MAXWELL, 1890).

Ideally, the shape of a liquid droplet is determined by the surface tension of the liquid (Fig. 2.6). In a pure liquid, each molecule in the bulk is pulled equally in every direction by neighboring liquid molecules, resulting in a net force of zero. The molecules exposed at the surface do not have neighboring molecules in all directions to provide a balanced net force. Instead, they are pulled inward by the neighboring molecules, creating an internal pressure (Fig. 2.7) (YUAN; LEE, 2013).

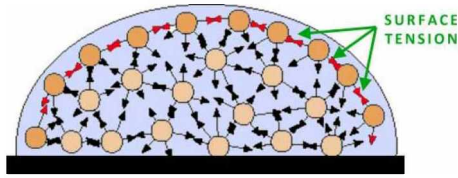


Figure 2.6: Surface tension is an intermolecular force which contract the surface, being responsible for the shape of liquid droplets. Source: www.butane.chem.uiuc.edu

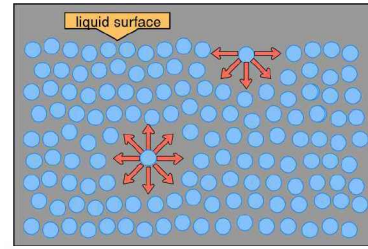


Figure 2.7: The molecules exposed at the surface are pulled inward, as they do not have neighboring molecules in all directions to provide a balanced net force. Source: www.fusedglass.org

The higher pressure is in the fluid medium on the concave side of the interface, since surface tension results in a net normal force directed towards the center of curvature of the interface (BRACKBILL; KOTHE; ZEMACH, 1992).

“Whenever there is a curved or angular surface, it may be found by collecting the actions of the different particles, that the cohesion must necessarily prevail over the repulsion, and must urge the superficial parts inward with a force proportionate to the curvature, and thus produce the effect of a uniform tension of the surface” (YOUNG, 1805).

The relation between the total force acting on the interface and the curvature is given by the well known Laplace equation (NAVASCUES, 1979):

$$p_2 - p_1 = \Delta p = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \quad (2.1)$$

where: σ is the surface tension coefficient and R_1 and R_2 are the principal radii of curvature.

Since the principal curvatures of the surface are $\kappa_1 = 1/R_1$ and $\kappa_2 = 1/R_2$, and the mean curvature H of the surface is defined as:

$$H = \frac{\kappa_1 + \kappa_2}{2} \rightarrow 2H = \kappa_1 + \kappa_2 \quad (2.2)$$

The Laplace equation can be written in terms of the mean curvature of the surface:

$$p_2 - p_1 = \Delta p = 2\sigma H \quad (2.3)$$

An alternate definition often used in fluid mechanics to avoid factors of two is simply

express Eq.(2.1) and (2.3) in terms of a curvature κ :

$$p_2 - p_1 = \Delta p = \sigma \kappa \quad (2.4)$$

From the macroscopic point of view, surface tension can be defined in two different contexts: in a mechanical way where it is related to a tension parallel to the surface, and in a thermodynamic way in which σ is associated with the work necessary to create a new surface (NAVASCUES, 1979).

2.3 Wetting, contact angle and contact point

The interface where solid, liquid, and vapor co-exist is referred to as the *three-phase contact line* (YUAN; LEE, 2013). It is encountered frequently in everyday experience and is of concern to many areas of science (DURBIN, 1988). Oil recovery, lubrication, liquid coating, printing, and spray quenching are areas where wetting plays an important role; superhydrophobic surfaces have potential applications in self-cleaning, nanofluidics, and electrowetting (YUAN; LEE, 2013).

The earliest direct recognition of wetting phenomena was probably given by Galileo, in 1612, who noticed that the top of a denser solid floating on a liquid was below the surface of the liquid, in what is known today as *capillary depression*. However, the treatment of the floating-dense-solid phenomenon, which explicitly includes contact angles, was only given in 1920, by Sulman (apud GOOD, 1992).

As shown in Fig. 2.8, three different wetting states are possible: partially wet, completely wet or completely dry (BONN et al., 2009).

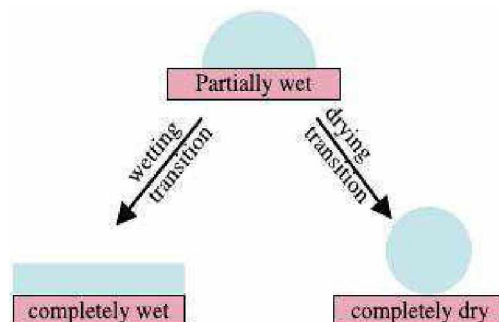


Figure 2.8: Three different possible wetting states. Source: Bonn et al. (2009)

Wetting phenomena are an area where chemistry, physics, and engineering intersect. On the short-ranged interactions, surface chemistry, or chemical interactions, act over the scale of molecules, and is of key importance in determining wetting behavior. In addition, surface forces such as van der Waals or electrostatic forces are paramount for determining

whether or not a fluid will wet a given surface. These forces are called long ranged, as they can still be important over distances of few tens of molecules (BONN et al., 2009).

Usually, the primary data in wettability studies is the contact angle (θ) (YUAN; LEE, 2013). It is the angle measured from the liquid side of the contact line (DUSSAN, 1979), and indicates the degree of wetting when a solid and liquid interact. Small contact angles ($\theta < 90^\circ$) correspond to high wettability, while large contact angles ($\theta > 90^\circ$) correspond to low wettability (YUAN; LEE, 2013). As shown in Fig. 2.9, its measurement is easily performed by establishing the tangent (angle) of a liquid drop with a solid surface at the base (KWOK; NEUMANN, 1999).

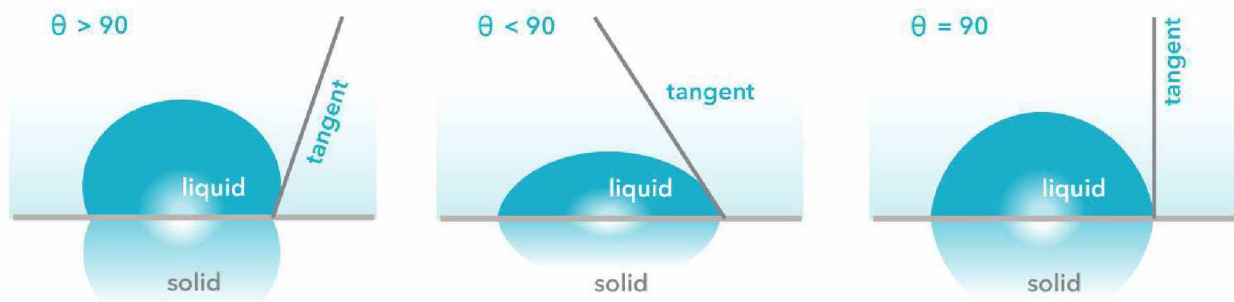


Figure 2.9: Illustration of contact angles formed by sessile liquid drops on a smooth homogeneous solid surface. Source: www.biolinscientific.com

The static contact angle (θ_S) is a direct consequence of the molecular interactions among the three materials at the contact line (DUSSAN, 1979). Its value is directly related to the relative strength of the cohesive and adhesive forces (Fig. 2.10). Cohesive forces are attractive forces between molecules of the same type. They cause the surface of a liquid to contract to the smallest possible surface area. This general effect is called surface tension (Macroscopic effect). Adhesive forces are attractive forces between molecules of different types.

The larger the strength of the cohesive force relative to the adhesive force, the larger θ is, and the more the liquid tends to form a droplet. The smaller θ is, the smaller the relative strength, so that the adhesive force is able to flatten the drop. When the cohesion and adhesion forces are in equilibrium, the matching contact angle is static (MOURIK, 2002). Theoretically, the contact angle is expected to be characteristic for a given solid-liquid system in a specific environment (YUAN; LEE, 2013).

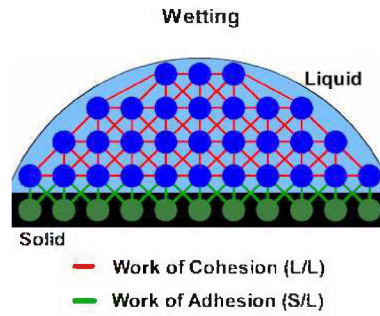


Figure 2.10: The degree to which wetting occurs (i.e., wettability) is determined by the cohesive forces of the liquid molecules among themselves and the adhesive forces that result from the molecular interactions between the liquid and the solid as illustrated in the figure. (In real life, the molecules are not so neatly organized). Source: www.ramehart.com

An expression of the static contact angle θ_S was first derived by Young (1805). Assuming that each of the three material boundaries possess a constant surface tension, he reasoned that for the static equilibrium of the system, the horizontal components of the surface tension must sum to zero at the contact line - Eq.(2.5) and Fig. 2.11.

$$\gamma_{LV} \cdot \cos(\theta_S) = \gamma_{SV} - \gamma_{SL} \quad (2.5)$$

where the subscripts LV stands for the liquid-vapor interface, VS for the vapor-solid interface and LS for the liquid-solid interface.

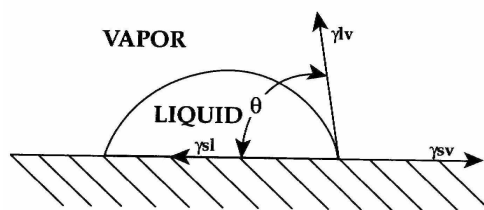


Figure 2.11: The equilibrium between the solid, liquid and vapor phase interfaces is expressed by the Young equation. Source: Burris (1999)

It has been implicitly supposed that Young's equation is applicable to a flat smooth solid surface, but in real life most solid surfaces show some degree of roughness or heterogeneity (NAVASCUES, 1979; QUÉRÉ, 2002).

One well-known example of departure from ideality is when the solid is rough (GOOD, 1992), as illustrated in Fig. 2.12a. Young's equation should then be modified by a roughness factor r (the actual contact area divided by the projected area of surface that the droplet contacts) (GAO; MCCARTHY, 2007). The relation between the apparent contact angle θ^* and that θ corresponding to a smooth surface was presented by Wenzel (1936):

$$r = \frac{\cos\theta^*}{\cos\theta} \quad (2.6)$$

Since no surface is completely smooth at the molecular level (every surface have some sort of roughness), it can be assumed that $r > 1$. Thus, a roughened surface magnifies the wetting properties of the solid: a wetting surfaces will become more hydrophilic, and a non-wetting surfaces will be more hydrophobic (WENZEL, 1936; CASSIE, 1948). This agreement is only qualitative, as a simple linear relation is not observed (QUÉRÉ, 2002).

Also, in Eq.(2.6), the roughness factor can be made arbitrarily large, which seems to imply that complete wetting ($\cos\theta^* > 1$) or complete drying ($\cos\theta^* < -1$) should be induced by large roughness ($r \gg 1$), which is not observed (QUÉRÉ, 2008). The effect of surface roughness can be described by the so-called Wenzel equation if the roughness is significantly below the wavelength of light (??). In other words, “drops should be much larger than the defects to use such an averaged model” (QUÉRÉ, 2008).

For a porous solid surface, Cassie and Baxter (1944) presents the following expression for the apparent contact angle θ^* :

$$\cos\theta^* = A_1\cos\theta - A_2 \quad (2.7)$$

where θ corresponds to the angle in a smooth surface, A_1 is the total area of solid-liquid interface and A_2 is the total area of liquid-air interface.

Since the total area A bellow the droplet is $A = A_1 + A_2 = 1$, Eq.(2.7) can be rewritten as:

$$\cos\theta^* = \phi_s(\cos\theta + 1) - 1 \quad (2.8)$$

where $\phi_s = A_1/A$ is the fraction of the solid/liquid interface below the drop.

Equation(2.7/2.8) is based on the assumption that the water droplet sits in top of air bubbles, as illustrated in Fig. 2.12b. “If there is only air, it predicts a ‘contact’ angle of 180° (*i.e.*, no contact). Any deviation from this value tells us the proportion of solid actually contacting the liquid” (QUÉRÉ, 2008).

“In order for the Cassie-Baxter state to be observed, there should be an energy barrier which slows the transition from this metastable state” (MURAKAMI; JINNAI; TAKAHARA, 2000). “When the surface is rough but not porous, A_2 is zero, and Eq.(2.7) reduces to Wenzel’s equation for the apparent contact angle of a rough surface with roughness factor A_1 ” (CASSIE; BAXTER, 1944).

For heterogeneous surfaces, Cassie (1948) presents an equation for the apparent contact angle θ^* , which reduces to Eq.(2.7) when applied to a porous surface:

$$\cos\theta^* = A_1\cos\theta_1 + A_2\cos\theta_2 \quad (2.9)$$

where θ_1 and θ_2 are the Young contact angles for the smooth flat solid surfaces 1 and 2, and A_1 and A_2 are the ratios in the actual solid surface of the surfaces 1 and 2 to the total area.

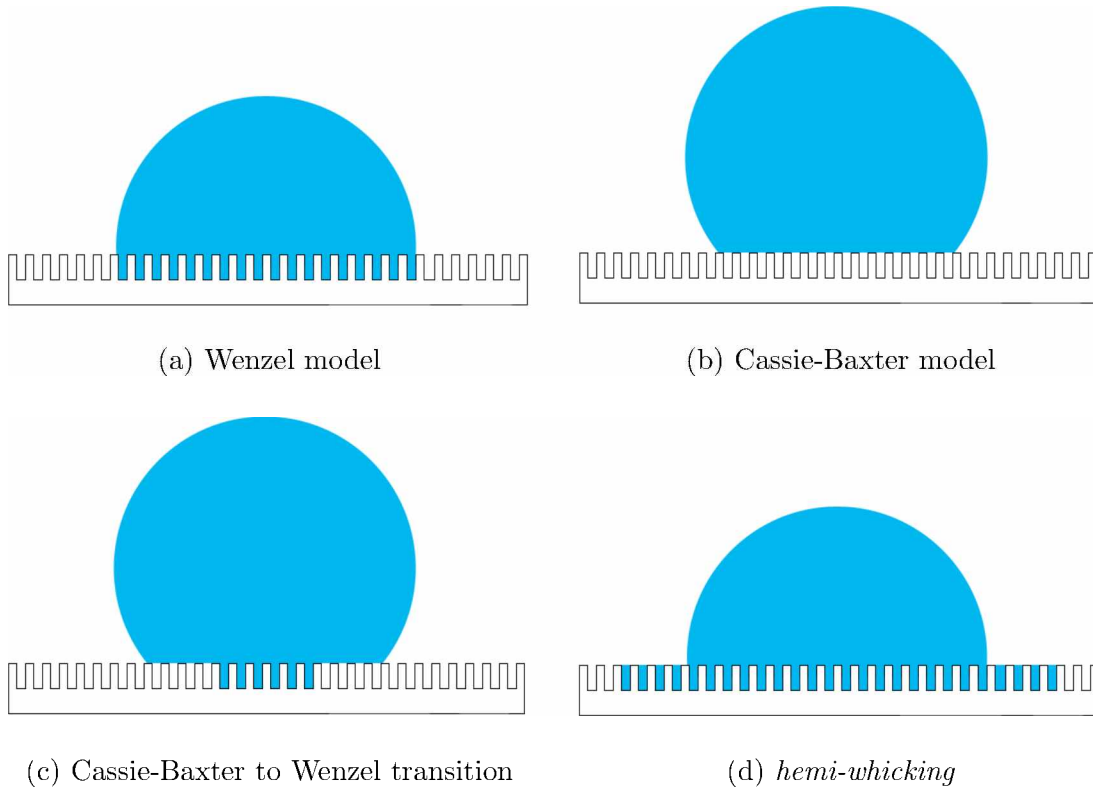


Figure 2.12: Wetting states defined by: a) Wenzel model; b) Cassie-Baxter model; c) transition from Cassie-Baxter to Wenzel; and d) *hemi-whicking*. Source: www.commonswiki.org/wiki/User:Spinal83

The two states, *i.e.* Wenzel and Cassie-Baxter states, illustrated in Fig. 2.12a and Fig. 2.12b, respectively, exhibit clear differences in drop mobility. Drops are extremely mobile when they are supported by composite solid-liquid-air interfaces (Cassie-Baxter state) and immobile when they fully wet the textured surfaces (Wenzel state). “For a variety of applications (e.g. fluid motion control in microfluidics), it is therefore of great interest that reversible transitions between these states can be induced at will” (BENTLEY, 2009).

Superhydrophobic properties are desired for applications such as self-cleaning or drag reduction, while for applications that require good wetting, mixing, or transport in between the structures, such as wafer cleaning in micro- and nanoelectronics fabrication processes, superhydrophobicity should be avoided (XU et al., 2014). The active control of the wetting properties may be induced in a number of ways, for example, by electrowetting, light irradiation, or a change in temperature or pH (BENTLEY, 2009).

The transition from the metastable Cassie-Baxter state to the stable Wenzel state, where the droplet partially penetrates between the pillars (Fig. 2.12c), has been studied extensively in recent years (LEE et al., 2010; MURAKAMI; JINNAI; TAKAHARA, 2000; GIACOMELLO et al., 2013). A reversible transition from Wenzel to Cassie-Baxter states, however, is normally very complicated to achieve, due to the existence of (Gibbs) energy barriers between the states (BENTLEY, 2009).

The dynamics of this transition (Cassie-Baxter to Wenzel) is very quick and follows a zipping mechanism: one row of cavities gets filled before jumping to the next row (QUÉRÉ, 2008). Air pockets should be metastable for Young contact angles between $\Pi/2$ and the threshold value given by Eq.(2.10), which relates the Wenzel and the Cassie-Baxter contact angles (BICO; THIELE; QUÉRÉ, 2002):

$$\cos\theta < \frac{\phi_s - 1}{r - \phi_s} \quad (2.10)$$

where θ is the Young contact angle, $\phi_s = A_1/A$ is the fraction of the solid/liquid interface below the drop and r is the surface roughness.

In the hydrophilic case, a second phenomenon can occur, which is intermediate between spreading and imbibition. It is called *hemi-wicking* and is illustrated in Fig. 2.12d. For a partial wetting, liquid film invades the texture of the solid, but the top of the spikes remains dry as the imbibition front advances. The Young contact angle θ should be smaller than a critical contact angle ($\theta < \theta_c$), intermediate between 0 and $\Pi/2$, for this condition to occur (BICO; THIELE; QUÉRÉ, 2002):

$$\cos\theta_c = \frac{1 - \phi_s}{r - \phi_s} \quad (2.11)$$

The Wenzel model is valid between θ_c and $\Pi/2$. If the contact angle is less than θ_c , the penetration front spreads beyond the drop and a liquid film forms over the surface. The film smoothes the surface roughness and the Wenzel model no longer applies (ISHINO; OKUMURA, 2008).

Both the Wenzel and Cassie equations reinforce the concept that the area of contact between the liquid and solid should affect the contact angle. Although the theories behind them are consistent with observed data for many surfaces, many examples where they are inconsistent have also been reported. Extrand (2003) showed that the three-phase structure at the contact line, not the liquid-solid interface beneath the droplet, controls the contact angle. Contact angle and hysteresis are a function of contact line structure and the kinetics of droplet movement, rather than thermodynamics, dictate wettability (GAO; MCCARTHY, 2007).

If the velocity of the contact line is positive ($\mathbf{U}_{CL} > 0$), it is considered that the

contact line is advancing. The extrapolated value of the (static) contact angle θ_S in the limit as $\mathbf{U}_{CL} \rightarrow 0$ is called the static advancing contact angle, $\theta_{A,S}$. In the same manner, the static receding contact angle, $\theta_{R,S}$, is the extrapolated value in the limit as $\mathbf{U}_{CL} \rightarrow 0$, with $\mathbf{U}_{CL} < 0$.

For many materials, there exists an interval $[\theta_{R,S} \leq \theta_S \leq \theta_{A,S}]$, with the property that if θ lies within this interval, then the contact line does not appear to move. If a droplet on a surface is allowed to evaporate in a low humidity environment or if water is carefully withdrawn from the droplet with a syringe (Fig. 2.13 - left), the droplet decreases in volume and contact angle, maintaining the same contact area with the surface until it begins to recede, with a constant contact angle θ_R , characteristic of the surface chemistry and topography. If the surface is cooled to below the dew point and water condenses on the droplet or if water is carefully added to the droplet with a syringe (Fig. 2.13 - right), the droplet volume and contact angle increase, and again, the same contact area is maintained until the droplet begins to advance, at a constant advancing contact angle, θ_A , which is also characteristic of the surface chemistry and topography (GAO; MCCARTHY, 2006).

This non-uniqueness in the (static) contact angle is often referred to as *contact-angle hysteresis*. It is common to find hysteresis, on practical surfaces, in the range of 10° or larger and 50° or more of hysteresis can easily be observed in some cases (GOOD, 1992). An experimentally verified cause of this phenomenon is roughness of the surface of the solid. This difference can also depend on time interval between movement and measurement, on contamination, and on many aspects of the state of the solid surface, as heterogeneity and content of liquid dissolved or “penetrated” in it (HUH; SCRIVEN, 1971).

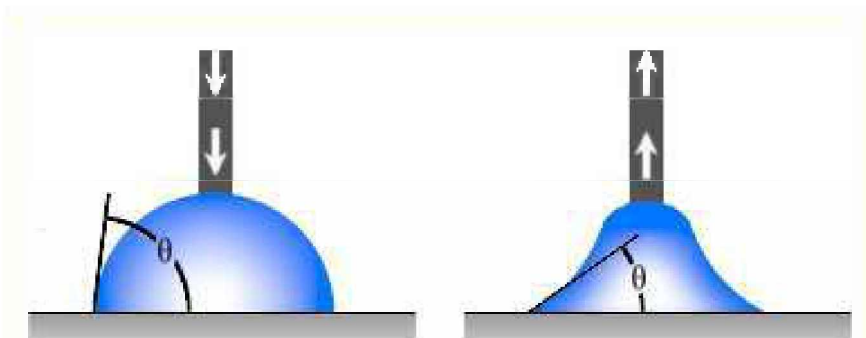


Figure 2.13: If water is carefully added to the droplet with a syringe, the droplet volume and contact angle increase, and the same contact area is maintained until the droplet begins to advance, at a constant advancing contact angle (left); if water is carefully withdrawn from the droplet with a syringe, the droplet decreases in volume and contact angle, maintaining the same contact area with the surface until it begins to recede, with a constant contact angle θ_R (right). Source: www.ramehart.com

2.4 Moving Contact Line (MCL)

In all processes in which a liquid is coated onto a solid, the liquid must dynamically wet the solid surface. However, the precise mechanism by which a liquid front advances across a solid surface remains only partially understood (BLAKE, 2006). Accordingly to Huh and Scriven (1971), studying resistance to motion of an isolated mercury index in an otherwise air-filled capillary tube, West (1911) appears to have been the first to pay attention to flow near a moving contact line.

Dynamic wetting operates on a scale that extends from macroscopic to the molecular level (BLAKE, 2006); understanding the multi-scale nature of the flow and how this gives rise to dynamical transitions is the main progress achieved in recent years in the fields of fluid mechanics, chemistry and engineering (SNOEIJER; ANDREOTTI, 2013). “Even for an infinitesimal velocity, the six decades separating the molecular size (nanometer scale) from the capillary length (millimeter scale) are the locus of a force absent from the static problem: viscosity.” (SNOEIJER; ANDREOTTI, 2013).

On the macroscopic scale of the capillary length l_λ , the shape of the meniscus is governed by the balance of gravity and surface tension; at smaller scales one encounters a visco-capillary regime, characterized by the capillary number Ca . The capillary number describes the visco-capillary balance, being a key parameter for moving contact lines (SNOEIJER; ANDREOTTI, 2013). Wetting speed, viscosity, surface tension, and contact angle are all macroscopic quantities. The relative velocity at which the liquid moves across the solid (the contact-line velocity) and the (dynamic) contact angle are the main parameters used to quantify the dynamics of wetting (BLAKE, 2006).

“In many flow situations, the apparent contact angle completely describes the dynamics. Examples are drop spreading (HOCKING, 1983), drops sliding down a window at low velocities (AMAR; CUMMINGS; POMEAU, 2003; RIO et al., 2005), or the relaxation of contact line perturbations (GOLESTANIAN; RAPHAËL, 2001; NIKOLAYEV; BEYSENS, 2003; SNOEIJER et al., 2007)”(SNOEIJER; ANDREOTTI, 2013).

Similar to the static contact angle, there exists an advancing and a receding dynamic contact angles, for the cases with $\mathbf{U}_{CL} > 0$ and $\mathbf{U}_{CL} < 0$, respectively, as illustrated in Fig. 2.14.

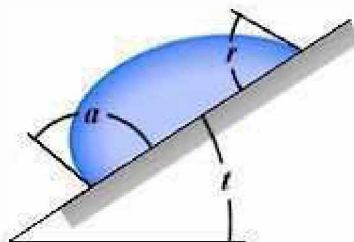


Figure 2.14: Advancing and receding contact angles captured by tilting base method. Source: www.ramehart.com

In forced wetting, the contact line is made to move by application of an external force. In such cases, a relationship is expected between θ_D and U_{CL} for a given system under a given set of conditions (BLAKE, 2006). Out of equilibrium, the liquid near the solid tends to move towards its equilibrium state, with the velocity of the contact line (U_{CL}) depending monotonically on the deviation of the time-dependent (dynamic) contact angle (θ_D) with the static contact angle (θ_S) (MOURIK, 2002). It is generally observed that advancing angles increase while receding angles decrease with increasing rates of steady contact-line displacement (BLAKE, 2006). Accordingly to Gutoff and Kendrick (1982), Deryagin and Levi (1964) were the first to note that with increasing velocities the dynamic contact angle increases to 180° .

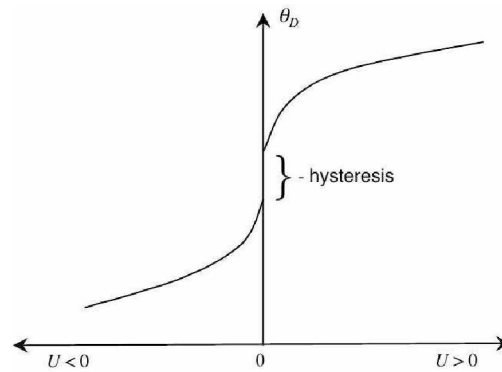


Figure 2.15: Schematic representation of the velocity dependence of the contact angle. source: Blake (2006)

“An interesting situation arises when a liquid is forced to flow over a surface that it does not spontaneously wet in thermodynamic equilibrium. In such partial wetting conditions, it is energetically favorable for the liquid to stick together as much as possible and to leave most of the surface dry. However, an external driving of the flow can push the system sufficiently far from equilibrium such that it undergoes a dynamical wetting transition. In practical terms, this means that the contact line motion cannot exceed a maximum speed: enforcing larger velocities leads e.g. to deposition of liquid films, break-up of liquid drops or entrainment air bubbles” (SNOELJER; ANDREOTTI, 2013).

The dynamic contact angle plays an important role especially when surface tension forces are dominant (low capillary number - Ca) and gravitational forces are not (low Bond number - Bo).

$$Ca = \frac{U\mu}{\sigma} = \frac{\text{viscous forces}}{\text{surface tension forces}} \quad (2.12)$$

$$Bo = \frac{\rho g R^2}{\sigma} = \frac{\text{gravitational forces}}{\text{surface tension forces}} \quad (2.13)$$

where: ρ , σ , U , μ and R denote the density, surface tension, viscosity, velocity and a characteristic length.

An important conclusion reached in nano-fluidics is that the Navier-Stokes equations remain valid for liquid layers down to nanometer scale for simple fluids like water, under normal conditions (BOCQUET; CHARLAIX, 2010). This means that in between ~ 10 nm from the contact line up to macroscopic scales, the shape of a moving interface can be described by continuum hydrodynamics (SNOEIJER; ANDREOTTI, 2013)

From the viewpoint of fluid mechanics there is a fundamental issue of how fluid displacement at a solid surface is to be understood. Whereas the ultimate resolution must rest on molecular consideration of non-equilibrium kinetics, the problem can be approached and delineated through kinematics and dynamics of fluids (BLAKE, 2006). About the kinetics of the dynamic contact angle, no satisfying theory has been found yet.

The classical hydrodynamic approach describing flow near a moving contact line does not result in a physically acceptable solution. Due the conflict between a moving contact line and the conventional no-slip boundary condition between a liquid and a solid, stresses are unbounded near the wetting line, and the force exerted by the liquid on the solid becomes infinite (BLAKE, 2006). The hydrodynamics is in essence described by a corner flow, which has no intrinsic length scale (HUH; SCRIVEN, 1971). One can, thus, only define a local Reynolds number based on the distance to the contact line r , which can becomes arbitrarily small. Thus, the Reynolds number is typically very small and inertia can often be neglected. Also, the viscous stress near the contact line scales as $\nu \mathbf{U}_{CL}/r$. Hence, the shear stress diverges upon approaching $r = 0$ (SNOEIJER; ANDREOTTI, 2013). The no-slip boundary condition of viscous flow gives rise to a non-integrable singularity in the surface shear stress. As Huh and Scriven (apud DURBIN, 1988) describe it, “not even Herakles could sink a solid if the physical model were correct, which it is not“. There is an obvious contradiction between the use of a no-slip boundary condition and the motion of the liquid front (WANG; PENG; DUAN, 2007).

The moving contact-line singularity arises when equations derived for continuous motion are applied to situation in which the continuum motion is discontinuous. This leads to the conclusion that a fundamental modification of these laws must be made (DURBIN, 1988).

On a nanoscale, when having a moving contact line, you will always get slip (DAVIS, 2014). Recently controlled experiments, with typical dimensions of microns or smaller, have demonstrated an apparent violation of the no-slip boundary condition for the flow of Newtonian liquid near a solid surface (LAUGA; BRENNER; STONE, 2005).

One way to incorporate this discontinuous motion is by allowing slip of the fluid relative to the solid boundary, which is described as a relation between slip velocity and surface shear stress (DURBIN, 1988).

A linear boundary condition was introduced by Navier (1823) and it is still the standard characterization of slip (LAUGA; BRENNER; STONE, 2005). The Navier-slip condition is a widely accepted boundary condition to relieve the singularity at the moving contact line (GANESAN; TOBISKA, 2009), and gives essentially the same macroscopic flow, except

if you look close to the contact line (DAVIS, 2014). The component of the fluid velocity tangential to the surface $\mathbf{u}_{||}$, is proportional to the rate of strain (or shear rate) at the surface:

$$\mathbf{u}_{||} = \lambda \mathbf{n} \cdot (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \cdot (\mathbf{1} - \mathbf{n}\mathbf{n}) \quad (2.14)$$

where \mathbf{n} denotes the normal to the surface, directed onto the liquid, and λ is the slip length, and can be interpreted, for a pure shear flow, as the fictitious distance below the surface where the no-slip boundary condition would be satisfied (LAUGA; BRENNER; STONE, 2005).

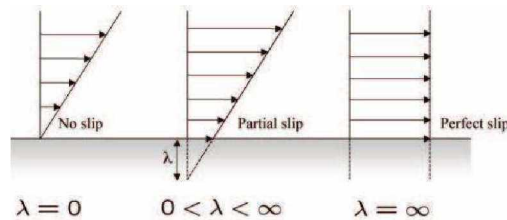


Figure 2.16: Interpretation of slip length. Source: Lauga, Brenner and Stone (2005)

The Navier slip boundary condition is a popular treatment to remove the moving contact line singularity: it is well-established experimentally and theoretically, and it is easily incorporated into a continuum description (SNOEIJER; ANDREOTTI, 2013). The advantage of the Navier slip model is that it specifies a slip length instead of relying on the mesh-dependent effective slip of the numerical discretization (AFKHAMI; ZALESKI; BUSSMANN, 2009). Two conditions are prescribed: 1) no penetration boundary condition (normal component of the velocity is zero); 2) the slip with friction boundary condition, in which the tangential velocities of the fluids are proportional to their corresponding tangential stress (GANESAN; TOBISKA, 2009).

The surface stress is imagined to increase as the contact line is approached, until a critical level is reached at which the liquid begins to slip, in a behavior that is loosely analogous to the phenomenon of yields in metals. The location at which the liquid begins to slip is not prescribed, it is simply the location where the surface stress reaches its critical level, which is a property of the fluid flow (DURBIN, 1988).

However, the method has some limitations. The main problem is that the slip condition introduces only a length scale, but not an energy scale that expresses the interaction with the solid wall. In practice this means that the hydrodynamic equations still lack a boundary condition for the microscopic contact angle, which is necessary to close the problem. Second, the introduction of slip regularizes the divergence of shear stress and energy dissipation, but it still leads to a logarithmically divergent pressure $p \sim (\eta^U/s) \ln(h/s)$ (BERTOZZI; SHEARER; BUCKINGHAM, 2003 apud SNOEIJER; ANDREOTTI, 2013).

Another model is Huh and Mason (apud DURBIN, 1988) free-slip condition, in which, at a given distance of the contact line the liquid slips freely over the solid, so that the shear rate equals zero, and a no-slip condition is applied elsewhere.

Both the no-slip and the free-slip can be seen as the extreme conditions of the Navier-Condition, in which the slip length parameter ($0 < \lambda < \infty$), allows for a slip velocity tangential to the wall. The three different boundary condition can be achieved, depending on the choice of λ , as illustrated by Fig. 2.16: a) No slip ($\lambda = 0$); b) partial slip $0 < \lambda < \infty$; c) Free slip $\lambda = \infty$.

The no-slip and the Navier-slip boundary conditions are studied by Afkhami, Zaleski and Bussmann (2009). Accordingly to the authors, the numerical model introduces an effective slip which is on the scale of mesh size. They choose a slip length $\lambda = 0.001$, that is approximately twice smaller than the finest mesh size they used on the simulations. The results obtained when the Navier-slip condition is applied underestimate actual values of contact line height and maximum shear rate, unless a very small slip length is employed.

2.5 Dynamic contact angle - empirical models

The most widespread working relation describing the contact angle is given by the so-called *Hoffman-Voinov-Tanner law* for capillary-dominated situations (low Ca number) (SIKALO; TROPEA; GANIC, 2005):

$$Ca = k(\theta_D - \theta_S)^3, \quad Ca = \frac{\mu U_{cl}}{\sigma} \quad (2.15)$$

For inertia-dominated situations (high Ca number) the constant angles maximum advancing θ_{mda} and minimum receding θ_{mdr} are used instead. This leads to:

$$\theta_D = \min\left[\theta_S + \left(\frac{Ca}{k}\right)^{1/3}, \theta_{mda}\right], \quad \text{if } U_{CL} \geq 0 \quad (2.16)$$

$$\theta_D = \max\left[\theta_S + \left(\frac{Ca}{k}\right)^{1/3}, \theta_{mdr}\right], \quad \text{if } U_{CL} < 0 \quad (2.17)$$

Other correlations are:

- Kistler (1993):

$$\theta_D = f_{Hoff}[Ca + f_{Hoff}^{-1}\theta_S] \quad (2.18)$$

with:

$$f_{Hoff} = \arccos \left[1 - 2 \tanh \left[5.16 \left[\frac{x}{1 + 1.31x^{0.99}} \right]^{0.706} \right] \right] \quad (2.19)$$

- Jiang, Oh and Slattery (1979):

$$\frac{\cos(\theta_S) - \cos(\theta_D)}{\cos(\theta_S) + 1} = \tanh(4.96Ca^{0.706}) \quad (2.20)$$

- Bracke, De Voeght and Joos (1989)

$$\frac{\cos(\theta_S) - \cos(\theta_D)}{\cos(\theta_S) + 1} = 2Ca^{0.5} \quad (2.21)$$

- Seebergh and Berg (1992)

$$\frac{\cos(\theta_S) - \cos(\theta_D)}{\cos(\theta_S) + 1} = K_1 Ca^{K_2} \quad (2.22)$$

$$Ca > 10^{-3}: K_1 = 2.24, K_2 = 0.54;$$

$$Ca \leq 10^{-3}: K_1 = 4.47, K_2 = 0.42$$

2.6 Dynamic contact angle: mathematical model

A different approach, named Interface Formation Model (IFM), is proposed by Shikhmurzaev (1993, 1994), Sprittles and Shikhmurzaev (2012), that criticises the “conventional” approach of using correlation relating the contact angle θ to the velocity of the contact line. Another criticism is that these models should have a slip-condition at the contact point. Billingham (2008) evaluates how the IFM affects the predictions for the gravity-driven flow of a thin film down an inclined plane, Sibley, Savva and Kalliadasis (2012) study the spreading of a thin two-dimensional droplet on a planar substrate, and Griebel and Klitz (2013) simulates droplet impact on a dry flat surface.

A key element of the IFM is the fact that as a liquid advances across a solid surface, liquid at the liquid/gas interface becomes transferred to the solid/liquid interface; there is

a material flux through the contact line, which is missing from the conventional models (BLAKE, 2006).

In this method, a microscopic layer is considered between each phase. In the continuum limit, these layers becomes a mathematical surface of zero thickness. The main aspect of the method is that, although with negligible thickness, the layer have properties, such as, surface density (ρ^s), surface velocity (\mathbf{v}^s) and surface tension σ^s . Accordingly to Shikhmurzaev (1994), Sprittles and Shikhmurzaev (2012), Billingham (2006), the main advantages of the IFM are:

1. Considers the interface as a thermodynamic system with mass, momentum and energy exchange with the bulk.
2. Contact angle determined by the flow field

Two interrelated mechanisms are paramount to the formulation of the IFM:

- Surface layers: liquid molecules near the surface being in an unfavorable energy state
- Surface tension relaxation

With the interface surfaces, it is possible to consider different conditions in each side of the layers. Thus, the no-slip condition is satisfied for fluid particles next to the solid.

The surface tension relaxation is related to the horizontal force balance from Young equation: $\sigma_{12}\cos\theta_s + \sigma_{1s} = \sigma_{2s}$. The dynamic contact angle varies from its static value, causing either σ_{12} , σ_{1s} , σ_{2s} or a combination to also deviate from their static value. Fluid particle are transferred from the gas interface (associated to σ_{12}) to the solid interface (associated to σ_{1s}). This relaxation of the surface tension to the equilibrium existing far from the contact line happens in a finite time τ , rather than instantaneously. The developments and a detailed explanation of the method can be found at Shikhmurzaev (1993, 1994), Billingham (2006), Sibley, Savva and Kalliadasis (2012).

Griebel and Klitz (2013) works with a reduced version of the Interface Formation model at small capillary and Reynolds number; the dynamic contact angle and a dimensionless contact-line velocity V are related by:

$$\cos(\theta_s) - \cos(\theta_d) = \frac{2V[\cos(\theta_s) - \sigma_{sg} + (1 + \rho_g^s)^{-1}(1 + \rho_G^s u_{(12)}(\theta_d, \kappa_\mu))]}{V + [V^2 + 1 + (\cos(\theta_s) - \sigma_{sg})(1 - \rho_G^s)]^{1/2}} \quad (2.23)$$

$$\rho_G^s = 1 - \frac{\sigma_{sg} - \sigma_{sl}}{\lambda \cos\theta_d} \quad (2.24)$$

where θ_s is the static contact angle, κ_μ is the gas-to-liquid viscosity ratio, σ_{sg} and σ_{sl} denote the surface tension of the gas-solid and liquid-solid interface, respectively, λ is a material

parameter and $u_{(12)}(\theta_d, \kappa_\mu)$ is a radial velocity derived from the solution of the outer region, being:

$$u_{(1,2)}(\theta_d, 0) = \frac{\sin\theta_d - \theta_d \cos\theta_d}{\sin\theta_d \cos\theta_d - \theta_d} \quad (2.25)$$

or, if the viscosity of the gas phase is taken into account:

$$u_{(1,2)}(\theta_d, \kappa_\mu) = \frac{(\sin\theta_d - \theta_d \cos\theta_d)K(\theta_2) - \kappa_\mu(\sin\theta_2 - \theta_2 \cos\theta_2)K(\theta_d)}{(\sin\theta_d \cos\theta_d - \theta_d) + \kappa_\mu(\sin\theta_2 \cos\theta_2 - \theta_2)K(\theta_d)} \quad (2.26)$$

with $\theta_2 = \pi - \theta_d$ and $K(\theta) = \theta^2 - \sin^2\theta$.

In Eq.(2.23) $u_{12}(\theta_s, \kappa_\mu)$ can be replaced by a numerically computed far field velocity sufficiently close to the contact line.

The dimensionless velocity is:

$$V = \frac{u_{cl}}{U} = \frac{u_{cl}\mu}{\sigma} Sc \quad (2.27)$$

with:

$$U = \sqrt{\frac{\gamma\rho_0^s(1 + 4\alpha\beta)}{\tau\beta}} \quad (2.28)$$

$$Sc = \sqrt{\frac{\sigma^2\tau\beta}{\mu^2\gamma\rho_0^s(1 + 4\alpha\beta)}} \quad (2.29)$$

where σ is the equilibrium surface tension, α and β are phenomenological constants depending on the “state of the interface”, γ is a phenomenological constant describing the compressibility of the fluid, τ is the surface tension relaxation time and ρ_0^s is the surface density for zero surface tension, both of which can be treated as material constants, and Sc can be chosen to fit the numerical results to the experimental data (GRIEBEL; KLITZ, 2013).

Table (2.1) presents the physical parameters estimated by Sibley, Savva and Kalliadasis (2012) for water at room temperature for the Interface Formation Model:

Table 2.1: Physical parameters estimated for water at room temperature for the Interface Formation Model (SIBLEY; SAVVA; KALLIADASIS, 2012)

$\gamma = 2.10^6 [m^2/s^2]$	$\tau = 10^{-8} [s]$
$\rho = 1000 [Kg/m^3]$	$\rho_{se}^s = \rho_{ge}^s = 10^{-7} [Kg/m^2]$
$\mu = 10^{-3} [Kg/m.s]$	$\beta = 10^7 [Kg/m^2.s]$
$\alpha\beta = \frac{1}{12} [dimensionless]$	$\sigma_{ge}^s = 7.10^{-2} [N/m]$

CHAPTER III

METHODOLOGY

The mathematical modeling and the spatial and temporal discretization used in the *AMR3D* code are presented. Their implementation and validation are described by Villar (2007) and N3s (2007).

For the VOF method, several steps are necessary. The implemented initialization process is explained in Section 3.6. The normal and curvature computation, the color function reconstruction (a step in which the interface geometry is approximated from the color function and its gradients) and the VOF advection (the step where this information is updated due the velocity flow field) are explained in Sections 3.7, 3.8 and 3.9, respectively.

The numerical procedure to apply the contact angle is explained in Section 3.10.

3.1 Mathematical Modeling

3.1.1 Eulerian formulation

The one-fluid formulation of the Navier-Stokes equation is used. A two-phase flow without phase change is modeled as a single phase flow with variable density and viscosity. The fluid is also considered to be incompressible. This formulation allows to treat the multi-phase flow in a similar fashion than a homogeneous fluid and any standard algorithm based on fixed-grids can, in principle, be used to integrate the discrete Navier-Stokes equations in time (PROSPERETTI; TRYGGVASON, 2007).

The Navier-Stokes equation for isotherm incompressible flow, in its vectorial form are (WHITE, 2006):

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] - \nabla p + \rho \mathbf{g} + \mathbf{f}_\sigma + \mathbf{f}, \quad (3.1a)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (3.1b)$$

where \mathbf{u} is the fluid velocity field, ρ is the fluid specific mass, μ is the fluid dynamic viscosity, p is the pressure, including the fluid-static part, \mathbf{g} is the gravity, \mathbf{f}_σ is the interface tension force and \mathbf{f} represents other forcing terms, such as the one to account for the Immersed Boundary.

The behavior of the fluid-fluid interface is modeled including a source term for the interface tension force. In the front-tracking method of Tryggvason, this force field is calculated on a separate framework with a Lagrangian reference frame, which explicitly represents the position of the interface (UNVERDI; TRYGGVASON, 1992). In the VOF method, the volume fraction (F) is used for associating each cell in the Eulerian domain to the respective fluid phase.

The surface tension is interpreted as a continuous, three-dimensional effect across an interface. In the absence of mass transfer, there is no need of establishing any additional jump conditions at the fluid interface as they are implicitly taken into account by this formulation. The coupling of the two fluids is provided by variable material properties ρ and μ (BOGER; SCHLOTTKE; MUNZ, 2010). They are computed as:

$$\phi = F \cdot \phi_1 + (1 - F) \cdot (\phi_2), \quad (3.2)$$

where ϕ_1 is the scalar property of fluid 1, ϕ_2 is the scalar property of fluid 2 and F is the color function at the computational cell.

The forcing term \mathbf{f} allows for the communication between the Navier-Stokes equation and the Immersed Boundary. It has value only at the interface and must be zero elsewhere.

3.1.2 Eulerian - Lagrangian coupling

The Immersed Boundary is discretized in a Lagrangian mesh, independent of the Eulerian mesh used to solve the Navier-Stokes equations. The Dirac delta function δ is used to communicate between the Eulerian and the Lagrangian domains, being approximated by a distribution function with Gaussian properties. The 3D distribution is approximated by the product of three 1D functions, accordingly to Eq(3.3):

$$D(\mathbf{x} - \mathbf{X}) = \frac{1}{\Delta x \Delta y \Delta z} W\left(\frac{x - X}{\Delta x}\right) W\left(\frac{y - Y}{\Delta y}\right) W\left(\frac{z - Z}{\Delta z}\right), \quad (3.3)$$

where

$$W(r) = \begin{cases} \frac{1}{4}(1 + \cos(\frac{\pi}{2}r)), & r < 2, \\ 0, & r \geq 2, \end{cases} \quad (3.4)$$

and

$$r = \frac{x - X}{\Delta x}, \frac{y - Y}{\Delta y}, \frac{z - Z}{\Delta z}. \quad (3.5)$$

It is responsible for distributing the Lagrangian interface force to the Eulerian points [Eq.(3.6)] and for interpolating the Eulerian velocities to the Lagrangian points of the interface [Eq.(3.7)]. These operations are called *spreading* and *interpolation*, respectively, and its mathematical formulation are

$$\mathbf{f}(\mathbf{x}, t) = \int \mathbf{F}(\mathbf{X}, t) D(\mathbf{X} - \mathbf{x}) d\mathbf{X}, \quad (3.6)$$

where t is the time, \mathbf{X} represents the position in the Lagrangian framework, \mathbf{F} represents the interface tension force field in the Lagrangian framework, \mathbf{x} represents the position in the Eulerian framework and \mathbf{f} is the interface tension force field in the Eulerian framework.

$$\mathbf{U}(\mathbf{X}, t) = \int \mathbf{u}(\mathbf{x}, t) D(\mathbf{x} - \mathbf{X}) d\mathbf{x} \quad (3.7)$$

where \mathbf{U} represents the velocity field in the Lagrangian framework and \mathbf{u} is the velocity field in the Eulerian framework.

Numerically, the integrals (3.6) and (3.7) are replaced by a discrete summation over a four cells-thick zone around the interface.

3.2 Numerical method: temporal discretization

A semi-implicit temporal discretization method is used in the *AMR3D* code, based on the second order Implicit-Explicit Schemes (IMEX), as described by Ascher, Ruuth and Wetton (1995). The diffusive term is treated implicitly and the advective term explicitly (VILLAR, 2007).

Through the IMEX schemes, a family of temporal discretization is obtained. The second-order temporal discretization of the Navier-Stokes through the semi-implicit scheme is (VILLAR, 2007; PIVELLO, 2012):

$$\frac{\rho^{n+1}(\phi)}{\Delta t} (\alpha_2 \mathbf{u}^{n+1} + \alpha_1 \mathbf{u}^n + \alpha_0 \mathbf{u}^{n-1}) = \beta_1 f(\mathbf{u}^n) + \beta_0 f(\mathbf{u}^{n-1}) + \quad (3.8a)$$

$$\lambda [\theta_2 \nabla^2 \mathbf{u}^{n+1} + \theta_1 \nabla^2 \mathbf{u}^n + \theta_0 \nabla^2 \mathbf{u}^{n-1}] - \nabla p^n + \rho^{n+1}(\phi) \mathbf{g}, \quad (3.8b)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0,$$

where the superscripts $(n + 1)$, (n) and $(n - 1)$ are relative to next, the current and the

previous time steps, respectively; ϕ is an indicator function for the fluid, being 0 for the continuous phase and 1 for the disperse phase; $\lambda = \|\mu\|_\infty$ and $f(\mathbf{u})$ depends on the diffusive, advective and forcing terms:

$$f(\mathbf{u}) = -\lambda \nabla^2 \mathbf{u} + \nabla \cdot \left[\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \right] - \mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{f}_\sigma \quad (3.9)$$

and the parameters $\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \theta_0, \theta_1$ and θ_2 from Eq. 3.8a are:

$$\begin{aligned} \alpha_0 &= \frac{(2\gamma - 1)\omega^2}{1 + \omega}, \\ \alpha_1 &= (1 - 2\gamma)\omega - 1, \\ \alpha_2 &= \frac{1 + 2\gamma\omega}{1 + \omega}, \\ \beta_1 &= -\gamma\omega, \\ \beta_0 &= 1 + \gamma, \\ \theta_0 &= \frac{c}{2}, \\ \theta_1 &= 1 - \gamma - \left(1 + \frac{1}{\omega}\right) \frac{c}{2}, \\ \theta_2 &= \gamma + \frac{c}{2\omega}, \end{aligned} \quad (3.10)$$

where $\omega = \Delta t^{n+1}/\Delta t^n$ is the ratio between two consecutive time steps, and the two parameters γ and c allows to choose the method from the IMEX scheme: (ASCHER; RUUTH; WETTON, 1995; PIVELLO, 2012)

- Crank-Nicholson Adams-Bashforth (CNAB): $(\gamma, c) = (0.5, 0.0)$;
- Modified Crank-Nicholson Adams-Bashforth (MCNAB): $(\gamma, c) = (0.5, 0.125)$;
- Crank-Nicholson Leap Frog (CNLF): $(\gamma, c) = (0.0, 1.0)$;
- Semi-Backward Difference (SBDF): $(\gamma, c) = (1.0, 0.0)$.

With this discretization two time steps are required. To allow its applicability, the Navier-Stokes equations are solved with an Euler temporal discretization for the first time step of the simulation.

The time step is calculated following stability criteria for explicit schemes, taking into account the advective, diffusive and capillary terms, as defined by Eq.(3.11):

$$\Delta t_{adv} = \frac{1}{\frac{\|u\|_\infty}{\Delta x} + \frac{\|v\|_\infty}{\Delta y} + \frac{\|w\|_\infty}{\Delta z}}, \quad (3.11a)$$

$$\Delta t_{diff} = a_1 h, \quad (3.11b)$$

$$\Delta t_{cap} = \sqrt{\left(\frac{\rho_C + \rho_D}{2}\right) \frac{h^3}{\pi \sigma}}, \quad (3.11c)$$

$$\Delta t = \min(b_1 \Delta t_{adv}, b_2 \Delta t_{diff}, b_3 \Delta t_{cap}). \quad (3.11d)$$

where Δt_{adv} , Δt_{diff} and Δt_{cap} are the maximum time step allowed by the advective term, the diffusive term and the interface force, respectively. $\|u\|_\infty$, $\|v\|_\infty$ and $\|w\|_\infty$ are the infinite-norm values of the velocity components u , v and w ; $h = \min(\Delta x, \Delta y, \Delta z)$ is the characteristic length of the grid and Δx , Δy and Δz are the grid spacing in the x-, y- and z- directions. $0 < a_1 \leq 1$, $0 < b_1 \leq 1$, $0 < b_2 \leq 1$ and $0 < b_3 \leq 1$ are safety coefficients. Equation (3.11b) requires that a_1 must have dimensions of time per length, that is: $[a_1] = TL^{-1}$.

According to Tryggvason et al. (2001), the constraint for second order schemes for the diffusive term is:

$$\Delta t \leq \frac{\rho h^2}{6\mu}. \quad (3.12)$$

Making $\Delta t_1 = a_1 h$ and $\Delta t_2 = \frac{\rho h^2}{6\mu}$ yields the following relation:

$$\frac{\Delta t_2}{\Delta t_1} = h \frac{\rho}{6a_1 \mu} \quad (3.13)$$

Since $a_1 \leq 1$ and, in the context of gas-liquid flows $\rho \gg 1$ and $\mu < 1$, Eq.(3.13) shows that the constraint adopted in this work is stricter than Eq.(3.12). The same can be stated about the capillary constraint since, in the present work, the mean density is adopted.

In the present work both the SBDF, CNAB and MCNAB temporal time discretization are used in the simulated cases with a variable time step.

3.3 Numerical method: spatial discretization

The Navier-Stokes equations are solved in a computational domain $\Omega = [A_1, B_1] \times [A_2, B_2] \times [A_3, B_3]$ locally refined. Initially the domain is discretized with a regular mesh containing $[N_x \times N_y \times N_z]$ computational cells with dimensions $\Delta_x = \frac{B_1 - A_1}{N_x}$, $\Delta_y = \frac{B_2 - A_2}{N_y}$

and $\Delta_z = \frac{B_3 - A_3}{N_z}$ (VILLAR, 2007).

Space is discretized using a Structured Adaptive Mesh Refinement (SAMR) framework, which is based on the version of the Immersed Boundary (IB) method introduced by Roma, Peskin and Berger (1999), and in the hierarchical grid structure proposed by Berger and Colella (1989). A block-structured local refinement assures that regions of interest, such as the transition between the two phases, are on the spacial mesh with bigger resolution, as illustrated in Fig. 3.1

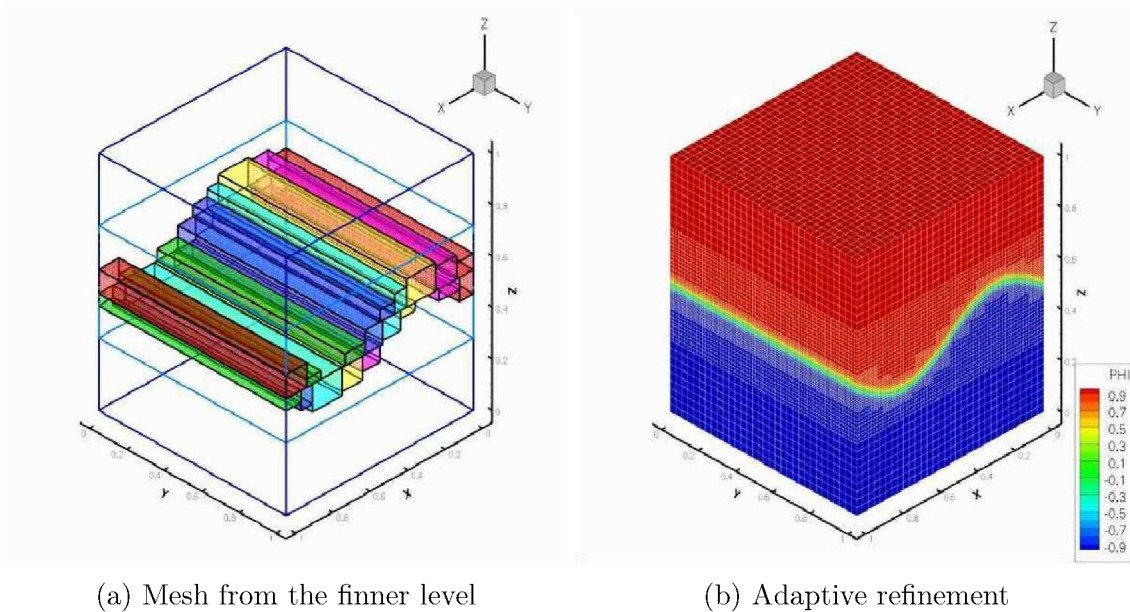


Figure 3.1: Example of mesh refinement in a two-phase flow. source: N3s (2007)

The dimension of the computational cell in each refinement level is:

$$\Delta x_{i,l+1} = \frac{\Delta x_{i,l}}{r} \quad (3.14)$$

where $i = 1, 2, 3$ corresponds to each coordinate axis, l identifies the level of refinement, from l_{base} (the coarse mesh resolution) to l_{top} (the most refined level) and $r = 2$ is the ratio of refinement between two successive levels.

The center of each computational cell is defined as:

$$\mathbf{x}_{i,j,k} = (x_i, y_j, z_k) = [A_1 + (i - 1/2)\Delta x, A_2 + (j - 1/2)\Delta y, A_3 + (k - 1/2)\Delta z] \quad (3.15)$$

for $1 \leq i \leq N_x$, $1 \leq j \leq N_y$, $1 \leq k \leq N_z$

A staggered composite grid is used, *i.e.*, pressure and other scalar variables are computed at the centers of the computational cells and vector variables are located at the cell faces (Fig. 3.2).

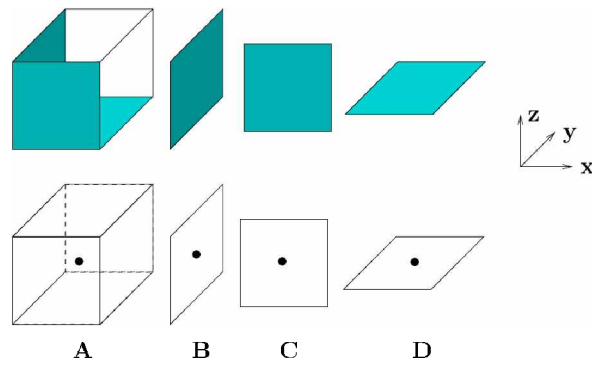


Figure 3.2: Computational cell; A) position of the scalar variables; B) position of the vector variables in the x direction; C) position of the vector variables in the y direction; D) position of the vector variables in the z direction. source: N3s (2007)

To account for the boundary condition, ghost cells are added around the domain, as illustrated in Fig. 3.3

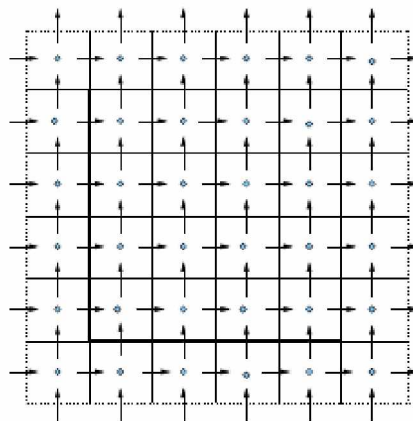


Figure 3.3: Computational mesh with ghost cells represented by the dashed lines and staggered variables represented by the arrows at the cell faces. source: Villar (2007)

The spatial discretization of the Navier-Stokes equation requires the discretization of the operators Gradient, Divergent and Laplacian. This is performed on the staggered grid with a second order central difference scheme. On the Eulerian cell (i, j, k) , a scalar variable is denoted by $\phi_{(i,j,k)}$ and a vector variable, such as the velocity \mathbf{u} is defined as:

$$\mathbf{u}_{i,j,k} = \left(u_{i-\frac{1}{2},j,k}, v_{i,j-\frac{1}{2},k}, w_{i,j,k-\frac{1}{2}} \right) \quad (3.16)$$

The Gradient operator is defined as:

$$\nabla \phi_{i,j,k} = \left(\nabla_x \phi_{i-\frac{1}{2},j,k}, \nabla_y \phi_{i,j-\frac{1}{2},k}, \nabla_z \phi_{i,j,k-\frac{1}{2}} \right) \quad (3.17)$$

where:

$$\nabla_x \phi_{i-\frac{1}{2},j,k} = \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x} \quad (3.18)$$

$$\nabla_y \phi_{i,j-\frac{1}{2},k} = \frac{\phi_{i,j,k} - \phi_{i,j-1,k}}{\Delta y} \quad (3.19)$$

$$\nabla_z \phi_{i,j,k-\frac{1}{2}} = \frac{\phi_{i,j,k} - \phi_{i,j,k-1}}{\Delta z} \quad (3.20)$$

The divergent operator is:

$$(\nabla \cdot \mathbf{u})_{i,j,k} = \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\Delta x} + \frac{v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}}{\Delta y} + \frac{w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}}{\Delta z} \quad (3.21)$$

And the Laplacian is:

$$(\nabla^2 \phi)_{i,j,k} = \nabla \cdot \nabla \phi_{i,j,k} = \frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta x^2} + \quad (3.22)$$

$$\frac{\phi_{i,j+1,k} - 2\phi_{i,j,k} + \phi_{i,j-1,k}}{\Delta y^2} + \quad (3.23)$$

$$\frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\Delta z^2} \quad (3.24)$$

3.3.1 Structured Adaptive Mesh Refinement

In the Structured Adaptive Mesh Refinement approach, regions of the flow with special interest are covered by block-structured grids, defined as a hierarchical sequence of nested, progressively finer levels (*composite grids*). Each level is formed by a set of disjoint rectangular grids and the refinement ratio between two successive refinement levels are constant and equal to two. Ghost cells are employed around each grid, for all the levels, and underneath fine grid patches to formally prevent the finite difference operators from being redefined at grid borders and at interior regions which are covered by finer levels. Values defined in these cells are obtained from interpolation schemes, usually with second or third order accuracy, and not from solving the equations of the problem (ROMA, 1996).

Although a variable can be defined or initialized in any level, in the current work the Lagrangian or VOF interface must to be completely covered by the finest level, ensuring that the most important physical phenomena are being captured.

Within the grid hierarchy, the levels are nested so that the coarsest level covers the entire computational domain and each finer level covers a part of the interior of the previous

coarse level. In each level, the grid is composed by the union of logically rectangular regions called patches or blocks, which do not intersect each other. All grid blocks in a given refinement level share the same grid spacing (BERGER; OLIGER, 1984) (BERGER; COLELLA, 1989).

The refinement ratio (r) is defined as the quotient between the grid spacing of two consecutively refined levels and is constant for all levels. In the present work, $r = 2$ is used. Also, the grid hierarchy must obey two conditions in order to be properly nested:

1. The corners of a block in a given level of refinement must match the corners of cells located at the next coarser level.
2. A grid block in a given level must not touch the boundary of a block belonging to a coarser level of refinement, except when these blocks lie at the boundaries of the computational domain.

Figure 3.4 shows the case of a properly nested grid hierarchy with three levels of refinement and Fig. 3.5 shows two grids violating these conditions: grid (a) violates condition 1 and grid (b) violates condition 2.

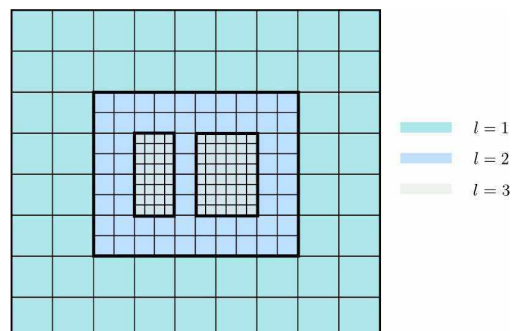


Figure 3.4: An example of block structured adaptive mesh refinement properly nested. Source: Villar (2007).

Grid adaptation involves selecting cells to be refined at a given level, using some error estimator or refinement criterion, and then grouping the cells into blocks in that region. These blocks are used for creating the next refinement level in the hierarchy. When the remeshing criteria are based on the magnitude of a particular variable, ranges of this variable are defined so that each refinement level covers a different range of values of that variable. In this work, the following approach was followed: let the remeshing criterion be based on the magnitude of some variable $\Phi \in [\Phi_{min}; \Phi_{max}]$. This variable is mapped onto the interval $[0; 1]$, which is further divided into as many intervals as the number of refinement levels, so that the finer refinement levels are applied on the top of the interval. Determining the range for each refinement level may depend on the kind of problem being solved. In the present work, the following rule is adopted: the finest level covers the interval $[0.1; 1.0]$ and the complement of the domain is equally divided between the remaining refinement levels.

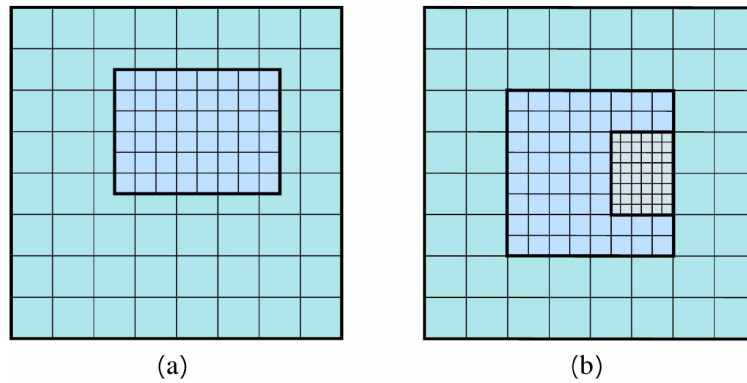


Figure 3.5: An example of block structured adaptive mesh refinement violating the nesting rules: the fine block in grid (a) does not lie on vertices of the coarse level block and the block in the finest level in grid (b) touches the boundary of a block in the next coarse level which is not part of the domain boundary. Source: Villar (2007).

Figure 3.6 shows an example of SAMR grid taken from a case simulating a bubble rising in the wobbling regime. The bubble creates a von Karman wake and the mesh refinement is placed over the bubble and over the bubble wake.

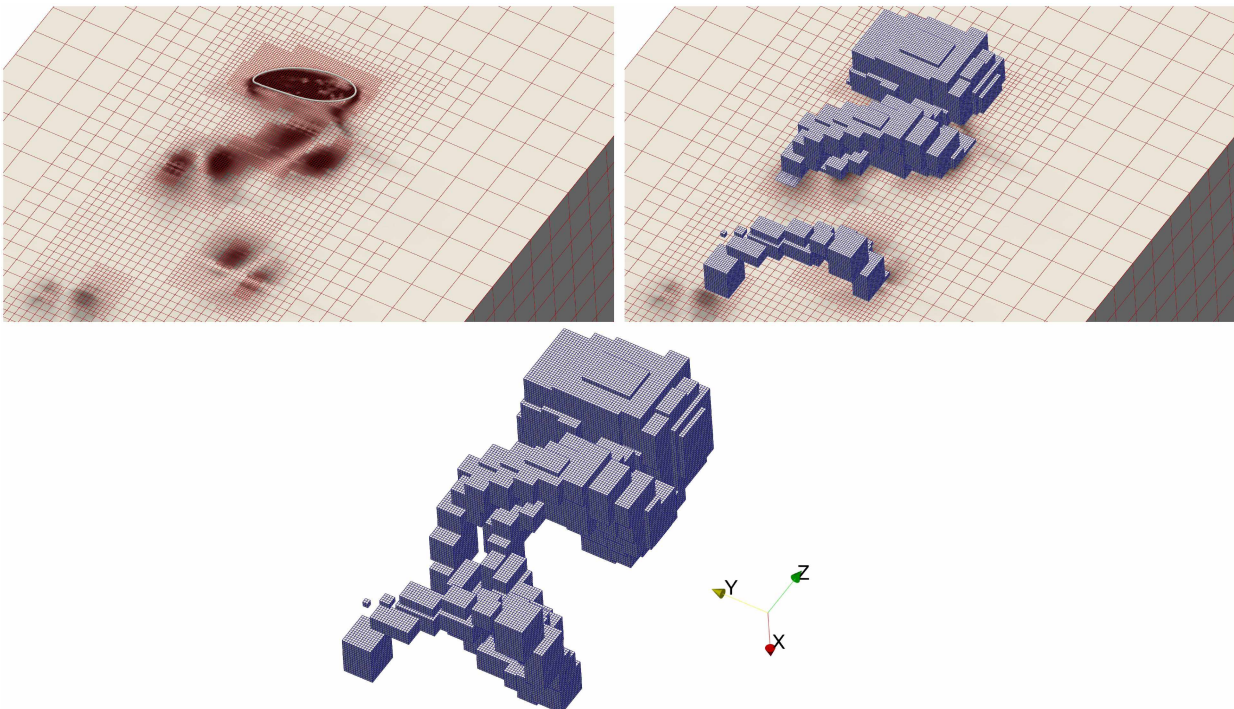


Figure 3.6: Example of adaptive mesh refinement based on the vorticity magnitude. Source: Pivello (2012)

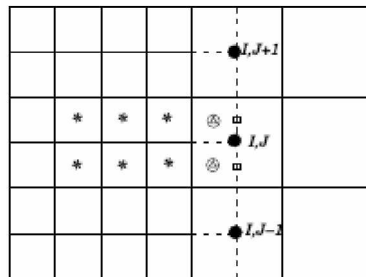
3.3.2 Ghost cells

The ghost cells store the values for the boundary conditions, and are defined in order to avoid a redefinition of the differential operator at the borders of the mesh. This allows the use of the same computational stencil inside the domain and at its borders. The number of

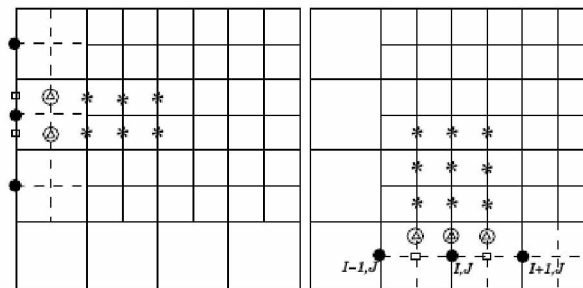
ghost cells depends on the stencil size. For a second order discretization the stencil for the Laplacian has five cells. The boundary condition for the ghost cells are of three types. The first involves an interpolation procedure from the coarse to the finer mesh to determine the values on the ghost cells that are in any mesh of the same level. On the second, known as *injection*, the values are taken from values already determined by sister cells. Finally, the third procedure consist in replacing the values from the ghost cells at the borders of the domain by the proper boundary condition.

For cell centered variables, quadratic polynomials are chosen for the interpolation procedure between the finer/coarse interface. This is illustrated schematically in Fig. 3.7a. The value of the ghost cell for the finer level (\circ) is obtained from an extrapolation of cells from the same finer level ($*$), followed by an interpolation from the coarse cells (\bullet). The final result is from an interpolation between the extrapolated value (Δ), the value at the finer cell ($*$) and the value interpolated from the coarse level (\square).

The procedure for staggered variables is similar to the one for centered variables. The main difference is in the computational mesh. Figure 3.7b illustrates the procedure for the west (left) and south (right) borders.



(a) Interpolation stencil for the ghost cells for a centered variable.



(b) Interpolation stencil for the ghost cells for a staggered variable at west (left) and south (right) borders.

Figure 3.7: Interpolation stencil. $*$ value at the finer level, \bullet value at the coarse level, Δ extrapolated value from the finer level, \square interpolated value from the coarse level, \circ final value at the ghost cell. source: Villar (2007)

3.4 Fractional step method

After the temporal and spacial discretization of Eq.(3.8a) and (3.8b) a fractional step method (KIM; MOIN, 1985) is used to allow for the couple between velocity and pressure. First, an auxiliary velocity field \mathbf{u}^* is advanced using pressure from the previous time step; later, the requirement that the new velocity satisfy the continuity equation leads to a Poisson equation for the new pressure. Upon solution of the pressure equation, the new velocity field is obtained, which satisfy the continuity and the momentum equations. The procedure is:

1. From Eq. (3.8a), compute the auxiliary velocity field:

$$\begin{aligned} \frac{\rho(\phi)^{n+1}}{\Delta t}(\alpha_2 \mathbf{u}^* + \alpha_1 \mathbf{u}^n + \alpha_0 \mathbf{u}^{n-1}) = & \beta_1 f(\mathbf{u}^n) + \beta_0 f(\mathbf{u}^{n-1}) + \\ \lambda \left[\theta_2 \nabla^2 \mathbf{u}^* + \theta_1 \nabla^2 \mathbf{u}^n + \theta_0 \nabla^2 \mathbf{u}^{n-1} \right] - & \nabla p^n + \rho^{n+1}(\phi) \mathbf{g} \end{aligned} \quad (3.25)$$

2. The requirement that the new velocity satisfy the continuity equation leads to a Poisson equation for the pressure correction q :

$$\nabla \left[\frac{1}{\rho^{n+1}} \nabla q^{n+1} \right] = \frac{\alpha_2}{\Delta t} \nabla \mathbf{u}^* \quad (3.26)$$

3. Correct the velocity field:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t \nabla q^{n+1}}{\alpha_2 \rho n + 1} \quad (3.27)$$

4. Correct the pressure:

$$p^{n+1} = p^n + q^{n+1} \quad (3.28)$$

5. Assure that the continuity equation, with the new velocity field \mathbf{u}^{n+1} is satisfied:

$$\nabla \mathbf{u}^{n+1} = 0 \quad (3.29)$$

3.5 Surface tension force

In the one-fluid formulation, a single set of equations for the whole flow domain is used, and the coupling of the fluids at the interface is taken into account by the use of variable density and viscosity and the surface tension, considered as a body force (PROSPERETTI; TRYGGVASON, 2007). The change in material properties has to happen in a smooth regularized way, in order to prevent numerically induced oscillations in the vicinity of the interface. The corresponding force \mathbf{F}_σ on the right-hand side of the equation is only present at the interface while it vanishes in grid cells away from it. In principle (for steady state) one could use directly the Laplace equation in order to derive the volume force \mathbf{F}_σ (BOGER; SCHLOTTKE; MUNZ, 2010)

$$\Delta P = \sigma \kappa, \tag{3.30a}$$

↓

$$\mathbf{F}_\sigma = \sigma \kappa \mathbf{n} \tag{3.30b}$$

where \mathbf{n} is the interface unit vector.

“Though, as we are dealing with a diffuse interface formulation, we are in need of some numerical smoothing operations near the interface. Therefore, surface tension has to be calculated based on a numerical model” (BOGER; SCHLOTTKE; MUNZ, 2010).

The evaluation of the surface tension force (\mathbf{F}_σ) on the momentum equation is one of the most difficult tasks in the VOF framework (POPINET, 2009). Implementing Eq.(3.30a) in a numerical scheme involves a twofold task: the curvature κ needs to be determined, and the resulting pressure jump Δp must be applied appropriately to the fluids (MEIER; YADIGAROGLU; SMITH, 2002).

Numerically, this condition is extremely difficult to apply and two paths to its evaluation are common in the literature: 1) the capillary forces are applied as a boundary condition along the free surface, considered as a discontinuity separating the two fluids. The equations are solved separately for each phase and coupled through the boundary condition (KANG; FEDKIW; LIU, 2000); 2) in a second approach, proposed by Brackbill, Kothe and Zemach (1992), since the interface location is not known *a priori*, the surface tension force is interpreted as a body force applied to a small number of cells. Following this idea, the most common approach applied in the VOF methods are the *Continuum Surface Force* (CSF) (BRACKBILL; KOTHE; ZEMACH, 1992) and the *Continuum Surface Stress* (CSS) (LAFURIE et al., 1994).

The CSF interprets surface tension as a continuous, three-dimensional effect across an interface, rather than as a boundary value condition on the interface. The interface where the fluid changes from fluid 1 to fluid 2 discontinuously is replaced by a continuous (smooth) transition. Thus, instead of considering the fluidic interface as a sharp discontinuity, the

interface is considered to have a finite thickness of $O(h)$, corresponding to the smallest length scale resolvable by the computational mesh. Consequently, surface tension is also considered to be of continuous nature and it acts everywhere within the transition region. The equivalent body force (\mathbf{F}_σ) is (BRACKBILL; KOTHE; ZEMACH, 1992):

$$\mathbf{F}_\sigma = \sigma\kappa\delta\mathbf{n} \quad (3.31)$$

where σ is the surface tension coefficient, κ is the interface curvature, δ is a delta function indicating the surface of an interface Γ and \mathbf{n} is the interface unit vector.

The literature presents some option for δ , as shown in Table (3.1), where F is the volume fraction.

Table 3.1: Some δ approximations presented in the literature

Reference	δ function
Brackbill, Kothe and Zemach (1992)	$ \nabla F $
Williams, Kothe and Puckett (1998)	$4F(1 - F)$
Meier, Yadigaroglu and Smith (2002)	$\frac{\nabla\rho(x) \cdot 2\rho(x)}{\rho_l - \rho_g \rho_l + \rho_g}$
Annaland, Deen and Kuipers (2005)	$2F \nabla F $

Since the normal can be approximated by $\mathbf{n} = \frac{\nabla\mathbf{F}}{|\nabla\mathbf{F}|}$, and approximating δ by $|\nabla F|$, the surface tension force is approximated by:

$$\mathbf{F}_\sigma = \sigma\kappa|\nabla F|\frac{\nabla F}{|\nabla F|} = \sigma\kappa\nabla F \quad (3.32)$$

This corresponds to a dispersion of the surface tension across the transition region, using the gradient of the volume fraction variable F to weight the dispersed volume force from Eq.(3.30b). In implementing this approach in a CFD code, one has to consider the: 1) estimation of curvature κ , in order to prevent parasitic currents; 2) spatial discretization of Eq.(3.32) (BOGER; SCHLOTTKE; MUNZ, 2010).

The difficulty associated with volume of fluid (VOF) methods to accurately estimate the interface curvature from an abruptly varying volume fraction distribution, F , is well known. In a large number of two-phase flow applications, the lack of accuracy in the estimation of curvature may produce unphysical velocities (commonly known as spurious or parasitic currents) at fluid interfaces, which may harm the simulation results (LÓPEZ et al., 2009).

As stated by Francois et al. (2006), the Continuum Surface Force has the propensity to generate unphysical flow (parasitic currents) near the interface when surface tension forces are dominant, that are best illustrated in the limiting case of an inviscid static drop in

equilibrium without gravity, where Laplace's formula applies. These artifacts are a serious problem for the capability of the volume-of-fluid approach, since it is a restriction of the approach rather than a numerical inaccuracy. Under some circumstances, it may result in strong vortices at the interface despite the absence of external forcing, which may lead to catastrophic instability of the interface or even to break-up (GERLACH et al., 2006).

For high density ratio flows, a density scaling of the CSF is said to improve the method's performance (KOTHE et al., 1996):

$$\mathbf{F}_\sigma = \sigma \kappa |\nabla F| \frac{\nabla F}{|\nabla F|} = \sigma \kappa \nabla F \frac{\rho}{[\rho]} \quad (3.33)$$

where ρ is the local density, computed as $\rho = \rho_2 + F(\rho_1 - \rho_2)$ and $[\rho] = \frac{\rho_2 + \rho_1}{2}$.

Concerning the spatial discretization of Eq.(3.32), the variables are stored on a staggered grid arrangement. The scalar variables (F, p, ρ, μ, κ) (and also the interface normal vector \mathbf{n}) are stored at the cell centers, while the velocities and surface tension force are stored at the centers of the cell faces. In order to guarantee an accurate, balanced-force discretization, the surface tension terms have to be calculated at the center of the cell faces. To avoid parasitic currents a local approach, that only uses two cell faces to disperse the jump in pressure, was found to be preferable over bigger stencils (BOGER; SCHLOTTKE; MUNZ, 2010).

The gradient evaluation of F only takes into account direct neighbors of the cell faces, which leads, for the cell face $(i-1/2, j, k)$, to:

$$\nabla F_{i-\frac{1}{2}, j, k} = \frac{F_{(i, j, k)} - F_{(i-1, j, k)}}{\Delta x} \quad (3.34)$$

The curvature κ is interpolated from the cell centers to the cell faces as a weighted average; the procedure for the cell face $(i-1/2, j, k)$ is (RENARDY; RENARDY, 2002):

$$\kappa_{i-\frac{1}{2}, j, k} = \frac{\omega_1 \kappa_{(i-1, j, k)} + \omega_2 \kappa_{(i, j, k)}}{\omega_1 + \omega_2} \quad (3.35)$$

where: $\omega_2 = F_{(i, j, k)} \times [1 - F_{(i, j, k)}]$ and $\omega_1 = F_{(i-1, j, k)} \times [1 - F_{(i-1, j, k)}]$.

On the CSS method, the volumetric force from the CSF is transformed in tension. The capillary force term is presented as a tension tensor \mathbf{T} , tangential to the interface:

$$\mathbf{T} = -\sigma(\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\delta \quad (3.36)$$

where \mathbf{I} is the Kronecker tensor and δ is the dirac delta function. The capillary force is:

$$\mathbf{F} = -\nabla \cdot \mathbf{T} \quad (3.37)$$

The equivalent volumetric force is presented by Lafaurie et al. (1994):

$$\mathbf{F}_\sigma = \sigma \nabla \cdot \left(|\nabla F| \mathbf{I} - \frac{\nabla F \otimes \nabla F}{|\nabla F|} \right) \quad (3.38)$$

Accordingly to Rudman (1997), both the CSF and CSS leads to parasitic currents, predominantly on the transition region between the fluids, which tends to grow with time. In problems in which the surface tension effects are predominant over the viscous tension, the spurious currents can cause oscillations along the interface and eventually destroy or completely deform the interface.

In this work the Continuum Surface Force method from Brackbill, Kothe and Zemach (1992) is used to compute the surface tension force.

3.6 Color function initialization

One of the first steps of the VOF method is the initialization of the color function (F). For a desired surface that we aim to analyze, we must transform it in fractional volumes for each Eulerian computational cell, in a way that $0 \leq F \leq 1$. A poor initialization of the color function will lead to a poor analysis of the curvature and also to spurious currents.

Especially for the analysis of the normal and curvature computation, it is desirable to initialize F with complex geometries, as the sphere shape can hide possible weakness of the methods that are being used.

3.6.1 Initialization with implicit functions

An implicit equation (ϕ) of a surface is an equation of the form $\phi(x, y, z) = 0$. One of the properties of the implicit surface is that it has an efficient check for whether a point is inside or outside the surface (FUNKHOUSER, 2002). For a given position $\mathbf{x} = (x, y, z)$, it is easy to check whether this position is inside ($\phi < 0$), outside ($\phi > 0$), or at the interface ($\phi = 0$).

3.6.1.1 The López et al. (2009) initialization method

The basic idea proposed by López et al. (2009) is to compute the signed distance (ϕ) of the function and the division of the interfacial cell in nsc^3 sub-cells to determine which sub-cell is full or empty of fluid.

First, they compute the signed distance from every cell vertex $\mathbf{x}_{ip} = (x_{ip}, y_{ip}, z_{ip})$ to a given interface. If all the vertices of the cell have positive values (outside the fluid) or negative values (inside the fluid), the volume fraction F in the cell is initialized with 0 or 1, respectively. If the cell contains an interface, it is uniformly divided in nsc^3 sub-cells, and the same procedure to define which sub-cell is empty or full is applied. For the interfacial sub-cells (not all vertices have the same sign of the distance function), a volume of a truncated polyhedron is computed. For the shape of a sphere, they show that this method is second-order accurate.

3.6.1.2 Initialization method with points inside the computational cell

There is one simple way to initialize the color function with the implicit equation. For the computational cells that are completely inside or outside one of the phases, the procedure is the same as the method proposed by López et al. (2009), that is, compute the distance from the implicit equation to all the vertices of the computational cell and verify whether this cell is completely full, empty, or has an interface. For the cells that contain an interface, one can populate it with a number of points (N_p^3), with known coordinates $\mathbf{x}_i = (x_{pi}, y_{pi}, z_{pi})$, and compute the distance (ϕ) of each of these points to the surface. Currently, the cell is populated with equidistant points. The fractional volume in each cell is the sum of the points inside the surface (n_{pi}), divided by the number of points inside the cell (N_p^3).

$$F = \frac{\sum n_{pi}}{N_p^3} \quad (3.39)$$

Once the implicit function is provided, the initialization of a complex shape is straightforward. To improve computational performance, each computational cell is also divided in nsc^3 sub-cells. The vertices of each sub-cell are first checked to identify if it is completely full or empty. For the sub-cells that contain an interface, they are populated with a number of points $\left(\frac{N_p}{nsc}\right)^3$.

3.7 Normal and curvature estimation

The most widely used technique to calculate the normal and subsequently the curvature is through the use of the spatial derivatives of the scalar function (RAESSI; MOSTAGHIMI; BUSSMANN, 2007). Hence, the interface normal vector \mathbf{n} is computed as the gradient of

the color function F :

$$\mathbf{n} = -\frac{\nabla F}{|\nabla F|} \quad (3.40)$$

The curvature is computed by taking the divergence of the interface unit normal vector (the second derivative of the VOF function) (RAESSI; MOSTAGHIMI; BUSSMANN, 2007).

To overcome the problem of inaccuracies due to the numerical differentiation of a discontinuous function, a smoothed color function \tilde{F} is suggested for the computational of the unit normal to the interface (BRACKBILL; KOTHE; ZEMACH, 1992), (ANNALAND; DEEN; KUIPERS, 2005).

In the present work, the methods implemented for the normal vector estimation are: 1) the Parker and Young's method; 2) the 125 Cells discretization; and 3) the Least Squares method.

For the curvature estimation the methods compared are: 1) the 27 Cells discretization; 2) the 125 Cells discretization; 3) the Paraboloid method; 4) the Least Squares method; 5) the Height function method.

Each of these methods is explained in the next sections and references to more detailed information are provided.

3.7.1 Parker and Young's method

This method, commonly referred to as Young's method, was proposed by Parker and Youngs (1992). It is one of the most cited methods concerning the normal computation on the VOF method, using a 9-point stencil in 2-D, and 27-point in 3-D (KOTHE; RIDER, 1994).

The implementation in the present work follows Aulisa et al. (2007), which is an extension to three-dimensions of the two-dimensional implementation presented in Scardovelli and Zaleski (2003). The gradient is computed with a finite difference method. The normal is first evaluated at the eight corners of the central cell (i, j, k) . Following Aulisa et al. (2007), the normal components n_x, n_y, n_z in the vertex of coordinates $x_{i+1/2}, y_{j+1/2}, z_{k+1/2}$ are:

$$n_x = \frac{1}{s_x}(\bar{F}_i - \bar{F}_{i+1}); \quad n_y = \frac{1}{s_y}(\bar{F}_j - \bar{F}_{j+1}); \quad n_z = \frac{1}{s_z}(\bar{F}_k - \bar{F}_{k+1}); \quad (3.41)$$

where $\bar{F}_i = (F_{i,j,k} + F_{i,j+1,k} + F_{i,j,k+1} + F_{i,j+1,k+1})/4$. The same scheme is applied in the other vertices of the eulerian cell, and the cell-centered normal vector is obtained by averaging the eight cell-corner values.

Pilliod and Puckett (2004), Kothe et al. (1996) show that this method is at best first-

order accurate. However, due to its simple implementation and low computational cost, it is generally used at least for a first estimate of the direction of the normals, as is required, for example, in the Height function method.

3.7.2 27 Cells discretization

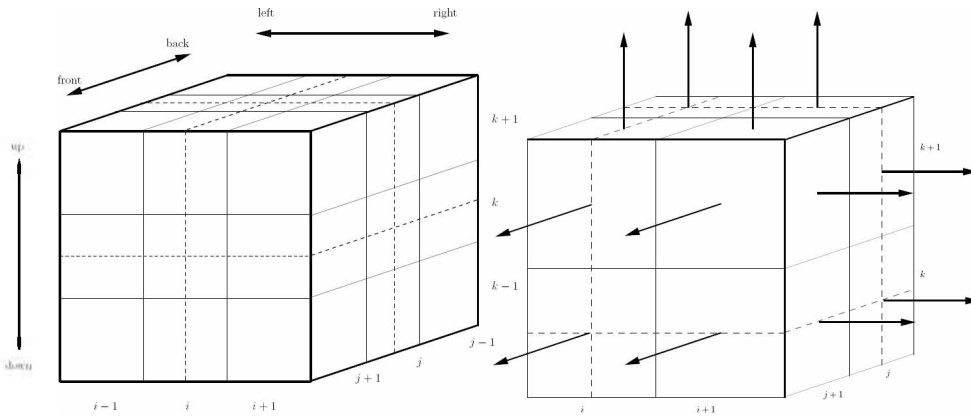


Figure 3.8: A 3D cube (left) and the right upper front octant (right) used for the 27 Cell discretization method. Source: Lam (2009)

This method, presented by Lam (2009), consist of a finite difference discretization of $\mathbf{n} = \frac{\nabla F}{|\nabla F|}$ and $\kappa = \nabla \cdot \mathbf{n}$, as the Young’s method, but evaluates the normals in quadrants, for two-dimensions, and octants, for three-dimensions (illustrated by the broken lines in the left part of Fig. 3.8), instead of determining the normals at the corners of the computational cell.

The octants are $\{\text{left}, \text{right}\} \times \{\text{down}, \text{up}\}, \{\text{back}, \text{front}\}$. For each one of them a direction component (i, j, k) of ∇F is determined by averaging the “arrows” of Fig. 3.8 - right.

The discretization of octant $\{\text{right}, \text{up}, \text{front}\}$ follows bellow. The subscript uf stands for “upper front”, ub for “upper back”, df for “down front” and db for “down back”. In i -direction, the arrows are calculated as:

$$\nabla F_{uf} = \frac{F(i+1, j+1, k+1) - F(i, j+1, k+1)}{\Delta x} \quad (3.42)$$

$$\nabla F_{ub} = \frac{F(i+1, j, k+1) - F(i, j, k+1)}{\Delta x} \quad (3.43)$$

$$\nabla F_{df} = \frac{F(i+1, j+1, k) - F(i, j+1, k)}{\Delta x} \quad (3.44)$$

$$\nabla F_{db} = \frac{F(i+1, j, k) - F(i, j, k)}{\Delta x} \quad (3.45)$$

The average of the arrows are taken to obtain the average value of ∇F in i - direction:

$$\nabla F_x = (\nabla F_{uf} + \nabla F_{ub} + \nabla F_{df} + \nabla F_{db})/4 \quad (3.46)$$

In j -direction:

$$\nabla F_{ru} = \frac{F(i+1, j+1, k+1) - F(i+1, j, k+1)}{\Delta y} \quad (3.47)$$

$$\nabla F_{lu} = \frac{F(i, j+1, k+1) - F(i, j, k+1)}{\Delta y} \quad (3.48)$$

$$\nabla F_{rd} = \frac{F(i+1, j+1, k) - F(i+1, j, k)}{\Delta y} \quad (3.49)$$

$$\nabla F_{ld} = \frac{F(i, j+1, k) - F(i, j, k)}{\Delta y} \quad (3.50)$$

$$\nabla \mathbf{F}_y = (\nabla F_{ru} + \nabla F_{lu} + \nabla F_{rd} + \nabla F_{ld})/4 \quad (3.51)$$

In k -direction:

$$\nabla F_{rf} = \frac{F(i+1, j+1, k+1) - F(i+1, j+1, k)}{\Delta z} \quad (3.52)$$

$$\nabla F_{lf} = \frac{F(i, j+1, k+1) - F(i, j+1, k)}{\Delta z} \quad (3.53)$$

$$\nabla F_{rb} = \frac{F(i+1, j, k+1) - F(i+1, j, k)}{\Delta z} \quad (3.54)$$

$$\nabla F_{lb} = \frac{F(i, j, k+1) - F(i, j, k)}{\Delta z} \quad (3.55)$$

$$\nabla F_z = (\nabla F_{rf} + \nabla F_{lf} + \nabla F_{rb} + \nabla F_{lb})/4 \quad (3.56)$$

With Eq.(3.46,3.51 and 3.56), the gradient at the {right,upper,front} octant is:

$$\nabla F_{ruf} = (\nabla F_x, \nabla F_y, \nabla F_z) \rightarrow |\nabla F_{ruf}| = \sqrt{(\nabla F_x)^2 + (\nabla F_y)^2 + (\nabla F_z)^2} \quad (3.57)$$

The unit normal in this octant ({right,upper,front}) is then:

$$\mathbf{n}_{ruf} = \left(\frac{\nabla F_x}{|\nabla F_{ruf}|}, \frac{\nabla F_y}{|\nabla F_{ruf}|}, \frac{\nabla F_z}{|\nabla F_{ruf}|} \right) \quad (3.58)$$

The direction of the normals for the other 7 octants are obtained in the same way, which leads to a normal component in i , j and k direction for every octant. The average of these (the big arrows in Fig. 3.9) leads to:

$$\kappa = \frac{n_f - n_b}{\Delta x} + \frac{n_r - n_l}{\Delta y} + \frac{n_u - n_d}{\Delta z} \quad (3.59)$$

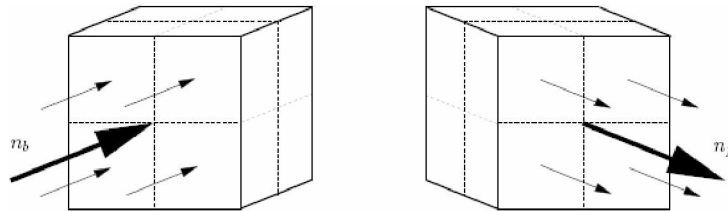


Figure 3.9: The 8 normal components in i - direction, where the big arrows are the average of 4 corresponding small ones. Source: Lam (2009)

3.7.3 125 Cell discretization based on Shirani, Ashgriz and Mostaghimi (2005)

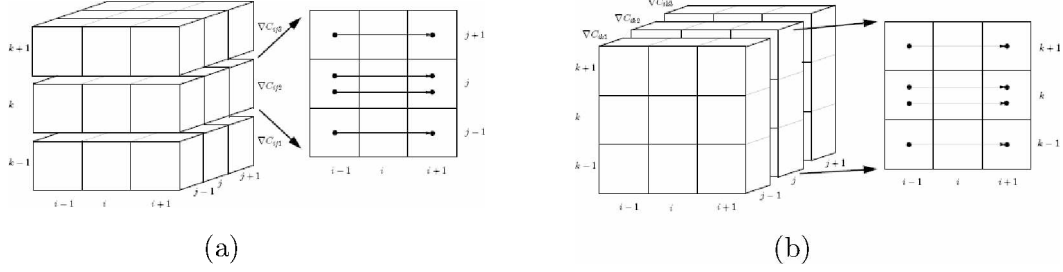


Figure 3.10: 125 Cells discretization in a) i, j -planes and b) i, k - planes. Source: Lam (2009)

The method is also presented by Lam (2009); the author claims it is an extension to three-dimensions of Shirani's unit normal discretization, presented in Shirani, Ashgriz and Mostaghimi (2005), which computed the gradient of F at each point using the values of F in its immediate nine neighboring points for $2D$. In the present work the method is called Shirani's method to follow Lam (2009), although Shirani, Ashgriz and Mostaghimi (2005) does not present it.

Following Lam (2009), first ∇F is approximated in a number of sequential planes. For the i - direction, for example, the planes ij and ik are used (Fig. 3.10). ∇F_{ij} , is the sum of the values computed in 3 sequential planes, $k - 1$ (Eq. 3.60), k (Eq. 3.61) and $k + 1$ (Eq. 3.62) (Fig. 3.10 a). In the same manner, ∇F_{ik} is the sum between the values computed in the planes $j - 1$, j and $j + 1$ (Fig. 3.10 b).

For every plane in Fig. 3.10a, ∇F is approximated as:

$$\nabla F_{ij1} = \frac{F_{(i+1,j-1,k-1)} - F_{(i-1,j-1,k-1)} + 2(F_{(i+1,j,k-1)} - F_{(i-1,j,k-1)}) + F_{(i+1,j+1,k-1)} - F_{(i-1,j+1,k-1)}}{\Delta x} \quad (3.60)$$

$$\nabla F_{ij2} = \frac{F_{(i+1,j-1,k)} - F_{(i-1,j-1,k)} + 2(F_{(i+1,j,k)} - F_{(i-1,j,k)}) + F_{(i+1,j+1,k)} - F_{(i-1,j+1,k)}}{\Delta x} \quad (3.61)$$

$$\nabla F_{ij3} = \frac{F_{(i+1,j-1,k+1)} - F_{(i-1,j-1,k+1)} + 2(F_{(i+1,j,k+1)} - F_{(i-1,j,k+1)}) + F_{(i+1,j+1,k+1)} - F_{(i-1,j+1,k+1)}}{\Delta x} \quad (3.62)$$

The mean is:

$$\nabla F_{ij} = \nabla F_{ij1} + \nabla F_{ij2} + \nabla F_{ij3} \quad (3.63)$$

The same procedure is applied for the planes in Fig. 3.10b, and ∇F is approximated as:

$$\nabla F_{ik1} = \frac{F_{(i+1,j-1,k-1)} - F_{(i-1,j-1,k-1)} + 2(F_{(i+1,j-1,k)} - F_{(i-1,j-1,k)}) + F_{(i+1,j-1,k+1)} - F_{(i-1,j-1,k+1)}}{\Delta x} \quad (3.64)$$

$$\nabla F_{ik2} = \frac{F_{(i+1,j,k-1)} - F_{(i-1,j,k-1)} + 2(F_{(i+1,j,k)} - F_{(i-1,j,k)}) + F_{(i+1,j,k+1)} - F_{(i-1,j,k+1)}}{\Delta x} \quad (3.65)$$

$$\nabla F_{ik3} = \frac{F_{(i+1,j+1,k-1)} - F_{(i-1,j+1,k-1)} + 2(F_{(i+1,j+1,k)} - F_{(i-1,j+1,k)}) + F_{(i+1,j+1,k+1)} - F_{(i-1,j+1,k+1)}}{\Delta x} \quad (3.66)$$

And the mean is:

$$\nabla F_{ik} = \nabla F_{ik1} + \nabla F_{ik2} + \nabla F_{ik3} \quad (3.67)$$

Finally, ∇F_i is computed:

$$\nabla F_i = \nabla F_{ij} + \nabla F_{ik} \quad (3.68)$$

The same procedure is applied for ∇F_j and ∇F_k . The normals are then determined by applying Eq.(3.40) using the averaged gradients.

The curvature κ is determined by the divergence of the unit normal \mathbf{n} using the computed normals directly. Details of the implementations can be seen in Lam (2009), who concludes that, for 2D simulations, this method produces smaller spurious velocities than the height function method till a certain time, when the spurious velocities suddenly increases, and for 3D simulations, the method is symmetric, and for some cases presented

better results than the height function.

3.7.4 The Height function method

To fully understand the height function method, the concept of a Monge Patch is used. A Monge patch is a local surface where $\mathbf{x} : U \rightarrow R^3$ of the form $\mathbf{x}(u, v) = (u, v, h(u, v))$, where U is an open set in R^2 and h is a function that is differentiable in R . Applying the Monge Patch to the first and the second fundamental forms, the mean curvature H can be defined to be (WEISSTEIN, 2012):

$$H = \frac{(1 + h_v^2)h_{uu} - 2h_u h_v h_{uv} + (1 + h_u^2)h_{vv}}{2(1 + h_u^2 + h_v^2)^{\frac{3}{2}}} \quad (3.69)$$

where h_i is the first derivative of h in the i th direction, and h_{ij} is derivative of h_i in the j th direction.

The height function method was first proposed by Torrey et. al. (POPINET, 2009). It is a geometrical technique where a local height is defined as the summation of volume fractions in a direction most normal to an interface (AFKHAMI; BUSSMANN, 2004). Figure 3.11 illustrates the 2-dimensional stencil used to compute the fluid heights.

$$h_{i,j} = \sum_{j-i_s}^{j+i_s} f_{i,j} \Delta y_i \quad (3.70)$$

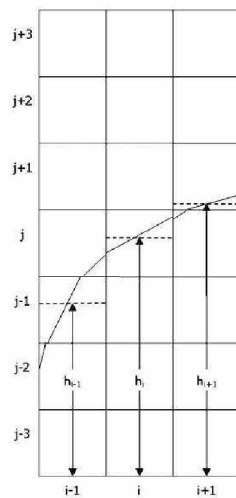


Figure 3.11: The 7x3 fluid stencil used to calculate fluid height. Source: Afkhani and Bussmann (2007)

The curvature of the interface cell (i, j, k) is determined from the derivatives of these heights, as (LÓPEZ et al., 2009; FRANCOIS et al., 2006):

$$\kappa_{i,j,k} = \frac{h_{xx} + h_{yy} + h_{xx}h_y^2 + h_{yy}h_x^2 - 2h_{xy}h_xh_y}{(1 + h_x^2 + h_y^2)^{\frac{3}{2}}} \quad (3.71)$$

where the partial derivatives of h are generally discretized using standard second-order finite-difference scheme.

$$h_{xx} = \frac{h_{x+1,y} - 2h_{x,y} + h_{x-1,y}}{\Delta x_i^2} \quad (3.72)$$

$$h_x = \frac{h_{i+1,j} - h_{i-1,j}}{\Delta x_i^2} \quad (3.73)$$

Comparing Eqs.(3.71) and (3.69),

$$H = \frac{1}{2}\kappa_{i,j,k} \quad (3.74)$$

Hence, the interface curvature from the Height function is twice the mean curvature from the Monge patch.

For the height function integration volume a 7 x 3 stencil in 2D and 7x3x3 stencil in 3D were first employed (FRANCOIS et al., 2006). However, when the radius of curvature of the interface becomes comparable to the mesh size, consistent interface heights cannot be achieved. For these under resolved regions, a smaller stencil, proportional to the radius of curvature should be used instead (POPINET, 2009; FRANCOIS et al., 2006).

López et al. (2009) proposed a smoothing on the first and second partial derivatives of h in order to improve the accuracy in the interface curvature computation which is applied when the angle formed between the height function direction and the interface normal vector reaches a certain critical value. However, in the present work the method proposed by Popinet (2009) is used instead, both for the identification of consistent/inconsistent heights and for the computation of the size of the variable stencil.

The standard method can lead to configurations where consistent heights cannot be formed; these inconsistent heights is an important aspect of the Height function method, stated by Popinet (2009); thus, the method is used in conjunction with another curvature computation method.

3.7.5 The Paraboloid method

A general equation for a paraboloid, with the z coordinate axis coincident with the normal at a point (p) of interest, is given by:

$$f(a_i, x, y) = a_0x + a_1x^2 + a_2xy + a_3y + a_4y^2 + a_5 \quad (3.75)$$

The mean curvature H is:

$$H = 4a_1a_4 - a_2^2 \quad (3.76)$$

For a given position on the surface, a paraboloid that represents a surface around this point p of interest is obtained through the minimization of:

$$F(a_i) = \sum_{1 \leq j \leq n} [z'_j - f(a_i, x, y)]^2 \quad (3.77)$$

The procedure for the paraboloid fitting method in the present work follows what was proposed by Popinet (2009), for a PLIC reconstruction scheme in a cartesian mesh. The points to be fitted can be determined from local height or from the barycenter of the reconstructed surface. Although Popinet (2009) uses both approaches, in the present work only the barycenter is used.

The procedure can be summarized as follows:

for each point P

- *Compute \mathbf{n}_p , the normal at point P*
- *Search the neighbours of P (the coordinates of the center of mass \mathbf{x}_g)*
- *Define a coordinate system in which the z coordinate is coincident with $\mathbf{n}_{p_z} = (0, 0, 1)$*
- *Compute the coordinates of the neighbours of P in this new coordinate system*
- *Fit the paraboloid to the neighbours points of P*
- *Compute the curvature with equation (3.76)*

The local coordinates of the barycenter \mathbf{x}_g can be determined by (AULISA et al., 2007):

$$\mathbf{x}_g = \frac{M_{123}}{A} \left[\alpha^2 \mathbf{x}_0 - \sum_{i=1}^3 (F_2(\alpha - n_i s_i) x_i - F_2(\alpha - \alpha_{max} + n_i s_i) x_{3+i}) \right] \quad (3.78)$$

with:

$$A = M_{123} \left[\alpha^2 - \sum_{i=1}^3 F_2(\alpha - n_i s_i) + \sum_{i=1}^3 F_2(\alpha - \alpha_{max} + n_i s_i) \right] \quad (3.79)$$

where s_i are the dimensions of the eulerian cell,

$$F_2(x) = \begin{cases} x^2, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.80)$$

and

$$M_{123} = \frac{\sqrt{n_1^2 + n_2^2 + n_3^2}}{2n_1 n_2 n_3}$$

In Eq.(3.78), the coordinates x_i are determined as:

$$x_i = \frac{1}{3}(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3) \quad (3.81)$$

\mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 are the coordinates of the vertices of each of the triangles generated from the reconstruction method and are summarized in Tab. 3.2.

3.7.6 Least Squares method

Consider writing out a Taylor series approximation from point P to all neighboring cell center nodes, called F , up to second order (DENNER; WACHEM, 2014):

Table 3.2: Coordinates of the vertices of the triangles from the PLIC reconstruction method

–	vertice 1			vertice 2			vertice 3		
Triangle	x	y	z	x	y	z	x	y	z
0	α/n_1	0	0	0	α/n_2	0	0	0	α/n_3
1	α/n_1	0	0	s_1	$\frac{\alpha-n_1s_1}{n_2}$	0	s_1	0	$\frac{\alpha-n_1s_1}{n_3}$
2	$\frac{\alpha-n_2s_2}{n_1}$	s_2	0	0	α/n_2	0	0	s_2	$\frac{\alpha-n_2s_2}{n_3}$
3	$\frac{\alpha-n_3s_3}{n_1}$	0	s_3	0	$\frac{\alpha-n_3s_3}{n_2}$	s_3	0	0	α/n_3
4	$\frac{\alpha-n_2s_2-n_3s_3}{n_1}$	s_2	s_3	0	$\frac{\alpha-n_3s_3}{n_2}$	s_3	0	s_2	$\frac{\alpha-n_2s_2}{n_3}$
5	$\frac{\alpha-n_3s_3}{n_1}$	0	s_3	s_1	$\frac{\alpha-n_1s_1-n_3s_3}{n_2}$	s_3	s_1	0	$\frac{\alpha-n_1s_1}{n_3}$
6	$\frac{\alpha-n_2s_2}{n_1}$	s_2	0	s_1	$\frac{\alpha-n_1s_1}{n_2}$	0	s_1	s_2	$\frac{\alpha-n_1s_1-n_2s_2}{n_3}$

$$\begin{aligned}
\phi(F) &= \phi(P) + \underbrace{\frac{\partial\phi}{\partial x}|_P}_{\text{unknown}}(x_F - x_P) + \underbrace{\frac{\partial\phi}{\partial y}|_P}_{\text{unknown}}(y_F - y_P) + \underbrace{\frac{\partial\phi}{\partial z}|_P}_{\text{unknown}}(z_F - z_P) \\
&+ \underbrace{\frac{\partial^2\phi}{\partial x\partial x}|_P}_{\text{unknown}} \frac{(x_F - x_P)^2}{2} + \underbrace{\frac{\partial^2\phi}{\partial y\partial y}|_P}_{\text{unknown}} \frac{(y_F - y_P)^2}{2} + \underbrace{\frac{\partial^2\phi}{\partial z\partial z}|_P}_{\text{unknown}} \frac{(z_F - z_P)^2}{2} \\
&+ \underbrace{\frac{\partial^2\phi}{\partial x\partial y}|_P}_{\text{unknown}}(x_F - x_P)(y_F - y_P) + \underbrace{\frac{\partial^2\phi}{\partial x\partial z}|_P}_{\text{unknown}}(x_F - x_P)(z_F - z_P) \\
&+ \underbrace{\frac{\partial^2\phi}{\partial y\partial z}|_P}_{\text{unknown}}(y_F - y_P)(z_F - z_P) + 0(\Delta_x^3, \Delta_y^3, \Delta_z^3)
\end{aligned} \tag{3.82}$$

where all the *unknown* terms have an underbrace and should be determined. There are 9 unknowns, so ideally 9 points around the mesh cell P should be used. However, it is also important to have a symmetric stencil of points around cell P . In practice, matching both of these is not possible, and an overdetermined system of equations is obtained.

For each neighbor point F , the distance is determined as:

$$\begin{aligned}
\Delta x^F &= (x_F - x_P) \\
\Delta y^F &= (y_F - y_P) \\
\Delta z^F &= (z_F - z_P)
\end{aligned} \tag{3.83}$$

and

$$\Delta\phi = (\phi_F - \phi_P) \tag{3.84}$$

Defining the matrix:

$$A = \begin{pmatrix} \Delta x^1 & \Delta y^1 & \Delta z^1 & \frac{1}{2} (\Delta x^1)^2 & \frac{1}{2} (\Delta y^1)^2 & \frac{1}{2} (\Delta z^1)^2 & \Delta x^1 \Delta y^1 & \Delta x^1 \Delta z^1 & \Delta y^1 \Delta z^1 \\ \Delta x^i & \Delta y^i & \Delta z^i & \frac{1}{2} (\Delta x^i)^2 & \frac{1}{2} (\Delta y^i)^2 & \frac{1}{2} (\Delta z^i)^2 & \Delta x^i \Delta y^i & \Delta x^i \Delta z^i & \Delta y^i \Delta z^i \\ \Delta x^N & \Delta y^N & \Delta z^N & \frac{1}{2} (\Delta x^N)^2 & \frac{1}{2} (\Delta y^N)^2 & \frac{1}{2} (\Delta z^N)^2 & \Delta x^N \Delta y^N & \Delta x^N \Delta z^N & \Delta y^N \Delta z^N \end{pmatrix} \quad (3.85)$$

Then, Eq.(3.82) can be rewritten in the form of:

$$A \cdot \begin{pmatrix} \frac{\partial \phi}{\partial x} \Big|_P \\ \frac{\partial \phi}{\partial y} \Big|_P \\ \frac{\partial \phi}{\partial z} \Big|_P \\ \frac{\partial^2 \phi}{\partial x^2} \Big|_P \\ \frac{\partial x \partial x}{\partial^2 \phi} \Big|_P \\ \frac{\partial y \partial y}{\partial^2 \phi} \Big|_P \\ \frac{\partial z \partial z}{\partial^2 \phi} \Big|_P \\ \frac{\partial x \partial y}{\partial^2 \phi} \Big|_P \\ \frac{\partial x \partial z}{\partial^2 \phi} \Big|_P \\ \frac{\partial y \partial z}{\partial^2 \phi} \Big|_P \end{pmatrix} = \begin{pmatrix} \Delta \phi^1 \\ \cdot \\ \cdot \\ \Delta \phi^i \\ \cdot \\ \cdot \\ \Delta \phi^N \end{pmatrix} \quad (3.86)$$

Only if $N = 9$ this system can be solvable exactly. In practice, however, more neighbors should be used to determine the derivatives and a method which does an approximate solution needs to be used. For instance, a least squares fit or a method involving calculating the eigenvalues first can determine an approximate solution.

The normal vector components are obtained from the solution of Eq.(3.86), as:

$$n_x = \frac{\partial \phi}{\partial x} \Big|_P \quad (3.87)$$

$$n_y = \frac{\partial \phi}{\partial y} \Big|_P \quad (3.88)$$

$$n_z = \frac{\partial \phi}{\partial z} \Big|_P \quad (3.89)$$

For the curvature computation, once the normal vector have been determined from the solution of Eq.(3.86), a Taylor series approximation from point P to all neighboring cell center nodes, up to second order, lead to:

$$\begin{aligned} \mathbf{n}(F) = & \mathbf{n}(P) + \underbrace{\frac{\partial \mathbf{n}}{\partial x} \Big|_P}_{\text{}} (x_F - x_P) + \underbrace{\frac{\partial \mathbf{n}}{\partial y} \Big|_P}_{\text{}} (y_F - y_P) + \underbrace{\frac{\partial \mathbf{n}}{\partial z} \Big|_P}_{\text{}} (z_F - z_P) \\ & + \underbrace{\frac{\partial^2 \mathbf{n}}{\partial x \partial x} \Big|_P}_{\text{}} \frac{(x_F - x_P)^2}{2} + \underbrace{\frac{\partial^2 \mathbf{n}}{\partial y \partial y} \Big|_P}_{\text{}} \frac{(y_F - y_P)^2}{2} + \underbrace{\frac{\partial^2 \mathbf{n}}{\partial z \partial z} \Big|_P}_{\text{}} \frac{(z_F - z_P)^2}{2} \\ & + \underbrace{\frac{\partial^2 \mathbf{n}}{\partial x \partial y} \Big|_P}_{\text{}} (x_F - x_P) (y_F - y_P) + \underbrace{\frac{\partial^2 \mathbf{n}}{\partial x \partial z} \Big|_P}_{\text{}} (x_F - x_P) (z_F - z_P) \\ & + \underbrace{\frac{\partial^2 \mathbf{n}}{\partial y \partial z} \Big|_P}_{\text{}} (y_F - y_P) (z_F - z_P) + 0(\Delta_x^3, \Delta_y^3, \Delta_z^3) \end{aligned} \quad (3.90)$$

Again, all the unknown terms have an underbrace and should be determined. For each neighbor point F , the distance from the point P of interest is determined with Eq.(3.83) and:

$$\Delta \mathbf{n}^F = \mathbf{n}^F - \mathbf{n}^P \quad (3.91)$$

which leads to:

$$\begin{aligned} \Delta n_x^F &= (n_x^F - n_x^P) \\ \Delta n_y^F &= (n_y^F - n_y^P) \\ \Delta n_z^F &= (n_z^F - n_z^P) \end{aligned} \quad (3.92)$$

And 3 set of linear systems $Ax = B$ need to be solved (one for each normal coordinate). These systems are represented in Eq.(3.93), with each column in the right hand side related to one of the coordinate systems.

$$A \cdot \begin{pmatrix} \frac{\partial \phi}{\partial x} \Big|_P \\ \frac{\partial \phi}{\partial y} \Big|_P \\ \frac{\partial \phi}{\partial z} \Big|_P \\ \frac{\partial^2 \phi}{\partial x^2} \Big|_P \\ \frac{\partial^2 \phi}{\partial y^2} \Big|_P \\ \frac{\partial^2 \phi}{\partial z^2} \Big|_P \\ \frac{\partial^2 \phi}{\partial x \partial y} \Big|_P \\ \frac{\partial^2 \phi}{\partial x \partial z} \Big|_P \\ \frac{\partial^2 \phi}{\partial y \partial z} \Big|_P \end{pmatrix} = \begin{pmatrix} \Delta n_x^1, \Delta n_y^1, \Delta n_z^1 \\ \cdot \\ \cdot \\ \Delta n_x^i, \Delta n_y^i, \Delta n_z^i \\ \cdot \\ \cdot \\ \Delta n_x^N, \Delta n_y^N, \Delta n_z^N \end{pmatrix} \quad (3.93)$$

The curvature is computed as:

$$\kappa = \frac{\partial \phi}{\partial x} \Big|_P^x + \frac{\partial \phi}{\partial y} \Big|_P^y + \frac{\partial \phi}{\partial z} \Big|_P^z \quad (3.94)$$

where the upper indice indicates the system for which the normal was solved (the column in the right hand side).

A geometrical weighting factor (g), determined as:

$$g = \frac{1}{d^n} \quad (3.95)$$

with $n \geq 1$ and $d = \sqrt{(\mathbf{x}_i - \mathbf{x}_p)^2}$ can be applied to the coefficient terms in the matrix A and in the right hand side vector, both for the computation of the normal vector and the curvature. With this geometrical weighting factor, the method is usually referred as Weighted Least Squares (WLS).

3.8 VOF PLIC method

The VOF methods consists of two steps: reconstruction of the interface and advection of the interface. In the Piecewise Linear Interface Calculation (PLIC) reconstruction step, the objective is to find a plane equation that satisfies volume conservation inside each eulerian cell. In the advection step, the volume inside each cell, represented by the intersection between the eulerian computational cell and the plane, is advected to determine the new

volumes in every eulerian cell (WACHEM; SCHOUTEN, 2002).

3.8.1 Reconstruction Step of the VOF PLIC method

The equation of a plane in 3D space is defined with normal vector (perpendicular to the plane) and a known point on the plane. Let the normal vector of a plane be \mathbf{n} and a known point on the plane, P_1 . Also, let any point on this plane be P . We can define a vector connecting from P_1 to P , which is lying on the plane.

$$(\mathbf{p} - \mathbf{p}_1) = (x - x_1, y - y_1, z - z_1) \quad (3.96)$$

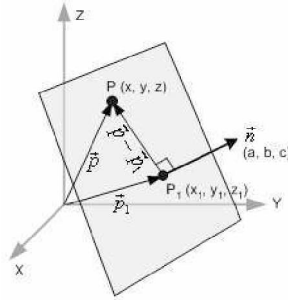


Figure 3.12: A general plane

Since the vector $\mathbf{P} - \mathbf{P}_1 = 0$ and the normal \mathbf{n} are perpendicular to each other, the dot product of two vectors should be 0:

$$\mathbf{n} \cdot (\mathbf{p} - \mathbf{p}_1) = 0 \quad (3.97)$$

This dot product of the normal vector and a vector on the plane becomes the equation of the plane. By calculating the dot product, we get:

$$(a, b, c) \cdot (x - x_1, y - y_1, z - z_1) = 0$$

$$a(x - x_1) + b(y - y_1) + c(z - z_1) = 0$$

$$ax + by + cz - (ax_1 + by_1 + cz_1) = 0 \quad (3.98)$$

If we substitute the constant terms of Eq.(3.98) by $d = ax_1 + by_1 + cz_1$, the plane equation becomes:

$$ax + by + cz = d \quad (3.99)$$

If the normal vector is normalized (unit length), then the constant term of Eq.(3.99), d , becomes the distance from the origin (of the reference computational cell). Therefore, we can find the distance from the origin by dividing the standard plane equation by the length (norm) of the normal vector (normalizing the plane equation). We use the L_2 norm, which is the euclidean norm:

$$norm = \sqrt{a^2 + b^2 + c^2} \quad (3.100)$$

Substituting Eq.(3.100) on Eq.(3.99):

$$\frac{a}{norm}x + \frac{b}{norm}y + \frac{c}{norm}z = \frac{d}{norm}$$

$$m_x x + m_y y + m_z z = \alpha \quad (3.101)$$

3.8.2 The interface reconstruction in the VOF/PLIC method

The volume fraction F is the relation between the volume of interest and the volume of the cell:

$$F_{ijk} = \frac{V_1}{V_{Cell}} \quad (3.102)$$

So, the volume of phase 1 inside the cell is given by:

$$V_1 = F_{ijk} V_{cell} \quad (3.103)$$

In VOF/PLIC methods, the problem is to determine the constants m_x, m_y, m_z, α of Eq.(3.101) so that the cut volume V under the plane is equal to the volume of phase 1 in the cell (i,j,k) (WACHEM; SCHOUTEN, 2002; AULISA et al., 2007).

We consider the following:

1. The volume fraction, the cell volume and the volume of phase 1 inside the cell is known at the beginning;
2. The normal vector, with components n_x, n_y, n_z , are known at the beginning;

Since the normal vector \mathbf{n} and the volume of phase 1 are known, the value of α is computed by enforcing volume conservation. Geometrically this constraint is applied by moving the interface plane along the normal direction, changing in this way the free parameter α , until the volume under the plane is equal to the volume of phase 1 inside the cell (AULISA et al., 2007).

3.8.3 The volume equation

Considering a computational cell with dimensions $s = (s_1, s_2, s_3)$ and normal vector components \mathbf{n} , the volume of fluid inside any computational cell can be calculated by the following algorithm:

Arrange \mathbf{n} , so: $n_1 \leq n_2 \leq n_3$, $\epsilon \simeq 0$

if($n_1 > \epsilon$) then *must perform a three-dimensional analysis*

$$\begin{aligned}
V = & \frac{1}{6.n_1.n_2.n_3} . [\alpha^3 + \\
& -H(\alpha - n_1.s_1).(\alpha - n_1.s_1)^3 + \\
& -H(\alpha - n_2.s_2).(\alpha - n_2.s_2)^3 + \\
& -H(\alpha - n_3.s_3).(\alpha - n_3.s_3)^3 + \\
& +H(\alpha - n_1.s_1 - n_2.s_2).(\alpha - n_1.s_1 - n_2.s_2)^3 + \\
& +H(\alpha - n_1.s_1 - n_3.s_3).(\alpha - n_1.s_1 - n_3.s_3)^3 + \\
& +H(\alpha - n_2.s_2 - n_3.s_3).(\alpha - n_2.s_2 - n_3.s_3)^3
\end{aligned} \tag{3.104}$$

else if($n_2 > \epsilon$) then (*must perform a two-dimensional analysis*)

$$V = \frac{s_1}{2.n_2.n_3} . [\alpha^2 - H(\alpha - n_2.s_2).(\alpha - n_2.s_2)^2 - H(\alpha - n_3.s_3).(\alpha - n_3.s_3)^2] \tag{3.105}$$

else (*must perform a one-dimensional analysis*)

$$V = s_1 \cdot s_2 \cdot \frac{\alpha}{n_3} \quad (3.106)$$

endif

$$H(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.107)$$

Hence, with Eqs.(3.104) to (3.106), the volume inside a computational cell is:

$$Volume = V(\alpha, n, s) \quad (3.108)$$

On Eq.(3.104), the first term represents the total volume under the plane, as illustrated by Fig. 3.13:

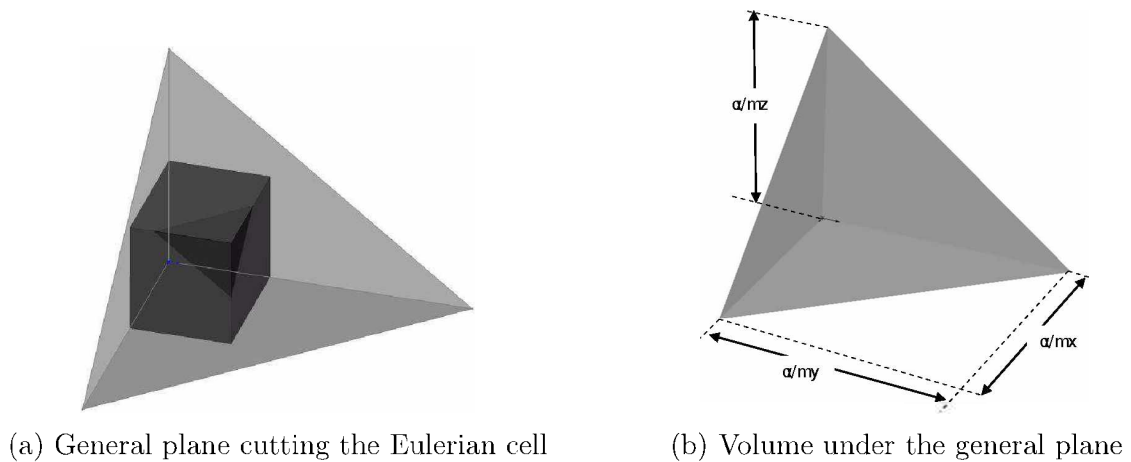


Figure 3.13: Volume under a general plane cutting an Eulerian cell

The subtracting terms on Eq.(3.104) represents the volumes under the plane that are outside the Eulerian computational cell, as illustrated by Fig. (3.14):

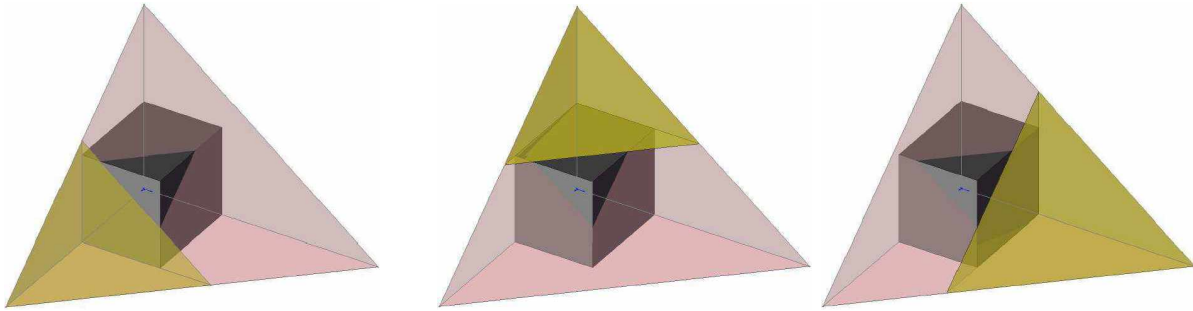


Figure 3.14: Volumes that should be subtracted (yellow) in Eq.(3.104), as they are outside the Eulerian cell

And the adding terms on Eq.(3.104) represents the volumes that should be added back if they were subtracted twice, as illustrated by Fig. (3.15):

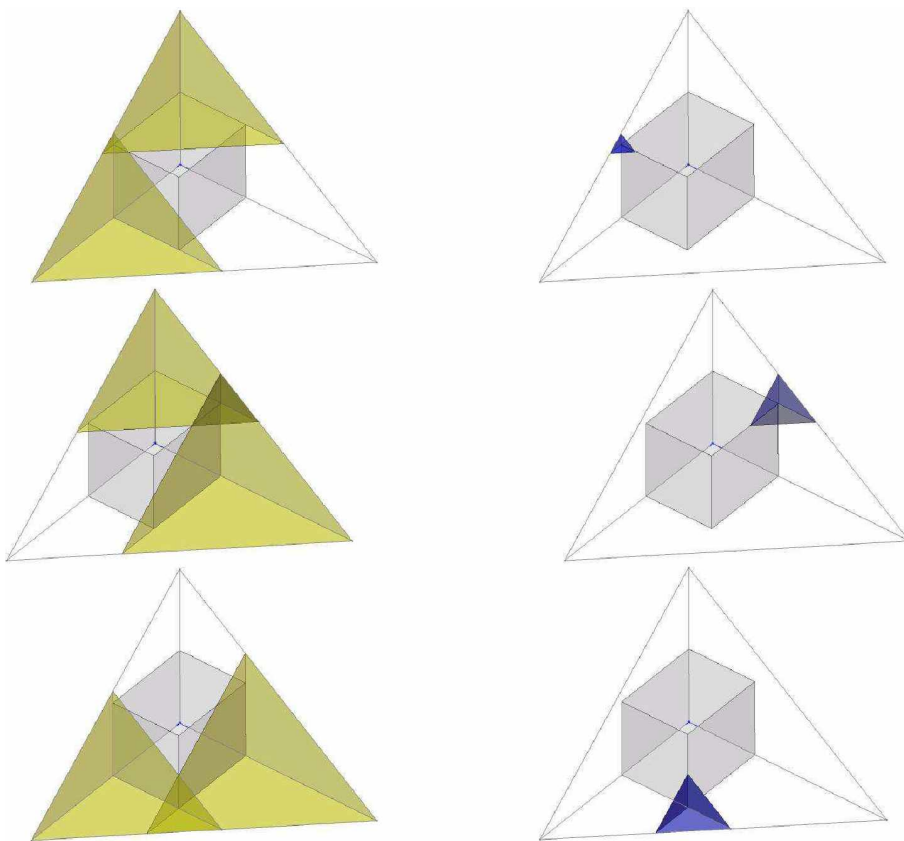


Figure 3.15: Volumes that should be added back (blue) in Eq.(3.104), if they are subtracted twice

Similarly, in Eq.(3.105), the subtracting term represents the volumes under the plane but outside the Eulerian cell, for the case where one of the normal component is zero, as illustrated in Fig. (3.16):

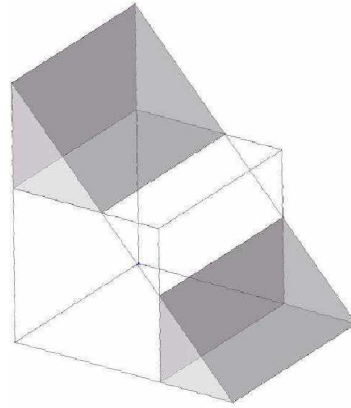


Figure 3.16: Volumes that should be subtracted (gray) in Eq.(3.105), as they are outside the Eulerian cell

Finally, the volume in Eq.(3.106), where only one normal component is different than zero, is illustrated in Fig. 3.17

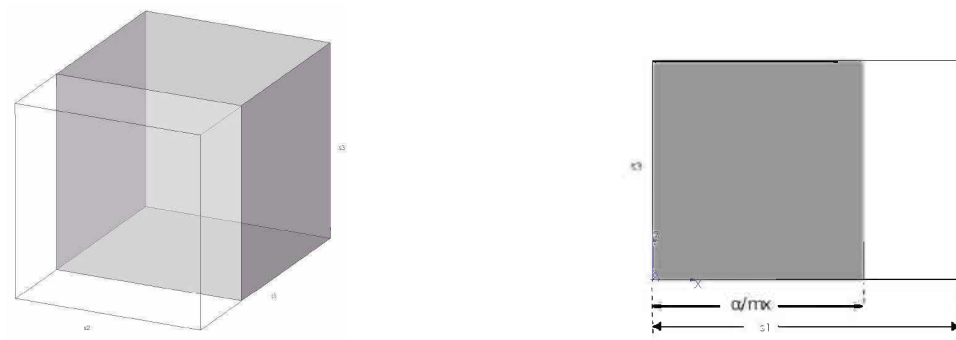


Figure 3.17: Volume represented by Eq.(3.106), considering the normal component at x – $direction$ different than zero

3.8.4 The reconstruction step

For each time step, the Volume (V_0) inside the computational cell is considered to be known. The objective is to determine the parameters (\mathbf{n}, α) that represents the plane inside that cell. In the previous section an equation relating the volume with these parameters was presented. Hence, the problem is to determine the zero of the equation:

$$f = V(\alpha, \mathbf{n}, s) - V_0 = 0 \quad (3.109)$$

As both the normal vector and the size of the cell is known, the only unknown is the α parameter. For its determination, the Newton-Raphson method to find the zero of f is used.

3.9 Lagrangian Propagation of the interface (VOF advection)

The second step of the VOF algorithm is propagation. Once the interface has been reconstructed, its motion by the underlying flow field must be modeled by a suitable advection algorithm (GUEYFFIER et al., 1999). The method described here is presented by Wachem and Schouten (2002) and consist of a Lagrangian Propagation of the Interface.

According to Wachem and Schouten (2002), the Lagrangian propagation of the interface can be best described by how the equation for describing the interface (Eq. 3.101) changes due to the movement of the flow. This change has two possible contributions:

1. The values of α and n_i change due to the fluid flow and lead to a movement of the interface within the computational cell.
2. Due to the movement of the rectangular sides of the volume, the values of s_i may change and the origin to the interface may shift.

3.9.1 First Contribution for the change in the interface equation - movement of the flow

The first contribution is found by updating Eq.(3.101) with new values for n_i and α . These new values are found by integration of the local fluid velocity over the interface. In one direction, this local velocity can be written as:

$$u_i(x_i) = U_L \cdot \left(1 - \frac{x_i}{s_i}\right) + U_R \cdot \frac{x_i}{s_i}$$

↓

$$u_i(x_i) = U_L + \frac{U_R - U_L}{s_i} x_i = Ax + B \tag{3.110}$$

where:

$$A = (U_R - U_L)/s_i, \quad B = U_L$$

In Eq.(3.110), U_L is the fluid velocity where the interface cuts the left face of the computational cell, U_R is the fluid velocity where the interface cuts the right face of the computational cell, x_i is the coordinate into the computational cell, being zero at the left edge and s_i at the right (WACHEM; SCHOUTEN, 2002).

According to Wang et al. (2009) the i th coordinate of each point on the interface can be updated to time $t + \Delta t$ with a second-order Runge-Kutta method:

$$x^{(*)} = [1 + A\Delta t + A^2\Delta t^2]x^{(n)} + AB\Delta t^2 + B\Delta t \quad (3.111)$$

↓

$$x^{(n)} = \frac{x^{(*)} - [AB\Delta t^2 + B\Delta t]}{[1 + A\Delta t + A^2\Delta t^2]} \quad (3.112)$$

After substituting Eq.(3.112) into Eq.(3.101), new values for α and n_i can be found. the equation of the advected interface is given by:

$$n_x^{(*)}x^{(*)} + n_y^{(*)}y^{(*)} + n_z^{(*)}z^{(*)} = \alpha^{(*)} \quad (3.113)$$

where, for the i_{th} direction:

$$n_i^{(*)} = \frac{n_i}{1 + A\Delta t + A^2\Delta t^2} \quad (3.114)$$

$$\alpha^{(*)} = \alpha + \frac{n_i[AB\Delta t^2 + B\Delta t]}{1 + A\Delta t + A^2\Delta t^2} = \alpha + n_i^{(*)}[AB\Delta t^2 + B\Delta t] \quad (3.115)$$

In Eq.(3.113), superscript (*) is used rather than $(n + 1)$ to denote a fractional step, since the interface is just propagated in one direction at this stage (WANG et al., 2009).

This fractional step or operator split method, updates the volume fraction - F - by advecting the interface along one spatial direction at a time. Intermediate F values are calculated during this process, and the final F field is obtained only after advection of the interface along all coordinate directions. Also, the procedure can be made second-order accurate by alternating the advection directions at each time step (GUEYFFIER et al., 1999).

To illustrate this contribution in the change of volume, consider that at the i th direction ($i = 1$), the component of the normal has increased. This leads to a change in the inclination of the interface, as illustrated by Fig. 3.18. As the inclination changes, the parameter α also changes, accordingly to Eq.(3.115).

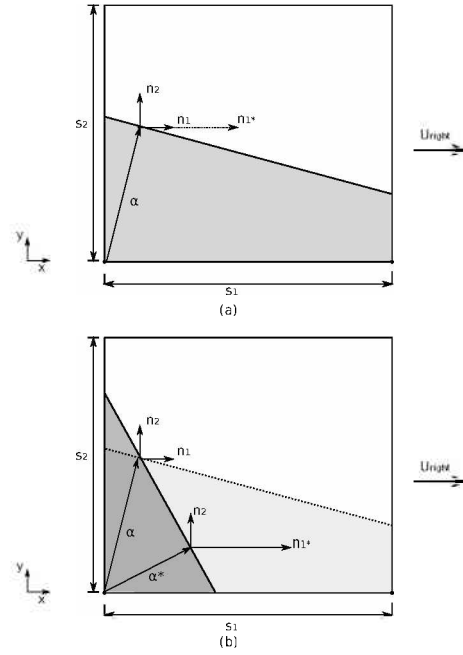


Figure 3.18: (a) Change in the normal component due to the velocity flow field: $n_{1^*} > n_1$ (b) Due to the advection of the normal component, the interface changes its inclination, and consequently the volume fraction inside the Eulerian cell.

3.9.2 Second Contribution for the change in the interface equation - movement of the rectangular sides of the volume

The second contribution occurs both on the left- and right- hand sides of each direction due to protrusion of the interface into neighboring cells. At the left face of the computational cell, the interface protrudes into the left-hand side cell if $U_L < 0$. The interface protrudes into the right-hand side cell if $U_R > 0$ and $\frac{\alpha}{n_i} > s_i$, as the interface has to intersect with the right face of the computational cell (WACHEM; SCHOUTEN, 2002).

For each of these volumes, a local α and a local n_i are calculated, and the volumes are obtained with Eq.(3.108), with parameters computed by Eq.(3.116):

$$Volume = V(\alpha, n, s) \quad (3.116)$$

3.9.2.1 Left face analysis

Consider a computational cell with dimensions s_1, s_2, s_3 , and a movement in the x direction, as illustrated by Fig. 3.19a. In a time change dt , the left face of the volume will translate, according to Eq.(3.117):

$$d_L = |U_L \cdot dt| \quad (3.117)$$

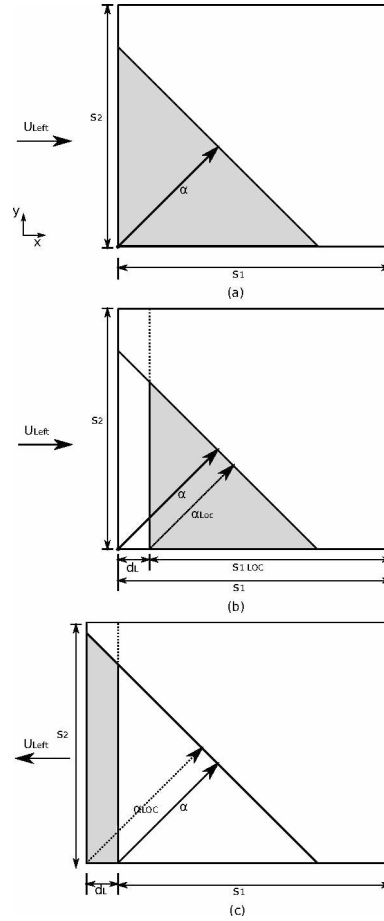


Figure 3.19: left face analysis (a) before movement (b) after movement to right (c) after movement to left

If $U_L > 0$, no volume is transferred to the left neighbor cell, and the volume that remains in the cell is the grey one at Fig. 3.19b.

In this case, it can be noted that the origin of the volume has changed, resulting in a local value of $\alpha \rightarrow \alpha_{LOC}$ that can be easily determined, conforming Eq.(3.118):

$$\frac{\alpha_{LOC}}{n_i} + d_L = \frac{\alpha}{n_i} \rightarrow \alpha_{LOC} = \alpha - n_i \cdot d_L \quad (3.118)$$

In the case illustrated at Fig. 3.19, $n_i = n_1$. Also, the local side (s_{1LOC}), is given by Eq.(3.119):

$$s_{1LOC} = s_1 - d_L \quad (3.119)$$

If $U_L < 0$, the volume that stays in the cell do not change, remaining the grey one at Fig. 3.19a. The volume moved to the left neighbor cell is the grey one at Fig. 3.19c. For this volume, the origin of the volume also change, resulting in a local value of $\alpha \rightarrow \alpha_{LOC}$, determined by Eq.(3.120):

$$\frac{\alpha_{LOC}}{n_i} = \frac{\alpha}{n_i} + d_L \rightarrow \alpha_{LOC} = \alpha + n_i \cdot d_L \quad (3.120)$$

The local side (s_{1LOC}), is given by Eq.(3.121), and is illustrated by Fig. 3.20.

$$s_{1LOC} = d_L \quad (3.121)$$

The grey area in Fig. 3.20a is the cell that must be considered, and in Fig. 3.20b the volume transferred to the left.

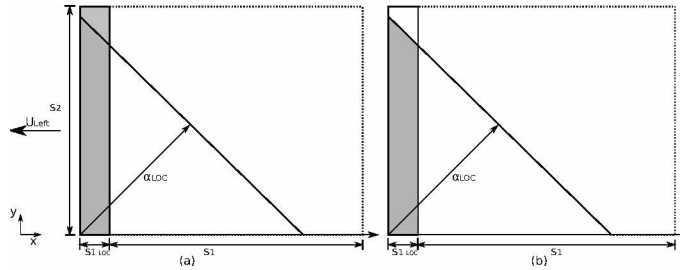


Figure 3.20: (a) Volume of the cell (b) Volume transferred to left

3.9.2.2 Right face analysis

Similar to the left face analysis, in a time change dt , the right face of the volume translate, according to Eq.(3.122):

$$d_R = |U_R \cdot dt| \quad (3.122)$$

Figure 3.21 illustrates the influence of U_R in the volume protruded to the right cell and that remains in the cell.

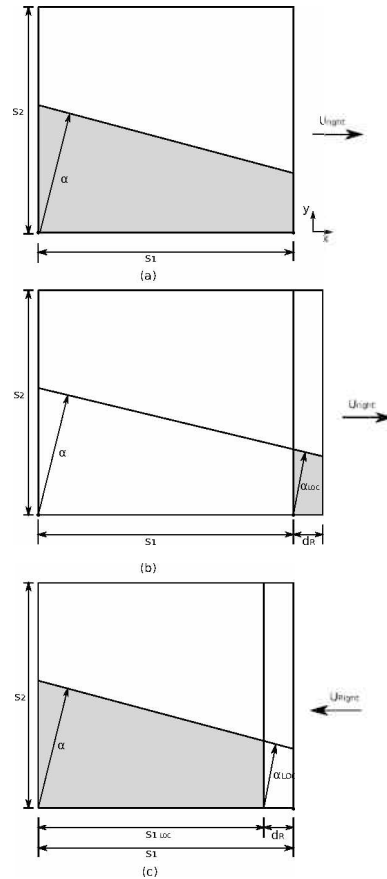


Figure 3.21: Right face analysis (a) before movement (b) after movement to right (c) after movement to left

If $U_R > 0$, the volume that stays in the cell do not change, and is represented by the grey area at Fig. 3.21a. The volume moved to the right neighbor cell is the grey one at Fig. 3.21b, but only if $\alpha/n_i > 0$, otherwise no volume is transferred to the right cell, as can be seen in Fig. 3.22.

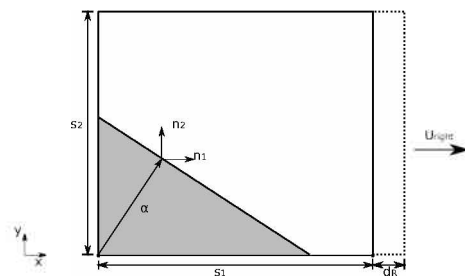


Figure 3.22: $U_R > 0$, but no volume is transferred to right

For the volume moved to right, the side of the computational cell in direction $i = 1$ is given by Eq.(3.123):

$$s_{1LOC} = d_R \quad (3.123)$$

The local α is determined by Eq.(3.124):

$$\frac{\alpha_{LOC}}{n_1} + s_1 = \frac{\alpha}{n_1} \rightarrow \alpha_{LOC} = \alpha - n_1 \cdot s_1 \quad (3.124)$$

If $U_R < 0$, α does not change, so:

$$\alpha_{LOC} = \alpha \quad (3.125)$$

The local side of the computational cell, s_{1LOC} , is given by Eq.(3.126):

$$s_{1LOC} = s_1 - d_R \quad (3.126)$$

3.9.2.3 A combination of the left and right face movement's: $U_L > 0$ and $U_R > 0$

If both velocities are positive, there would be only volume protruded to the right neighbor cell, as shown by Fig. 3.23.

For the volume to right - Fig. 3.23b

From Eq.(3.124): $\alpha_{LOC} = \alpha - n_1 \cdot s_1$

From Eq.(3.123): $s_{1LOC} = d_R$

For the volume that remains in the cell - Fig. 3.23c

From Eq.(3.118): $\alpha_{LOC} = \alpha - n_1 \cdot d_L$

From Eq.(3.119): $s_{1LOC} = s_1 - d_L$

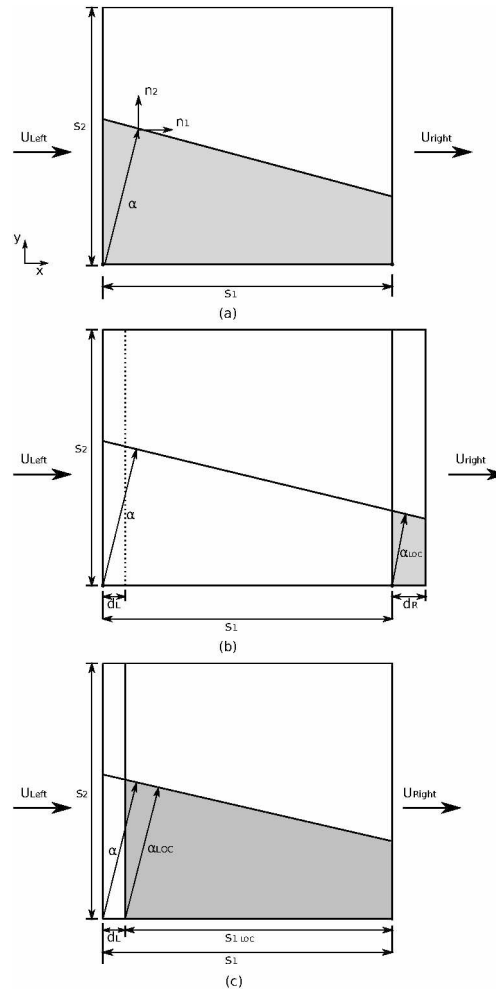


Figure 3.23: $U_L > 0$ and $U_R > 0$ (a) Initial Volume (b) Volume to right (c) Volume that remains in the cell

3.9.2.4 A combination of the left and right face movement's: $U_L < 0$ and $U_R < 0$

If both velocities are negative, there would be only volume protruded to the left neighbor cell, as shown by Fig. 3.24.

For the volume to left - Fig. 3.24b

$$\text{From Eq.(3.120): } \alpha_{LOC} = \alpha + n_1 \cdot d_L$$

$$\text{From Eq.(3.121): } s_{1LOC} = d_L$$

For the volume that remains in the cell - Fig. 3.24c

$$\text{From Eq.(3.125): } \alpha_{LOC} = \alpha$$

$$\text{From Eq.(3.126): } s_{1LOC} = s_1 - d_R$$

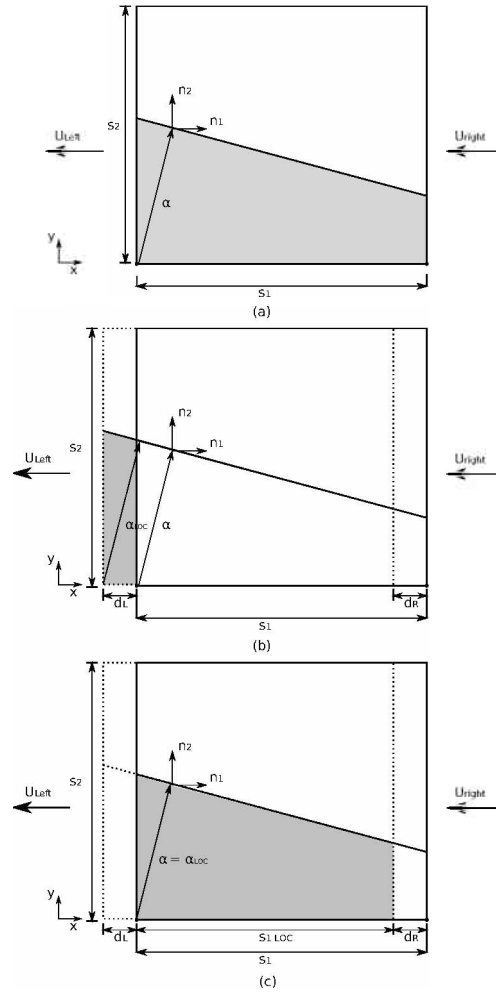


Figure 3.24: $U_L < 0$ and $U_R < 0$ (a)Initial Volume (b)Volume to left (c)Volume that remains in the cell

3.9.2.5 A combination of the left and right face movement's: $U_L > 0$ and $U_R < 0$

In this case, there would be no volume protruded to the left neighbor cell, nor to the right, as shown by Fig. 3.25.

For the volume that remains in the cell - Fig. 3.25b

$$\text{From Eq.(3.118): } \alpha_{LOC} = \alpha - n_1 \cdot d_L$$

$$\text{From Fig. 3.25b: } s_{1LOC} = s_1 - d_R - d_L$$

3.9.3 Performing the lagrangian propagation of the interface

To perform the propagation of the interface, the steps to be followed are:

For each ith direction:

- update n_i with Eq.(3.114)

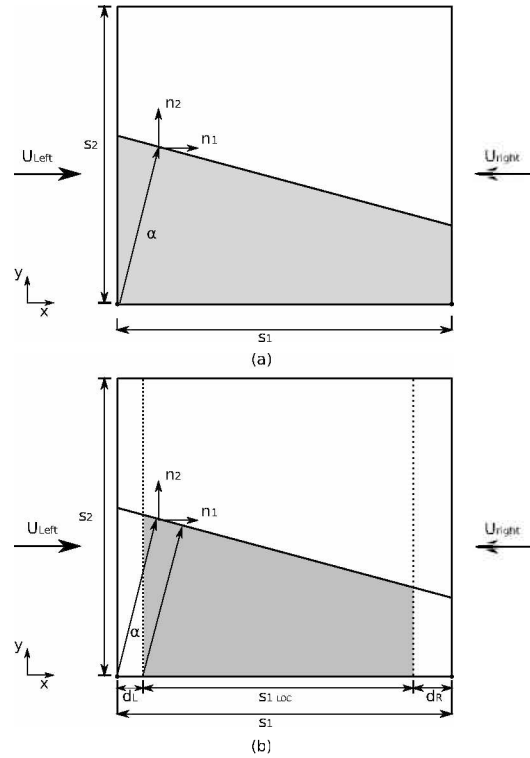


Figure 3.25: $U_L > 0$ and $U_R < 0$ (a)Initial Volume (b)Volume that remains in the cell

- update α with Eq.(3.115)
- For each volume (volume to left, to right, and that stays in the cell), conforming U_L and U_R :

Obtain α_{LOC}

Obtain s_{iLOC}

Calculate the volumes moved to the left, to the right and that remains in the cell, conforming Eq.(3.116) $\rightarrow V(\alpha_{LOC}, n, s_{iLOC})$

- Update F

3.10 Numerical procedure to apply the contact angle

Although it is widely recognized that the dynamic contact angle can deviate significantly from its static value, its influence on the fluid dynamics is not necessarily dominant, for example when gravitational or momentum forces on the liquid are large. When the influence of the dynamic contact angle is negligible a static contact angle is numerically imposed at the contact point at all times (MOURIK, 2002). This model is referred as the static model, and is used in the present work.

Fukai et al. (1995) include the contact angle into the model through the curvature approximation at the contact point. In the Volume-of-fluid approach, Renardy, Renardy and Li (2001) incorporate the contact angle while determining the normal to the interface

at the solid surface. Sikalo, Tropea and Ganic (2005) and Sikalo et al. (2005) include it as a local force in the Volume-of-fluid approach. Spelt (2005), in a Level Set approach, impose it in the re-distance step of the level set function after solving the flow equations.

With the use of the correlations presented to compute the contact angle, Afkhami and Bussmann (2009) propose the use of the Height Function method to account for its influence. First, the Eulerian cells that contains a contact point or are neighbors of a cell with a contact point are identified - Fig. 3.26 - and then the height function in these cells and the ghost ones are computed, with the use of an extrapolation to compute on ghosts - Figs. 3.27 and 3.28.

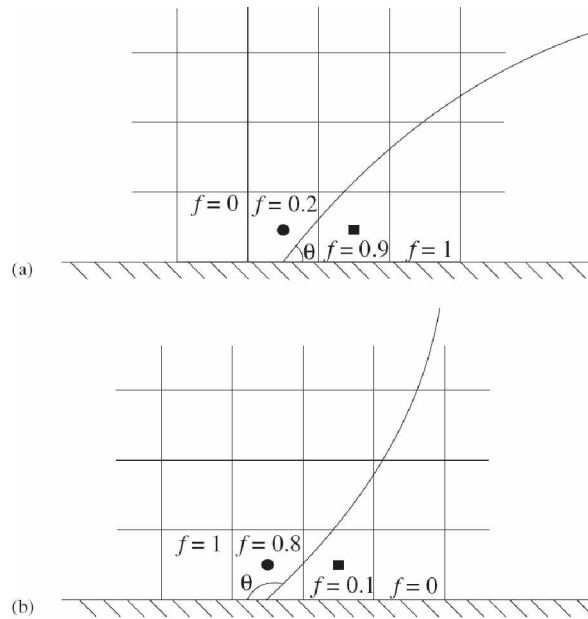


Figure 3.26: A contact point cell (circle) and an adjacent cell (squares): a) $\theta < 90^\circ$ and b) $\theta > 90^\circ$ Identification of contact point, contact cells and adjacent ones. Source: Afkhami and Bussmann (2009)

The local contact angle at which fluid interface meets solid is, strictly, the angle between the normals \mathbf{n}_1 and \mathbf{n}_2 to the two surfaces at a given common point (HUH; SCRIVEN, 1971). As the mean curvature (H) is directly related to the normal vector of the interface, a change in the contact angle will, ultimately, change the mean curvature of the interface at the contact point (CP); in the present work, this idea is used to account for the influence of the contact angle on the model.

Numerically, there is a current position, with the actual (current) contact angle θ_c , mean curvature $H_{current}$ and surface tension force $\mathbf{f}_{\sigma-current} = f(H_{current})$ and a desired condition, with the desired contact angle $\theta_{desired}$, mean curvature $H_{desired}$ and surface tension force $\mathbf{f}_{\sigma-desired} = f(H_{desired})$. At the contact point, the surface tension force is, thus, determined by:

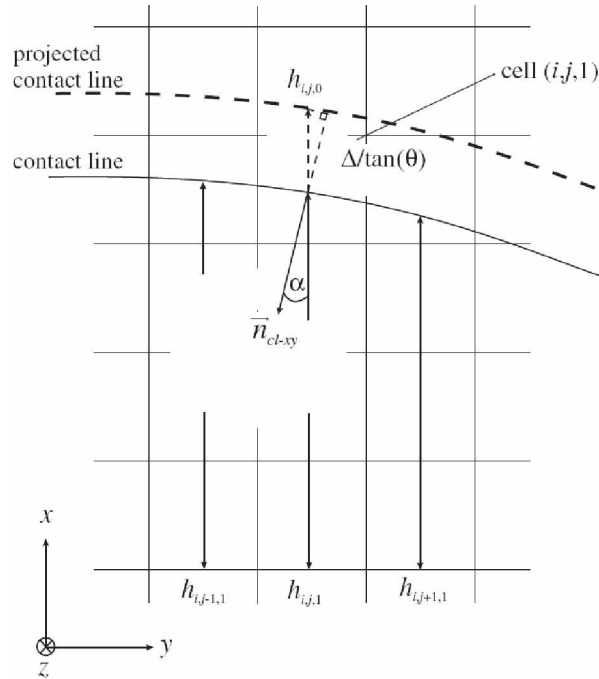


Figure 3.27: The ghost cells heights $h_{i,j-1,0}$, $h_{i,j,0}$ and $h_{i,j+1,0}$ are determined so that the interface orientation at the contact point corresponds to the contact point normal. Source: Afkhami and Bussmann (2009)

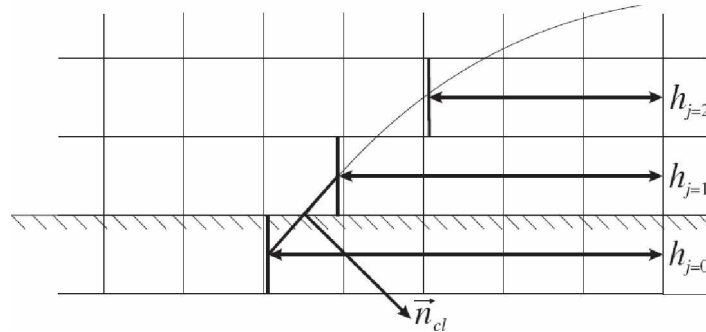


Figure 3.28: The height $h_{j=0}$ in the ghost cell is determined so that the interface orientation at the contact point corresponds to \mathbf{n}_{cl} . Source: Afkhami and Bussmann (2009)

$$\mathbf{f}_\sigma = \mathbf{f}_{\sigma-current} + \Delta(\mathbf{f}_\sigma) \quad (3.127)$$

where $\Delta(\mathbf{f}_\sigma) = f(H_{desired} - H_{current}) = \mathbf{f}_{\sigma-desired} - \mathbf{f}_{\sigma-current}$ and $\Delta(\mathbf{f}_\sigma) \rightarrow 0$ as $H_{current} \rightarrow H_{desired}$.

This formulation is the same as directly imposing the surface tension force obtained from the desired curvature at the contact point (with known contact angle), $\mathbf{f}_\sigma = f(H_{desired})$, since, from Eq.(3.127):

$$\mathbf{f}_\sigma = \mathbf{f}_{\sigma\text{-current}} + (\mathbf{f}_{\sigma\text{-desired}} - \mathbf{f}_{\sigma\text{-current}}) \rightarrow \mathbf{f}_\sigma = \mathbf{f}_{\sigma\text{-desired}} \quad (3.128)$$

The formulation from Eq.(3.127) allows for the coupling between the contact point model and the Immersed Boundary, as explained in the next section. It also allows to analyze if it is possible to speed up the process of the interface achieving its equilibrium condition (where $H_{\text{current}} = H_{\text{desired}}$), by applying a over relaxation factor ($\gamma \geq 1$) on Eq.(3.127).

3.11 Applying the contact angle in the AMR3D Code

The contact angle is imposed through the adjustment of the normal vector \mathbf{n} at the contact point, with both the normal and the curvature at the contact point being computed with the Least Squares approach. With a known contact angle (θ), the objective is to transfer this information to the normal vector at the contact point. As the curvature is obtained from its divergent, this change in the normal vector should lead to a change in the curvature and, consequently, in the surface tension force.

A generic plane, with normal $\mathbf{n}_s = (0, 0, 1)$ is considered, as in Fig. 3.29. At this generic plane, the adjustment of the normal vector is performed with Eq.(3.129):

$$n_z = \frac{\sqrt{n_x^2 + n_y^2}}{\text{tg}(\theta)} \quad (3.129)$$

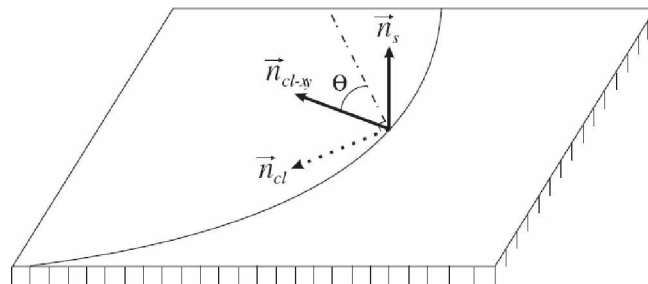


Figure 3.29: $\mathbf{n}_{cl\text{-}xy}$ is the projection of the contact point normal \mathbf{n}_{cl} onto the $x - y$ plane, defined by an outward normal \mathbf{n}_s . Source: Afkhami and Bussmann (2009)

To make this step generic to any wall of the domain, with known normal vector \mathbf{n}_w , a rotation matrix [RM] and a generic plane with normal \mathbf{n}_s is considered. With the use of the rotation matrix [RM], a rotated vector \mathbf{n}_r is obtained. Equation (3.129) is then applied to this rotated vector, and then the reverse matrix (RM^T) is applied to \mathbf{n}_r to obtain the desired normal vector \mathbf{n} .

This procedure, detailed below, allows for the use of Eq.(3.129) to adjust the normal vector at the contact point for any wall with normal direction \mathbf{n}_w .

- Define a generic plane with normal vector $\mathbf{n}_s = (0, 0, 1)$;
- With the known wall normal (\mathbf{n}_w) and the generic plane, build a rotation matrix [RM] at the point:

$$\gamma = \text{acos}[nw(3)] \quad (3.130)$$

$$\mathbf{r}_a = \mathbf{n}_w \wedge \mathbf{n}_s \quad (3.131)$$

$$c = \cos(\gamma) \quad (3.132)$$

$$c1 = 1 - c \quad (3.133)$$

$$s = \sin(\gamma) \quad (3.134)$$

$$RM = \begin{pmatrix} ra(1)^2.c1 + c & ra(1).ra(2).c1 - ra(3).s & ra(1).ra(3).c1 + ra(2).s \\ ra(2).ra(1).c1 + ra(3).s & ra(2)^2.c1 + c & ra(2).ra(3).c1 - ra(1).s \\ ra(3).ra(1).c1 - ra(2).s & ra(3).ra(2).c1 + ra(2).s & ra(3)^2.c1 + c \end{pmatrix} \quad (3.135)$$

- Compute the rotated normal $\mathbf{n}_r = [RM].\mathbf{n}$
- Apply the desired contact angle (θ) to the rotated normal vector: $nr_z = \frac{\sqrt{(nr_x^2 + nr_y^2)}}{\tan(\theta)}$
- Return the rotated vector to the original coordinate axis: $\mathbf{n} = [RM]^T.\mathbf{n}_r$

3.11.1 Determining the direction of displacement of the interface

A necessary step in obtaining the dynamic contact angle is to determine if the interface is advancing or receding. The following procedure is used for this:

- Assembly the rotation matrix [RM] (3x3), with the vector $\mathbf{n}_s = (0, 0, 1)$ and the normal wall vector (\mathbf{n}_w) at the point.
- Rotate the normal and velocity vector:
 - * $\mathbf{r}_n = [RM]\mathbf{n}$
 - * $\mathbf{r}_U = [RM]\mathbf{u}$
- Compute the angle between the rotated normal and velocity vectors: $\gamma = \text{acos} \left(\frac{\mathbf{r}_n \cdot \mathbf{r}_U}{|\mathbf{r}_n \cdot \mathbf{r}_U|} \right)$

* If $\gamma \geq 90$ the interface is advancing (the velocity and the normal vectors are in opposite direction), otherwise the interface is receding.

- At this rotated axis system, the velocity of the contact point is computed as: $\mathbf{U}_c = \sqrt{rU_x^2 + rU_y^2}$
- The dynamic contact angle θ_D is then computed at this rotated coordinate axis system.

3.12 Changes in the Least Squares system to account for the contact point

Two changes are proposed in the solution of the Least Squares system at the contact point:

- since the normals at the wall are known, its values can be used directly in the solution of the system for the normal vector and the curvature;
- use only values inside the domain (do not use values from ghost cells).

3.12.1 Proposed change 1 - Use the values of the known normals at the walls

For the normal vector, considering a Taylor series around point P of interest and a neighbor point w1:

$$\begin{aligned} \phi(P) = \phi(w1) + \frac{\partial\phi}{\partial x}|_{w1}(x_P - x_{w1}) + \frac{\partial\phi}{\partial y}|_{w1}(y_P - y_{w1}) + \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial x}|_{w1}(x_P - x_{w1})^2 + \frac{1}{2} \frac{\partial^2\phi}{\partial y\partial y}|_{w1}(y_P - y_{w1}) + \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial y}|_{w1}(x_P - x_{w1})(y_P - y_{w1}) \end{aligned} \quad (3.136)$$

Isolating w1:

$$\begin{aligned} \phi(w1) = \phi(P) - \frac{\partial\phi}{\partial x}|_{w1}(x_P - x_{w1}) - \frac{\partial\phi}{\partial y}|_{w1}(y_P - y_{w1}) - \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial x}|_{w1}(x_P - x_{w1})^2 - \frac{1}{2} \frac{\partial^2\phi}{\partial y\partial y}|_{w1}(y_P - y_{w1}) - \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial y}|_{w1}(x_P - x_{w1})(y_P - y_{w1}) \end{aligned} \quad (3.137)$$

Disconsidering the higher order terms:

$$\phi(w1) = \phi(P) - nwx_{w1}(x_P - x_{w1}) - nwy_{w1}(y_P - y_{w1}) \quad (3.138)$$

Now, considering a Taylor Series around the desired point P:

$$\begin{aligned} \phi(w_1) = \phi(P) + \frac{\partial\phi}{\partial x}\Big|_P(x_{w_1} - x_P) + \frac{\partial\phi}{\partial y}\Big|_P(y_{w_1} - y_P) + \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial x}\Big|_P(x_{w_1} - x_P)^2 + \frac{1}{2} \frac{\partial^2\phi}{\partial y\partial y}\Big|_P(y_{w_1} - y_P)^2 + \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial y}\Big|_P(x_{w_1} - x_P)(y_{w_1} - y_P) \end{aligned} \quad (3.139)$$

The RHS is then:

$$\begin{aligned} \phi(P) + \frac{\partial\phi}{\partial x}\Big|_P(x_{w_1} - x_P) + \frac{\partial\phi}{\partial y}\Big|_P(y_{w_1} - y_P) + \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial x}\Big|_P(x_{w_1} - x_P)^2 + \frac{1}{2} \frac{\partial^2\phi}{\partial y\partial y}\Big|_P(y_{w_1} - y_P)^2 + \\ \frac{1}{2} \frac{\partial^2\phi}{\partial x\partial y}\Big|_P(x_{w_1} - x_P)(y_{w_1} - y_P) = \\ \phi(P) \underbrace{-nwx_{w_1}(x_P - x_{w_1}) - nwy_{w_1}(y_P - y_{w_1})} \end{aligned} \quad (3.140)$$

This leads to:

$$\frac{\partial\phi}{\partial x}\Big|_P(x_{w_1} - x_P) + \frac{\partial\phi}{\partial y}\Big|_P(y_{w_1} - y_P) + \dots = \underbrace{-nwx_{w_1}(x_P - x_{w_1}) - nwy_{w_1}(y_P - y_{w_1})}$$

For the curvature:

$$n_{Fx} = n_{Px} + \frac{\partial n}{\partial x}\Big|_P(x_F - x_P) \longrightarrow n_{Wx} = n_{Px} + \frac{\partial n}{\partial x}\Big|_P(x_W - x_P) \quad (3.141)$$

$$\frac{\partial n}{\partial x}\Big|_P(x_W - x_P) = n_{Wx} - n_{Px} \quad (3.142)$$

3.12.2 Proposed change 2 - Use only internal cells information

The second proposed change consist in using only computational cells inside the domain to evaluate the normal vector and the curvature. That is, no information from ghost cells

(considered here as outside cells), are used. For this procedure, a loop over the current Eulerian cell and its neighbors is performed and the computational cells are marked either as: (0) inside cells; (1) wall cells; (2) outside cells.

This procedure is particularly important when coupling the contact-angle model with the Immersed Boundary method. The Eulerian cells that contains a Lagrangian point are marked as wall cells, and its correspondent normal vectors are determined as an average of all the Lagrangian normal vectors from points inside the Eulerian cell over analysis. A loop over the neighbors of the ‘‘Eulerian wall cells’’ is performed, and they are marked as ‘‘inside’’ or ‘‘outside’’ cells. This is due only to computational costs, since both the normal vector and the curvature computations are local computations that required information of a small region (for the Least Squares a stencil of 2 neighbors for the normal and one neighbor for the curvature). The procedure is, then:

1. Loop over all Lagrangian points and mark the correspondent Eulerian cell as an interface cell;
2. For each Eulerian cell marked as an interface cell, loop over its neighbors;
3. For each neighbor, find the closest Lagrangian point from its center, and obtain the vectors which connects its vertices to this closest Lagrangian point ($\mathbf{V2P}$);
4. Compute the dot product between $\mathbf{V2P}$ and the normal of the Lagrangian point (\mathbf{n}_L). The signal of the dot product shows in which side the vertice is relative to the Lagrangian point. If all signals are positive, the cell is considered to be ‘‘inside’’ the domain. If all signals are negative, the cell is considered to be ‘‘outside’’, which indicates no information of this particular cell will be used to compute the normal vector and the curvature.

3.13 A direct forcing based scheme to account for the contact point force

From Eq.(3.127), the contact point force is divided in two terms, one that accounts for the current interface position ($\mathbf{f}_{\sigma-current}$), and the second that is responsible to move the interface to the desired position ($\mathbf{f}_2 = \Delta(\mathbf{f}_\sigma)$).

In the AMR3D code the first force term enters the right-hand-side of the Navier-Stokes momentum equations as any source term to solve for the velocity field. For the second force term, a direct forcing based scheme is used instead. The velocity at the contact point is adjusted with Eq.(3.143):

$$\mathbf{u} = \mathbf{u} + \frac{\Delta t \mathbf{f}_2}{\alpha_2} \tag{3.143}$$

where α_2 is obtained from the temporal discretization, as indicated by Eq.(3.10).

This formulation is specially useful if one wants to couple the contact point model with the Immersed Boundary Method.

The Immersed Boundary method consists in obtaining a force term \mathbf{f}_i at each Lagrangian point, and spread it to the Eulerian domain to assure no-slip at the wall, which makes it numerical incompatible with a moving contact point. Also, if the resulting surface tension force is computed accounting for both the current curvature of the interface and its desired final position (the two terms on Eq.(3.127)), the force term which comes from the Immersed Boundary method will always be a term that guarantees no-slip at the wall. The contact point will not move.

To assure that the interface will actually move at the contact point, the second term from Eq.(3.127) enters the solution system after the force term from the immersed boundary method and in a similar fashion (correcting the velocity at the contact point). To guarantee the no-penetration condition, only the force terms parallel to the wall should be considered.

The couple between the contact point and the immersed boundary methods in the code is performed as follows:

- Compute the two terms of the surface tension force
- First term goes to RHS of the Navier-Stokes momentum equation, to solve for the velocity field.
- If there is Immersed Boundary, compute its force and adjust the velocity with the direct forcing scheme.
- Adjust the velocity at the contact point with the second term of the surface tension force and the direct forcing scheme, guaranteeing also no-penetration on the wall.
- Loop to solve for the pressure.

CHAPTER IV

RESULTS

The results presented are divided in two parts: part 1 is related to verification tests, which aims to verify the correctness of the code implementation and comparison with literature data; part 2 is related to applications of the implemented code to contact point simulations. They are presented in the following sections:

- Part 1: verification tests:
 - An analysis of the initialization method is presented in section 4.1.1.
 - An analysis of the reconstruction and advection step of the VOF method is presented in section 4.1.2.
 - The normal and curvature computation are presented in sections 4.1.3 and 4.1.4.
 - Simulations for a 3D static drop in equilibrium and an analysis on the pressure jump error is presented in section 4.1.5.
 - Analysis of the parasitic currents with different curvature computation methods is presented in section 4.1.6.
 - Comparison of two-phase flow simulations with both VOF and Front-Tracking, and with different curvature computation method, is presented in section 4.1.7.
- Part 2: contact point analysis and simulation
 - Contact point study with physical property ratios equals to 1 is presented in section 4.2.1.
 - Contact line-driven drop spreading is presented in section 4.2.2
 - Contact point study with physical property ratios equals to 1, coupled with the Immersed Boundary method is presented in section 4.2.3.

4.1 Verification

4.1.1 Analysis of the initialization methods

An analysis of the convergence error and the computational time is performed for the López et al. (2009) and the Points initialization methods. In the equation that follows, $(x, y, z) = (x_p - x_c, y_p - y_c, z_p - z_c)$, which is the position of a point $p (x_p, y_p, z_p)$ minus the center of the surface (x_c, y_c, z_c) .

The implicit equation, in Cartesian coordinates, for an ellipsoid with semi-principal axes of length a , b and c is

$$\phi(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0 \quad (4.1)$$

with exact volume $V_{exact} = \frac{4}{3}\pi abc$.

Equation (4.1) degenerate to the equation of a sphere when all the semi-principal axes are equal to the radius $r = a = b = c$.

A torus radially symmetric about the z -axis is given by

$$\phi(x, y, z) = R - \sqrt{(x^2 + y^2)} + z^2 - r^2 = 0 \quad (4.2)$$

where R is the distance from the center of the tube to the center of the torus, r is the radius of the tube and its exact volume is: $V_{exact} = 2\pi^2 Rr^2$.

To analyze the quality of the initialization method, the exact volume of the implicit surface is compared with the volume from the color function of the computational cells, computed with Eq.(4.3)

$$V_{initialized} = \Sigma(F_i \times V_i) \quad (4.3)$$

where F_i is the color function at the cell i and V_i is the volume of the cell i .

The error of the initialization method is computed with Eq.(4.4)

$$\epsilon = \left| \frac{V_{initialized} - V_{exact}}{V_{exact}} \right| \quad (4.4)$$

All the tests are performed in one processor, with uniform mesh $n_x = n_y = n_z$ ranging from 4 to 128, which result in a mesh with maximum resolution $\Delta = 2, 1, 1/2, 1/4, 1/8, 1/16$.

For the López et al. (2009) method, the interface cell is uniformly divided in nsc^3 sub-cells. For the Points method the interface cell is also uniformly divided in nsc^3 sub-cell, and in each sub-cell the number of points used is np^3 . Table 4.1 presents the parameters for each test case.

Table 4.1: Input parameters for the initialization analysis of a sphere with radius $r = 2[m]$ and $r = 0.25[m]$, an ellipsoid with semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$ and $a = 0.30[m]$, $b = 0.25[m]$ and $c = 0.20[m]$ and a torus with $R = 2.0$ and $r = 1.0$, all centered in a domain of size $\Omega = 8.0^3[m^3]$

Sphere and Ellipsoid			
Test	Method	nsc	np
1	López et al. (2009)	4^3	-
2	López et al. (2009)	8^3	-
3	Points	1^3	100^3
4	Points	10^3	500^3
Torus			
Test	Method	nsc	np
1	López et al. (2009)	4^3	-
2	López et al. (2009)	8^3	-
3	Points	1^3	100^3
4	Points	10^3	50^3

4.1.1.1 Test initialization for a sphere with radius $r = 2.0[m]$ centered in a domain of size $\Omega = 8^3[m^3]$

A drop (sphere) with radius $r = 2.0$ (mean curvature $H = 0.50$), centered in a domain of size $\Omega = [8]^3[m^3]$ is first analyzed. This setup is presented by López et al. (2009) and is also used in Section 4.1.5 for a series of properties analyses proposed by Francois et al. (2006). The input parameters for each initialization method are presented in Table 4.1. The volume fraction initialization error as a function of the mesh resolution is presented in Fig. 4.1 and the computational time for each method is presented in Fig. 4.2. The convergence ratio is presented in Tab. 4.2.

The López et al. (2009) method presents order 4 when $nsc = 4^3$. For $nsc = 8^3$ it starts with order 4, increase to order 6 and for the more refined mesh it drops the order to 0.5. The reason for this is that, as presented by López et al. (2009), if the volume fraction F is lower than ϵ or higher than $1.0 - \epsilon$, with $\epsilon = 10^{-14}$, F is initialized with 0.0 or 1.0, respectively.

The Points method starts with order 3 and 6, for tests 3 and 4 of Table 4.1, respectively. For the more refined mesh the order also decreases, but less than the López et al. (2009) method, being 2.5 and 3.8 for tests 3 and 4, respectively.

The time for the initialization of the color function also increase with order 4 for both methods, as presented in Fig. 4.2.

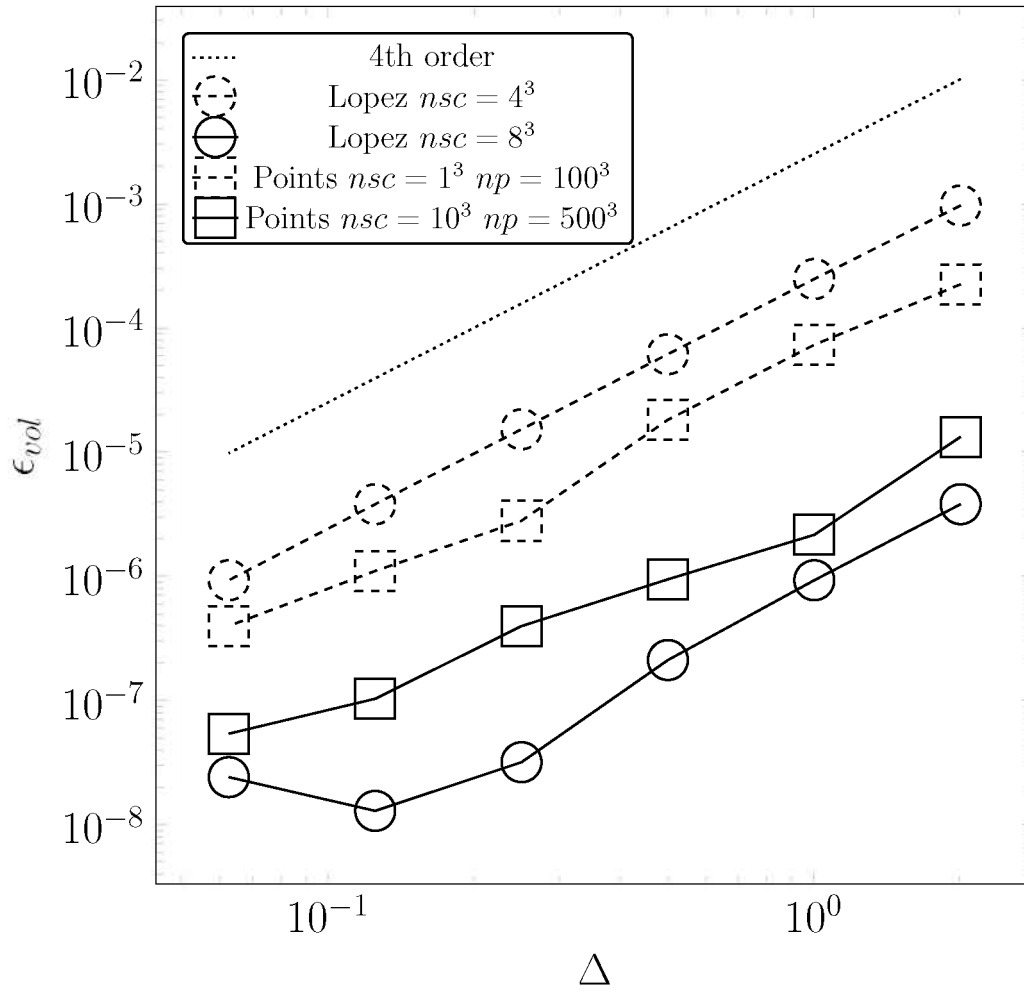


Figure 4.1: Volume fraction initialization errors as a function of grid resolution for a sphere of radius $r = 2$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $n_{sc} = 4^3$ and $n_{sc} = 8^3$ and the Points method with $n_{sc} = 1^3$ $n_p = 100^3$ and $n_{sc} = 10^3$ $n_p = 500^3$.

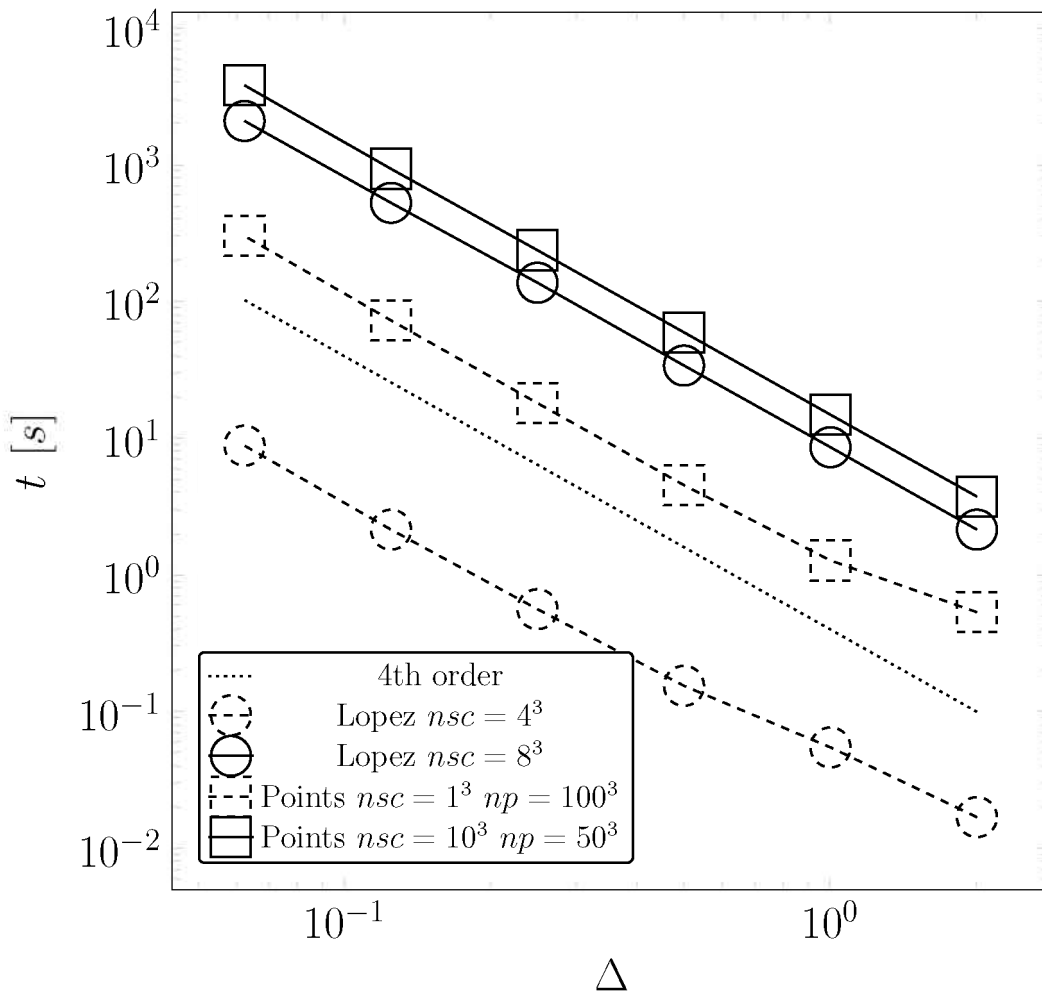


Figure 4.2: Initialization time as a function of grid resolution for a sphere of radius $r = 2$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$.

Table 4.2: Convergence ratio for the error in the color function initialization of a sphere with radius $r = 2.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$

Δ	Lopez		Points	
	$nsc = 4^3$	$nsc = 8^3$	$nsc = 1^3$ $np = 100^3$	$nsc = 10^3$ $np = 500^3$
$2/1$	-	-	-	-
$1/1$	3.91	4.09	3.06	6.12
$1/2$	4.02	4.40	4.03	2.28
$1/4$	4.01	6.63	6.48	2.38
$1/8$	4.02	2.46	2.52	3.83
$1/16$	4.09	0.54	2.78	1.92

Figure 4.3 presents the color function for the sphere interface initialized with the Points method for each mesh resolution tested.

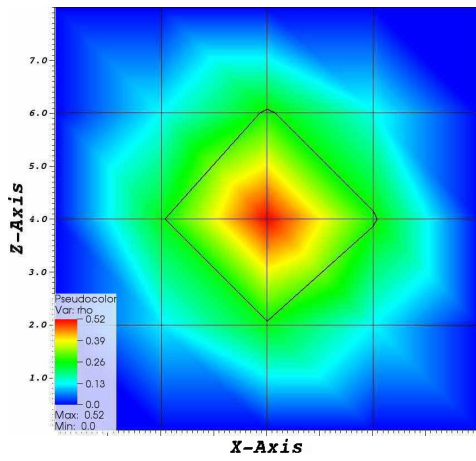
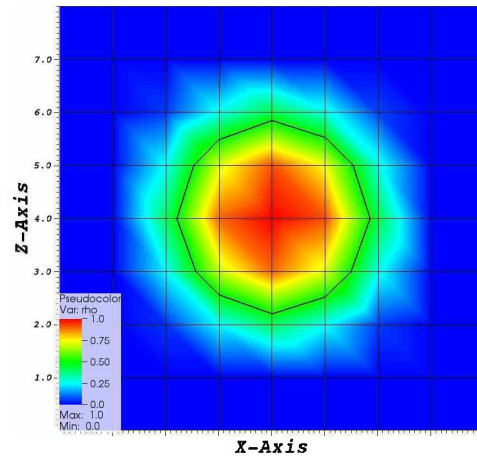
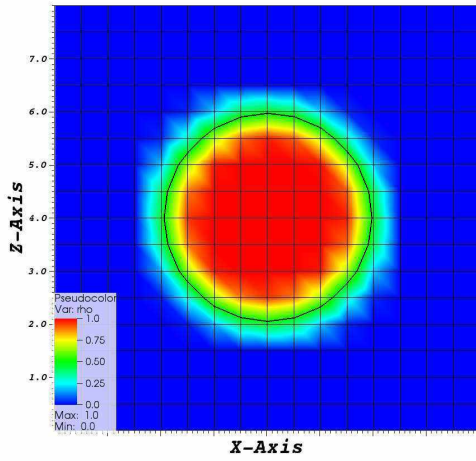
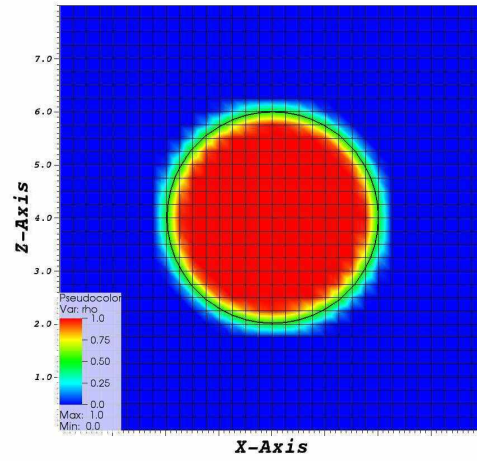
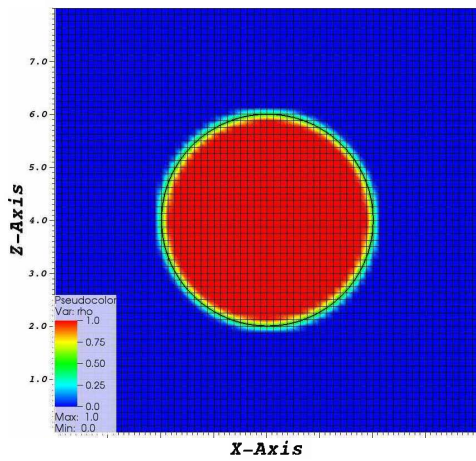
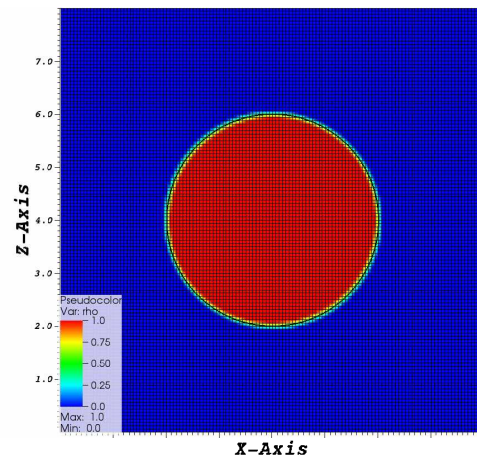
(a) $\Delta = 2$ (b) $\Delta = 1$ (c) $\Delta = 1/2$ (d) $\Delta = 1/4$ (e) $\Delta = 1/8$ (f) $\Delta = 1/16$

Figure 4.3: Color function for the sphere with radius $r = 2[m]$, centered in a domain of size $\Omega = 8^3[m^3]$, initialized in the AMR3D code with the López et al. (2009) method, for maximum mesh resolution ranging from $\Delta = 2$ to $\Delta = 1/16$

4.1.1.2 Test initialization for a sphere with radius $r = 0.25[m]$ centered in a domain of size $\Omega = 1.0^3[m^3]$

A drop (sphere) with radius $r = 0.25$ (mean curvature $H = 4.0$), centered in a domain of size $\Omega = [1]^3[m^3]$ is initialized. This setup is used in Section 4.1.3 for an analysis of the normal and curvature computation with different methods. The input parameters for each initialization method are presented in Table 4.1.

The volume fraction initialization error as a function of the mesh resolution is presented in Fig. 4.4. The convergence ratio for the initialization error is presented in Tab. 4.3.

The computational time increase with order 4, as presented in Fig. 4.5.

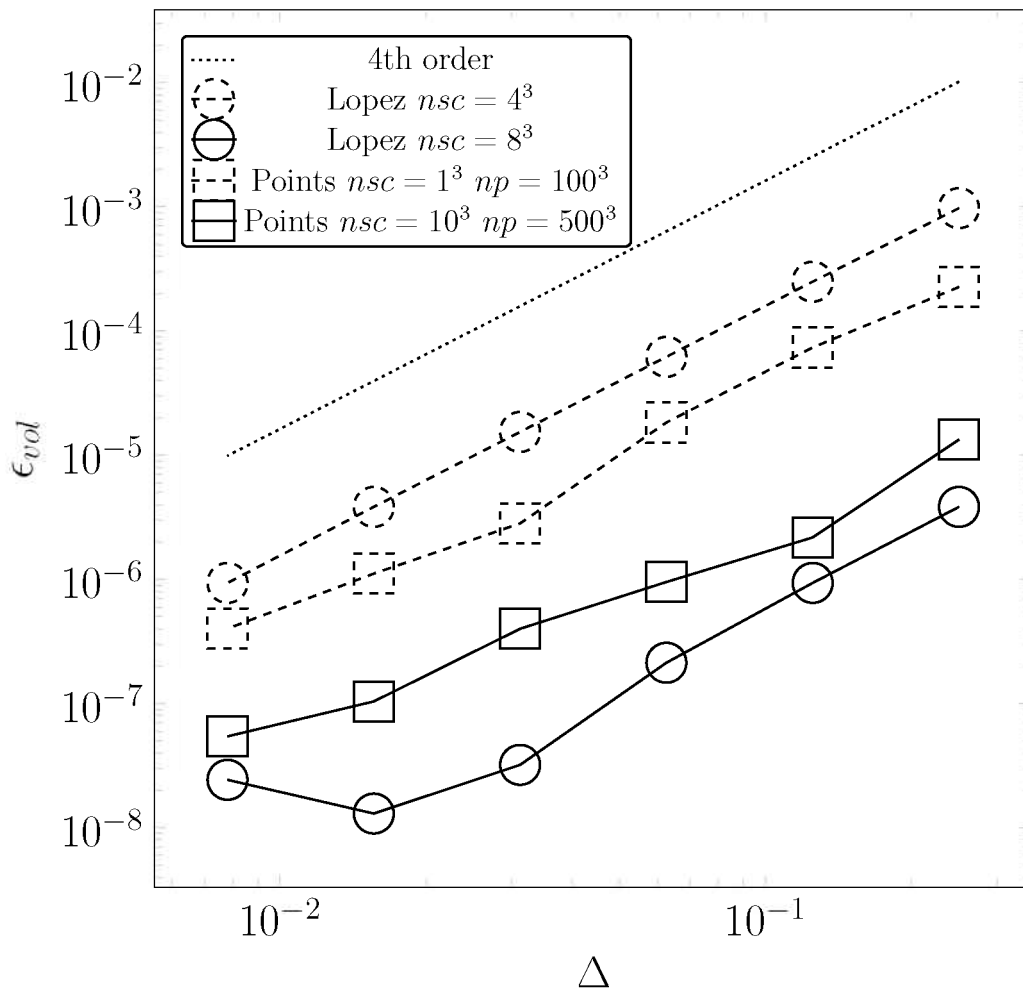


Figure 4.4: Volume fraction initialization errors as a function of grid resolution for a sphere of radius $r = 0.25[m]$, centered in a domain of size $\Omega = 1^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3 np = 100^3$ and $nsc = 10^3 np = 500^3$.

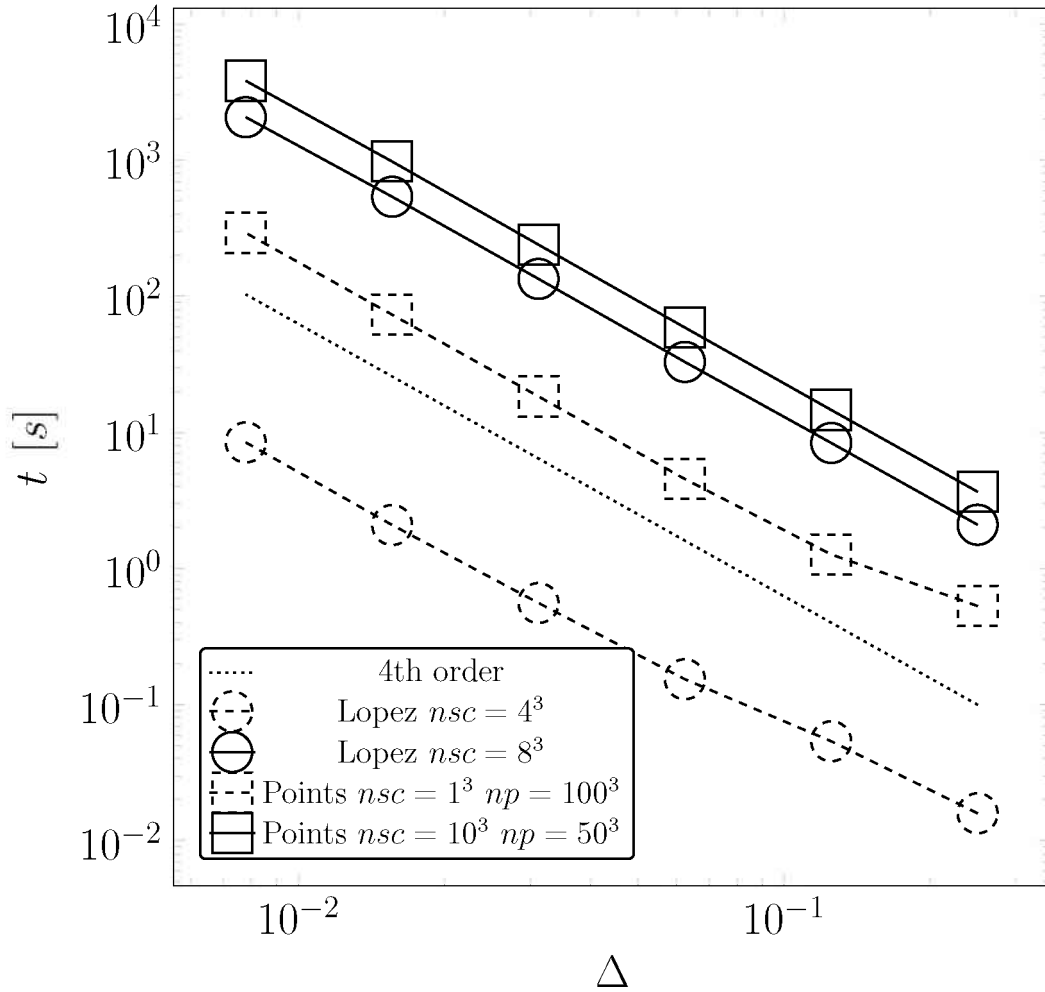


Figure 4.5: Initialization time as a function of grid resolution for a sphere of radius $r = 0.25[m]$, centered in a domain of size $\Omega = 1^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3 np = 100^3$ and $nsc = 10^3 np = 500^3$.

Table 4.3: Convergence ratio for the error in the color function initialization of a sphere with radius $r = 0.25[m]$, centered in a domain of size $\Omega = 1^3[m^3]$

Δ	Lopez		Points	
	$nsc = 4^3$	$nsc = 8^3$	$nsc = 1^3 np = 100^3$	$nsc = 10^3 np = 500^3$
$2/1$	-	-	-	-
$1/1$	3.91	4.09	3.06	6.12
$1/2$	4.02	4.40	4.03	2.28
$1/4$	4.01	6.63	6.48	2.38
$1/8$	4.02	2.46	2.52	3.83
$1/16$	4.09	0.54	2.78	1.92

4.1.1.3 Test initialization for an ellipsoid with semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8.0^3[m^3]$

Based on the test case for the drop (sphere) with radius $r = 2.0[m]$ an ellipsoid with semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$ is used for the initialization analysis. The input parameters are presented in Table 4.1. The volume fraction initialization error as a function of the mesh resolution is presented in Fig. 4.6 and the computational time for each method is presented in Fig. 4.7. The convergence ratio is presented in Tab. 4.4.

As for the sphere initialization, the López et al. (2009) method presents order 4 when $nsc = 4^3$. For $nsc = 8^3$ it starts with order 4, has an increase for $\Delta = 1/8$ and decrease the order to 0.2 when $\Delta = 1/16$. The Points method starts with order 4 for tests 3 and 4 of Table 4.1, respectively. For test case 3, the order decreases for intermediate *Delta* values, and goes to order 4 for $\Delta < 1/8$. For test case 4 the order decreases to 2 for $\Delta < 1/8$.

The time for the initialization of the color function also increase with order 4 for both methods, as presented in Fig. 4.7.

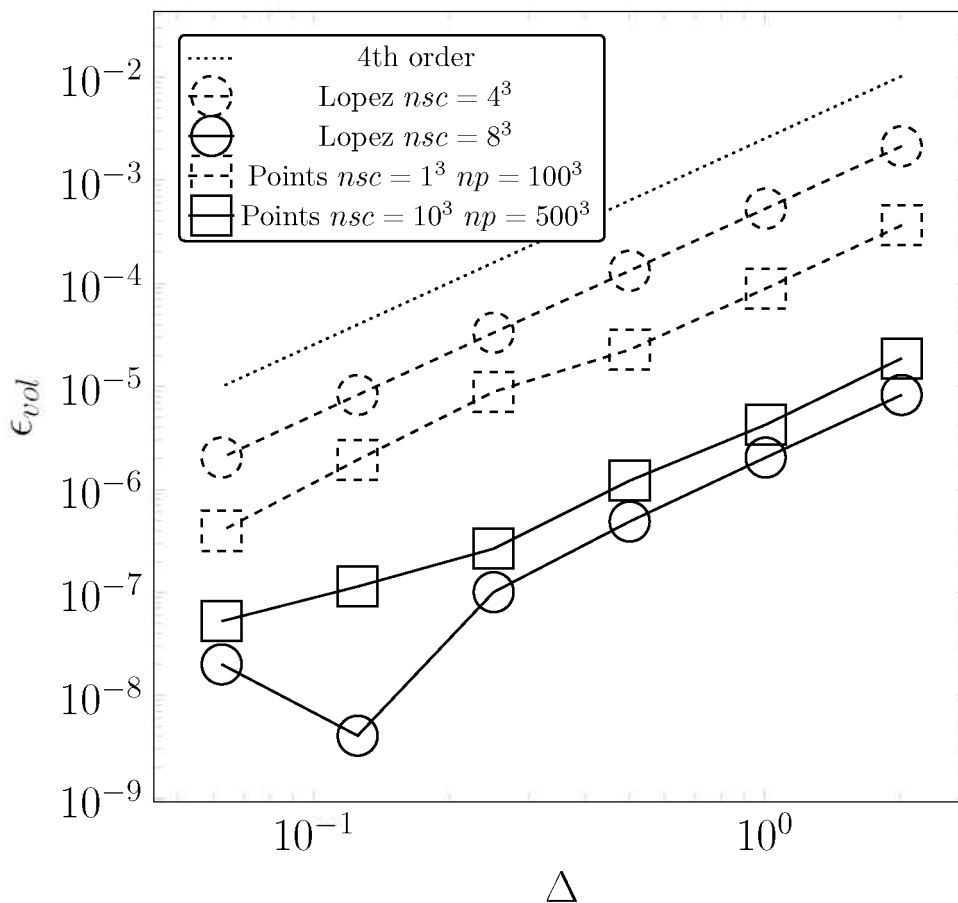


Figure 4.6: Volume fraction initialization errors as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3 np = 100^3$ and $nsc = 10^3 np = 500^3$.

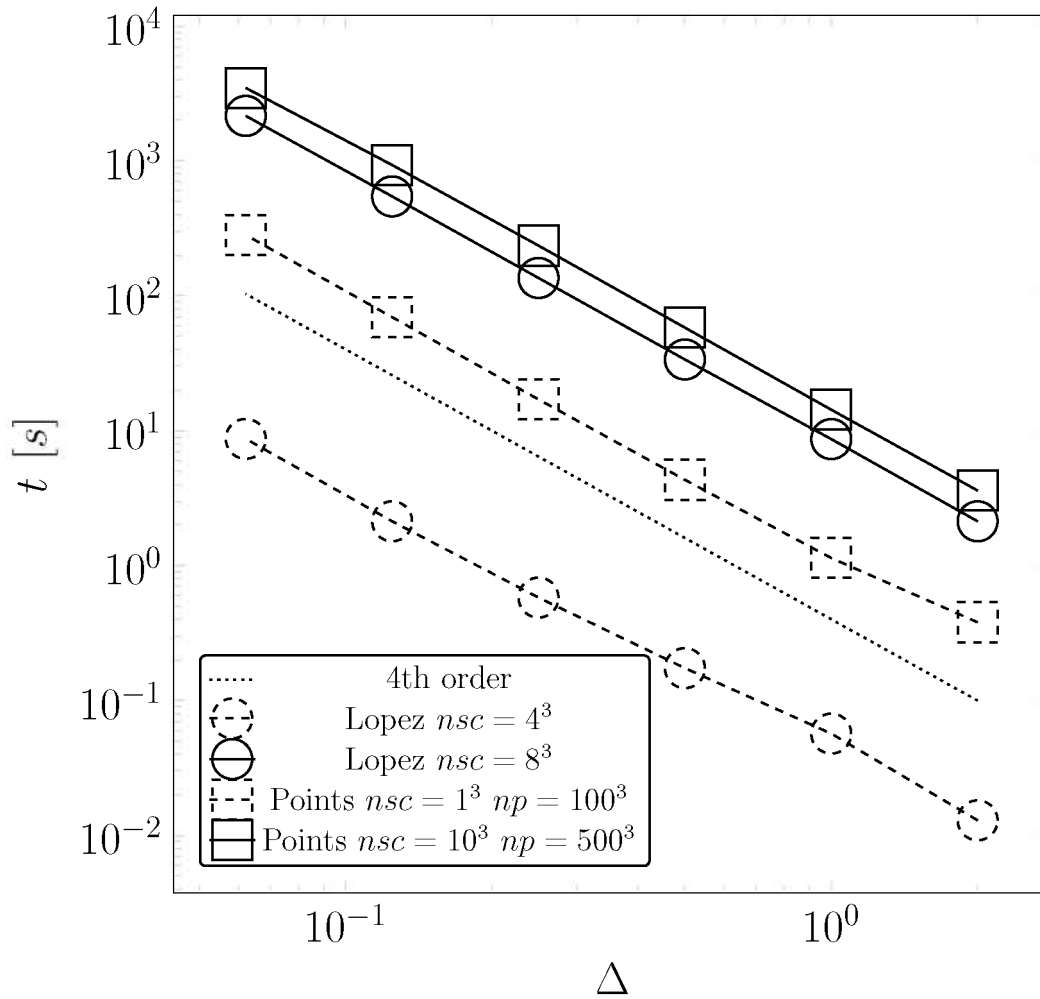


Figure 4.7: Initialization time as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$.

Table 4.4: Convergence ratio for the error in the color function initialization of an ellipsoid with semi-principal axes of length $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$

Δ	Lopez		Points	
	$nsc = 4$	$nsc = 8$	$nsc = 1$ $np = 100^3$	$nsc = 10$ $np = 500^3$
$2/1$	-	-	-	-
$1/1$	4.02	4.04	4.10	4.39
$1/2$	3.99	4.17	3.91	3.48
$1/4$	4.00	4.84	2.56	4.57
$1/8$	4.02	24.61	4.56	2.33
$1/16$	4.04	0.20	4.89	2.17

Figure fig: initialization ellipsoid amr3d presents the color function for the ellipsoid interface initialized with the Points method for each mesh resolution tested.

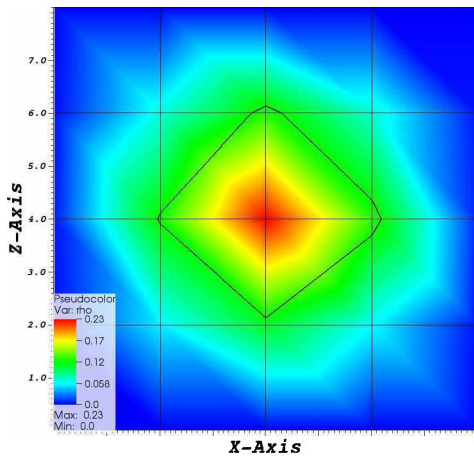
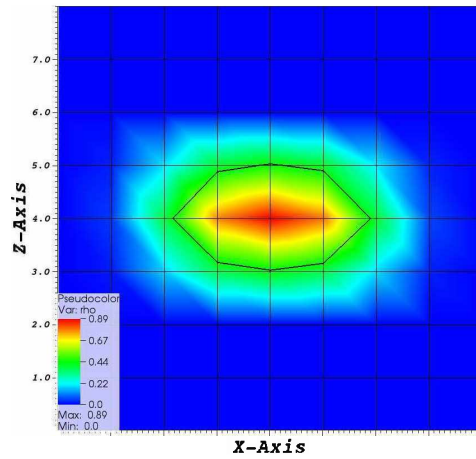
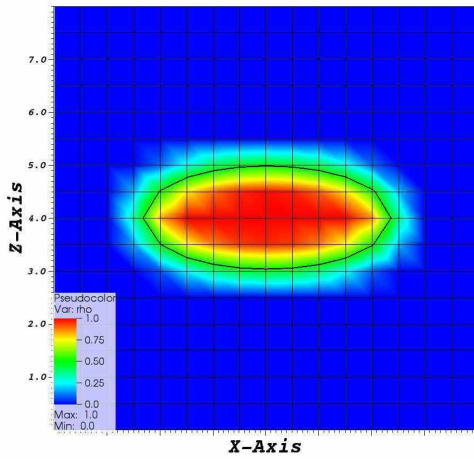
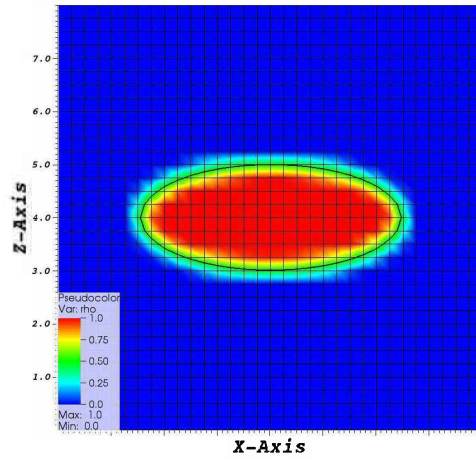
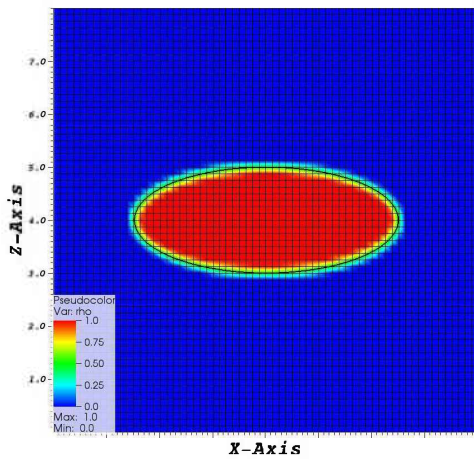
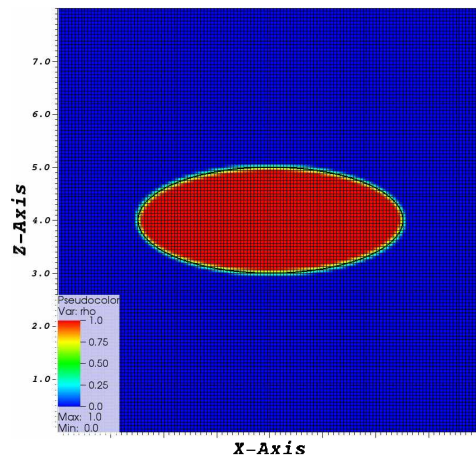
(a) $\Delta = 2$ (b) $\Delta = 1$ (c) $\Delta = 1/2$ (d) $\Delta = 1/4$ (e) $\Delta = 1/8$ (f) $\Delta = 1/16$

Figure 4.8: Color function of the ellipsoid with semi-axes $a = 2.5[m]$, $b = 1.5[m]$ and $c = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$, initialized in the AMR3D code with the Points method, for maximum mesh resolution ranging from $\Delta = 2$ to $\Delta = 1/16$

4.1.1.4 Test initialization for an ellipsoid with semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1.0^3[m^3]$

Based on the test case for the drop (sphere) with radius $r = 0.20[m]$ an ellipsoid with semi-principal axes of length $a = 0.3[m]$, $b = 1.5[m]$ and $c = 1.0[m]$ is used for the initialization analysis. This setup is used in Section 4.1.3 for the normal and curvature analysis.

The input parameters are presented in Table 4.1. The volume fraction initialization error as a function of the mesh resolution is presented in Fig. 4.9. For this interface, the convergence ratio is between order 3 and 4 for both methods, but the magnitude of the error is smaller with the Points method. The convergence ratio is presented in Tab. 4.5.

The computational time increase with order 4, as presented in Fig. 4.10.

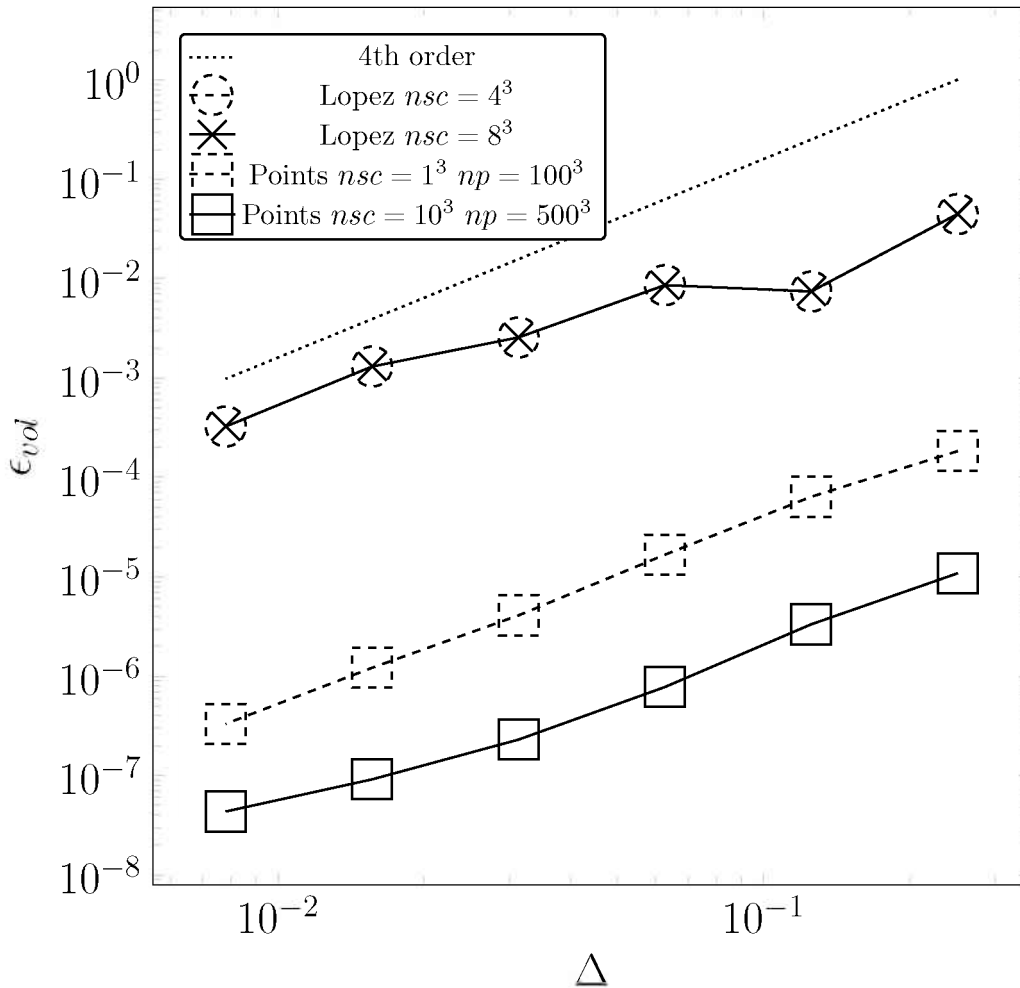


Figure 4.9: Volume fraction initialization errors as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1.0^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3 np = 100^3$ and $nsc = 10^3 np = 500^3$.

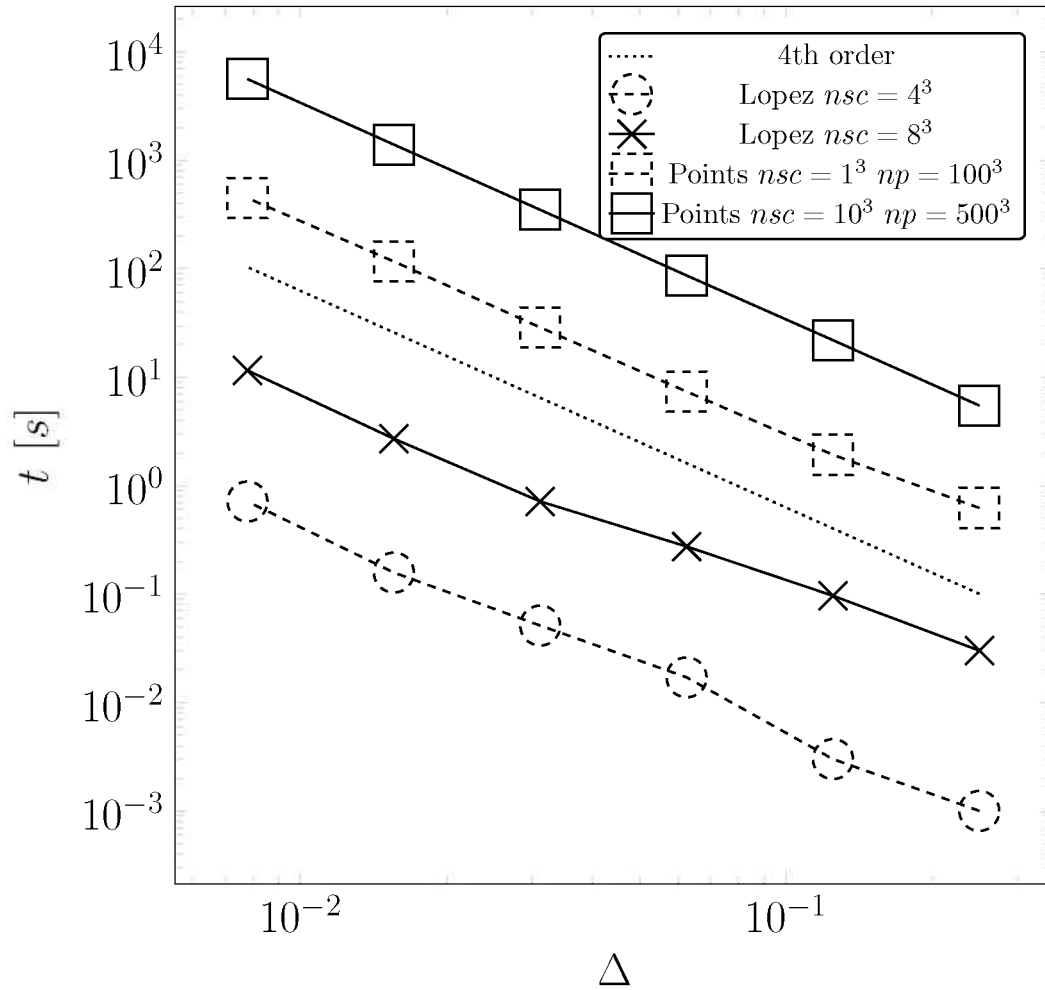


Figure 4.10: Initialization time as a function of grid resolution for an ellipsoid of semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3$ $np = 100^3$ and $nsc = 10^3$ $np = 500^3$.

Table 4.5: Convergence ratio for the error in the color function initialization of an ellipsoid with semi-principal axes of length $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, centered in a domain of size $\Omega = 1.0^3[m^3]$

Δ	Lopez		Points	
	$nsc = 4$	$nsc = 8$	$nsc = 1$ $np = 100^3$	$nsc = 10$ $np = 500^3$
$1/4$	-	-	-	-
$1/8$	3.00	3.20	3.09	4.00
$1/16$	5.67	2.84	3.82	3.94
$1/32$	3.00	2.60	3.90	4.03
$1/64$	3.08	3.78	4.05	3.99
$1/128$	4.52	4.26	3.84	4.02

4.1.1.5 Test initialization for a Torus with parameters $R = 2.0[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = [8]^3[m^3]$

A Torus with parameters $R = 2.0[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = [8]^3[m^3]$ is analyzed. The input parameters for each initialization method are presented in Table 4.1. The volume fraction initialization error as a function of the mesh resolution is presented in Fig. 4.11 and the computational time for each method is presented in Fig. 4.12. The convergence ratio is presented in Tab. 4.6.

The López et al. (2009) method presents order ≈ 1 up to $\Delta = \frac{1}{2}$, both for $nsc = 4^3$ and $nsc = 8^3$. For $\Delta < \frac{1}{2}$ it presents order 3. The Points method present order ≈ 3 for all mesh resolution tested. Both method present order 4 for the initialization time, as presented in Fig. 4.12

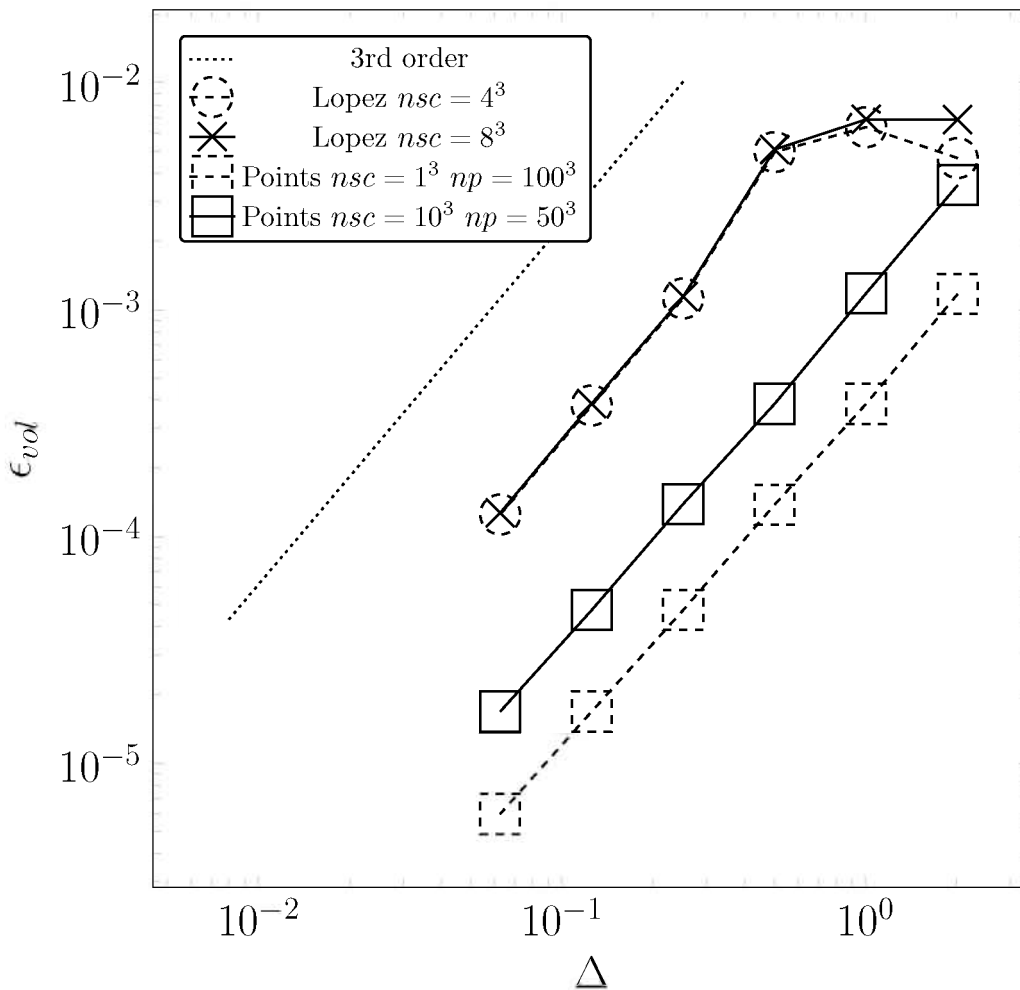


Figure 4.11: Volume fraction initialization errors as a function of grid resolution for a Torus of parameters $R = 2[m]$, $r = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3 np = 100^3$ and $nsc = 10^3 np = 50^3$.

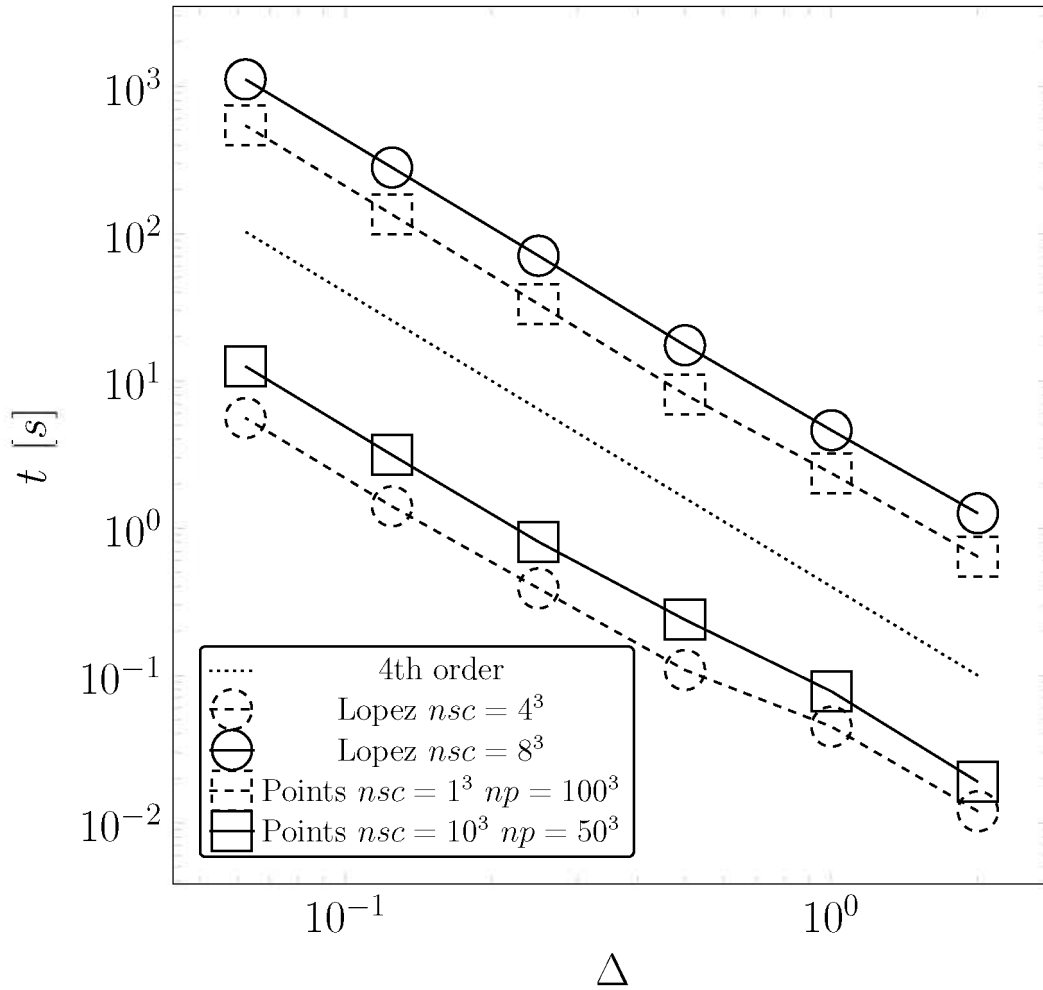


Figure 4.12: Initialization time as a function of grid resolution for a Torus of parameters $R = 2[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m^3]$. Comparison between the results obtained using the López et al. (2009) method with $nsc = 4^3$ and $nsc = 8^3$ and the Points method with $nsc = 1^3 np = 100^3$ and $nsc = 10^3 np = 50^3$.

Table 4.6: Convergence ratio for the error in the color function initialization Torus with parameters $R = 2.0[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = [8]^3[m^3]$

Δ	Lopez		Points	
	$nsc = 4^3$	$nsc = 8^3$	$nsc = 1^3 np = 100^3$	$nsc = 10^3 np = 50^3$
$2/1$	-	-	-	-
$1/1$	0.73	1.00	3.05	3.00
$1/2$	1.29	1.36	2.77	3.07
$1/4$	4.44	4.43	2.92	2.77
$1/8$	2.95	2.97	2.80	2.92
$1/16$	3.01	3.03	2.81	2.80

Figure fig: initialization Torus amr3d presents the color function for the Torus interface initialized with the Points method for each mesh resolution tested.

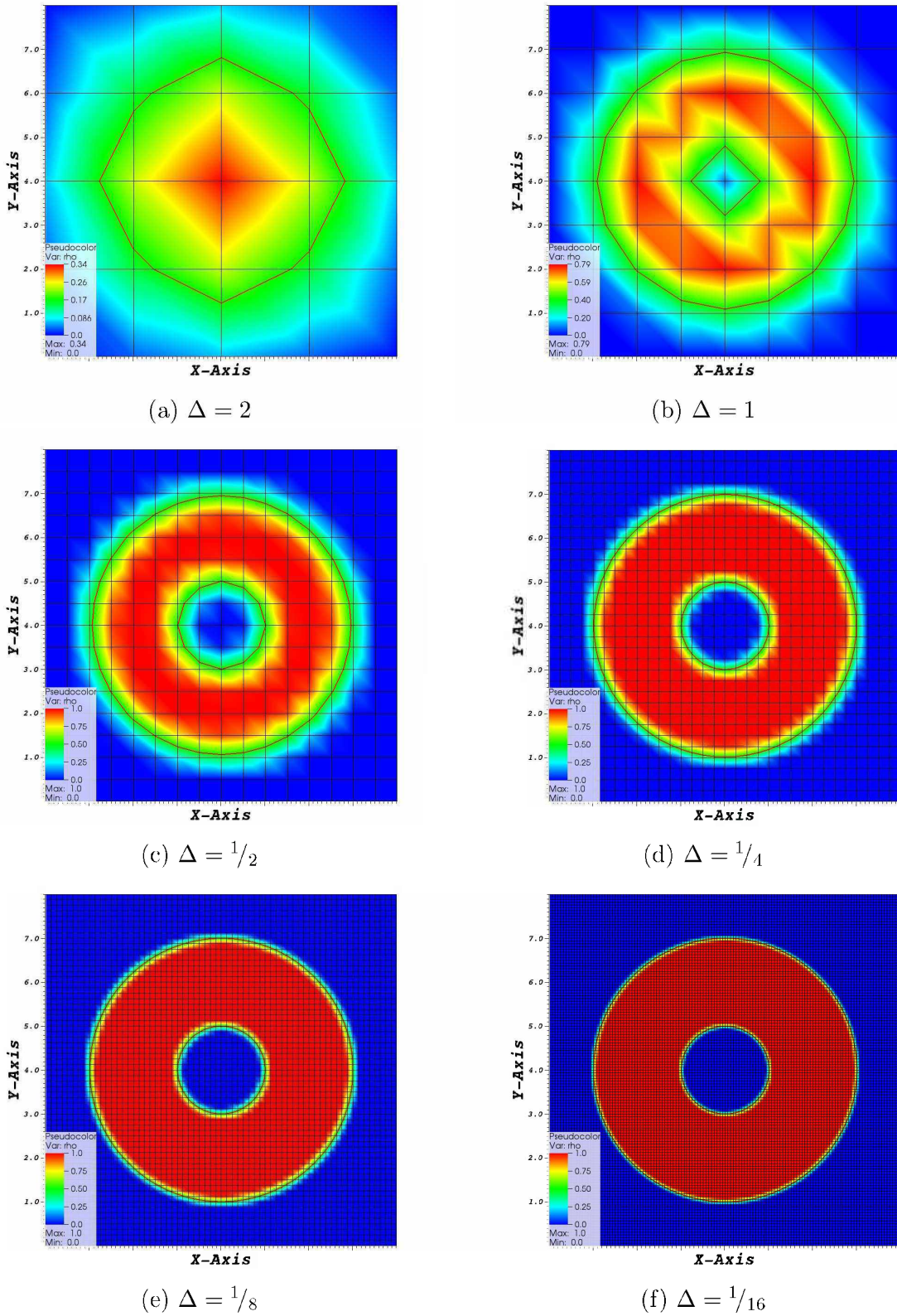


Figure 4.13: Color function of the Torus with parameters $R = 2[m]$ and $r = 1.0[m]$, centered in a domain of size $\Omega = 8^3[m]$, initialized in the AMR3D code with the Points method, for maximum mesh resolution ranging from $\Delta = 2$ to $\Delta = 1/16$

4.1.2 Analysis of the reconstruction and advection step

To analyze the VOF reconstruction and advection, a common test is the so called Shear Flow, proposed by LeVeque (1996). After the initialization of a sphere, a three-dimensional velocity field, given by Eq.(4.5), is imposed in order to perform its movement.

$$\begin{aligned}
 u &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos\left(\frac{\pi t}{T}\right) \\
 v &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos\left(\frac{\pi t}{T}\right) \\
 w &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos\left(\frac{\pi t}{T}\right)
 \end{aligned} \tag{4.5}$$

where $T = 3[s]$ is the time necessary for a complete cycle from the beginning to the return of the starting position.

A sphere with $r = 0.15[m]$ in a unit domain, at the starting position $x_c = y_c = z_c = 0.35[m]$, and a uniform mesh of $n_x = n_y = n_z = 150$ is used. The mass loss is compared with literature data and is presented in Tab. 4.7. The result obtained with the AMR3D code is presented in Fig. 4.14.

Table 4.7: Mass loss for the VOF method in a 3D Shear Flow test. $T = 3.0[s]$ and $\Delta t = \frac{\Delta x}{2}[s]$

Mesh [150^3]	Mass loss
Current work	0.0163 %
Menard, Tanguy and Berlemont (2007)	< 0.0300 %
Aulisa et al. (2007)	3.5050 %

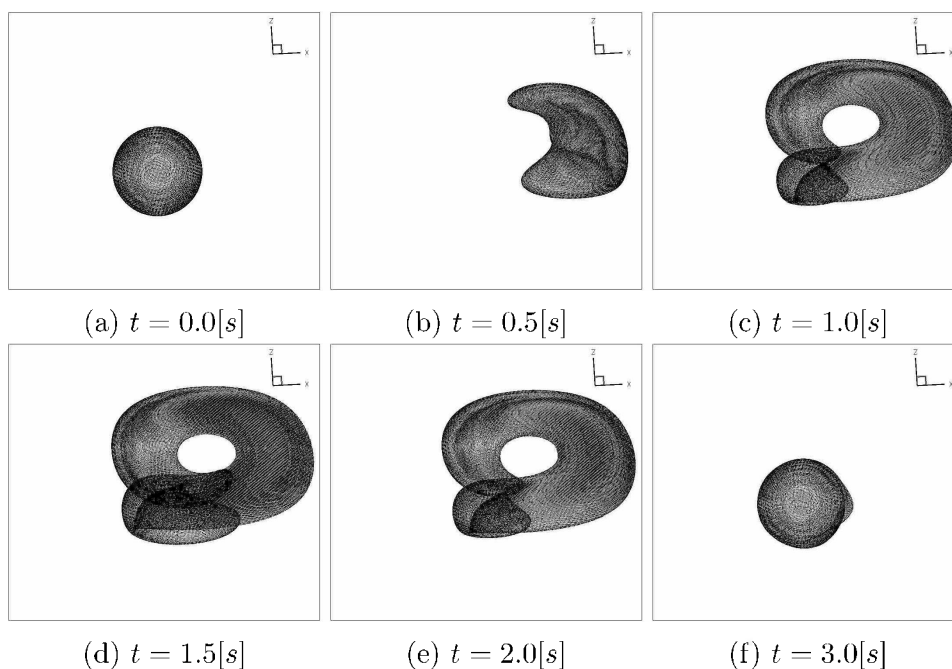


Figure 4.14: Three dimensional Shear flow test performed with the AMR3D code

4.1.3 Normal and Curvature computation - Simulation set-up

An overview of the methods to compute the normal vector and the curvature in the context of the VOF method were presented. In this section the simulation set-up used to evaluate the results are explained.

For all simulations performed the smoothed color function is computed accordingly to Annaland, Deen and Kuipers (2005)

$$\tilde{F}_{(x,y,z)} = \sum_m D(x - x_m)D(y - y_m)D(z - z_m)F_{(x_m,y_m,z_m)} \quad (4.6)$$

with the smoothing function D as:

$$D(x) = \frac{1}{2h} \left(1 + \cos \left(\pi \frac{x}{h} \right) \right) \quad (4.7)$$

where $h = 2$ is the width of the computational stencil used for smoothing and Δ is the Eulerian grid size.

To compute the exact normal vector of the surface symmetric about its center, a vector \mathbf{r} is defined as:

$$\mathbf{r} = \mathbf{x} - \mathbf{x}_0 \quad (4.8)$$

where \mathbf{x}_0 is the center of the surface, and \mathbf{x} is the center of the cell which contains the interface.

The “theoretical” normal vector \mathbf{n}_t is defined as:

$$\mathbf{n}_t = \frac{\mathbf{r}}{|\mathbf{r}|} \quad (4.9)$$

With the “theoretical” normal (\mathbf{n}_t) and the computed normal (\mathbf{n}) the error (\mathbf{e}_{rr}) is defined as:

$$\mathbf{e}_{rr} = \mathbf{n} - \mathbf{n}_t \quad (4.10)$$

The L_2 and L_∞ norms are used to evaluate the errors, both for the normal vector and

the curvature. These norms are defined as:

$$L_2(\mathbf{u}) = \sqrt{\sum_{n=1}^N \|\mathbf{u}_n\|^2} \quad (4.11)$$

$$L_\infty(\mathbf{u}) = \|\mathbf{u}\|_{max} = \max(\|\mathbf{u}\|) \quad (4.12)$$

4.1.4 Normal and Curvature simulation results

An analysis of the Least Squares method for the normal and curvature computation is performed, which allows to choose a proper configuration of the method. The analytic solution for the normal vector is then compared with the results obtained from the Young's, the Shirani's based discretization and the Least Squares methods, and the influence of smoothing the color function is evaluated.

Later, the methods to compute the curvature are also compared with the analytic solution. They are: the Shirani's method, the 27 Cells discretization, the Paraboloid, the Least Squares, the Height Function coupled with the Paraboloid, Height Function coupled with the Shirani and Height Function coupled with the 27 Cells discretization.

For the following analysis, two interfaces are tested, a sphere with radius $r = 0.25[m]$ and an ellipsoid with parameters $a = 0.3[m]$, $b = 0.25[m]$ and $c = 0.20[m]$, both centered in a unit cubic domain. They are initialized with the Points method, with $nsc = 10^3$ and $np = 500^3$, as presented in Sections 4.1.1.2 and 4.1.1.4.

The L_2 and L_∞ norms of the errors are computed, both for the normal and curvature, and the mean curvature is also presented for the sphere, as it is a constant value inversely proportional to its radius ($H = \frac{1}{r}$).

4.1.4.1 Normal analysis: sphere. Least Squares Method

For the normal computation with the Least Squares method, the number of neighbors, the geometrical weighting factor g and the use or not of the smoothed color function are parameters that can be adjusted. A stencil of size s means the number of points used in each coordinate direction. This results in $s^2 - 1$ neighbor points in 2D and $s^3 - 1$ neighbor points in 3D, since the point itself is not used.

The nomenclature adopted is: LS $s_i g_j$, where the subscript i means the stencil size, as explained, and the subscript j means which geometrical weighting factor is used.

Figures 4.15 to 4.18 presents the influence of the stencil size, the geometrical weighting

factor and the smoothening of the color function on the results from the normal computed with the Least Squares method.

The use of the smoothed color function reduces the magnitude of the error, but for the same stencil size and geometrical weighting factor, it does not alter the convergence ratio, presented in Tab. 4.8, with the exception being the cases with $s \leq 5$ and $\Delta = 1/32$.

Concerning the geometrical weighing factor, it also decreases the magnitude of the error. For the L_2 error (for both the non-smoothed and the smoothed color function), the convergence ratio, presented in Tab. 4.8, is not influenced by $g = 1$. With $g = 2$ the convergence ratio is bigger than with $g = 1$ for $\Delta = 1/32$, and is less influenced for $\Delta < 1/64$.

For $\Delta = 1/128$, the convergence ratio has order 1.4 for the non-smoothed color function, with the exception being the case with $s = 7$, $g = 0$ and $g = 1$, with order 1.26. For the smoothed color function the L_2 convergence ratio is approximately 1.3, and for $s = 7$ and $g = 2$ the order is 1.2.

For the L_∞ error, the geometrical weighting factor has no influence on the convergence error, which is of order 1 for $\Delta = 1/128$.

With these results, the Least Square method with stencil size $s = 5$ and geometrical weighting factor $g = 2$ is chosen for the comparison with the other normal computation methods: Young and Shirani.

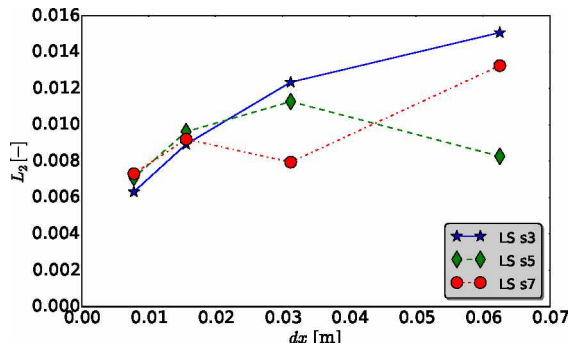
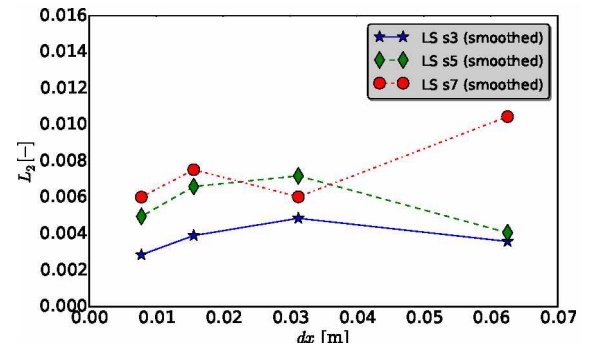
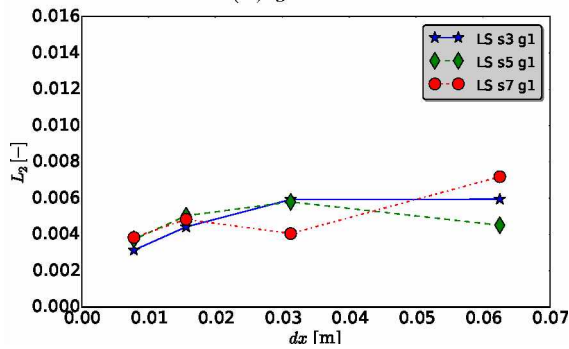
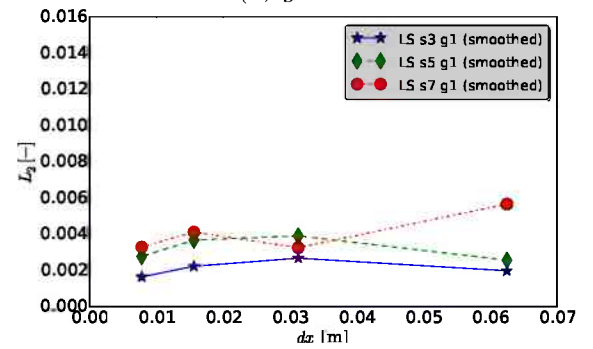
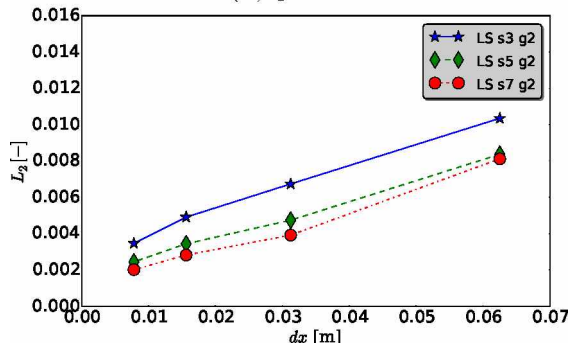
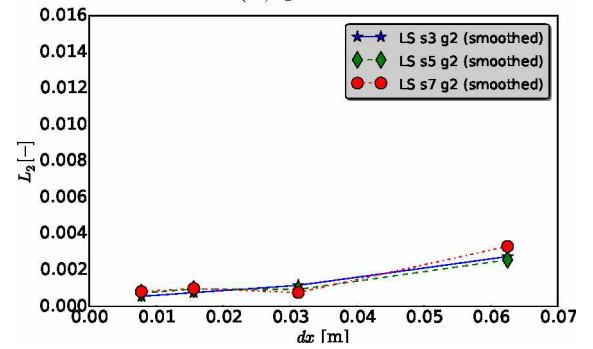
(a) $g = 0$ (a) $g = 0$ (b) $g = 1$ (b) $g = 1$ (c) $g = 2$ (c) $g = 2$

Figure 4.15: L_2 error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method.

Figure 4.16: L_2 error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method and the smoothed color function.

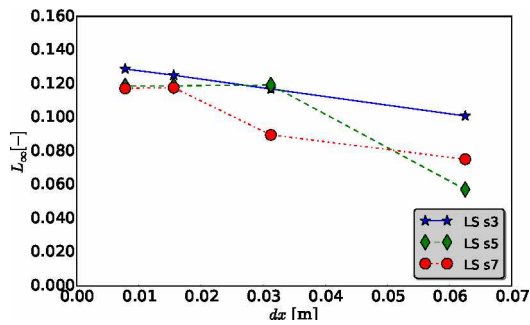
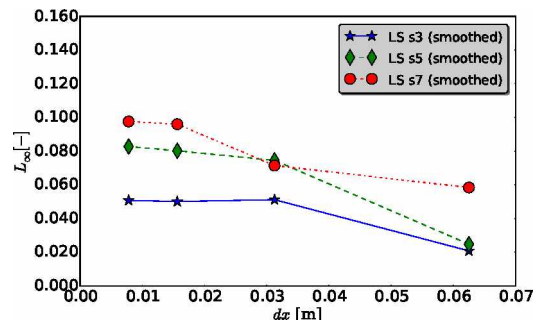
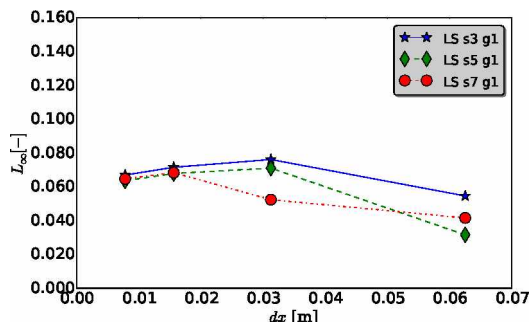
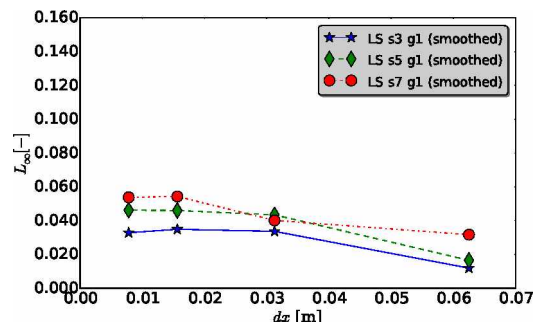
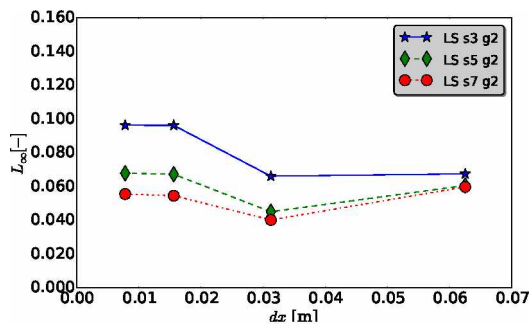
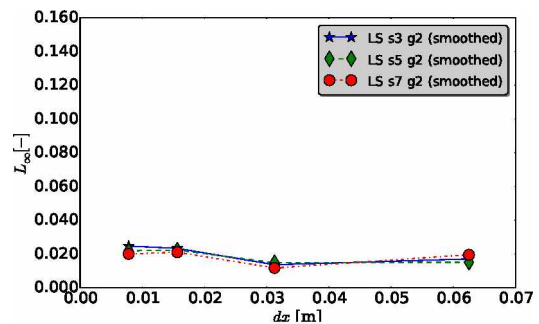
(a) $g = 0$ (a) $g = 0$ (b) $g = 1$ (b) $g = 1$ (c) $g = 2$ (c) $g = 2$

Figure 4.17: L_∞ error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method.

Figure 4.18: L_∞ error norm as a function of the mesh refinement, the stencil size and the geometrical weighting factor, for the normal vector of a sphere with $r = 0.25[m]$ computed with the Least Squares method and the smoothed color function.

Table 4.8: Convergence ratio for normal vector of a sphere with radius $r = 0.25[m]$, computed with the Least Squares method.

	$s3$	$s5$	$s7$	$s3g1$	$s5g1$	$s7g1$	$s3g2$	$s5g2$	$s7g2$
Δ	L_2								
$1/32$	1.22	0.73	1.67	1.00	0.78	1.77	1.54	1.77	2.07
$1/64$	1.38	1.17	0.86	1.34	1.15	0.84	1.37	1.38	1.39
$1/128$	1.41	1.36	1.26	1.41	1.35	1.26	1.41	1.41	1.40
	L_2 Smoothed color function								
$1/32$	0.74	0.57	1.73	0.74	0.66	1.74	2.37	2.75	4.36
$1/64$	1.24	1.09	0.80	1.20	1.06	0.79	1.53	0.99	0.78
$1/128$	1.37	1.33	1.25	1.36	1.33	1.25	1.38	1.29	1.20
	L_∞								
$1/32$	0.86	0.48	0.84	0.72	0.44	0.79	1.02	1.35	1.49
$1/64$	0.93	1.00	0.76	1.06	1.05	0.77	0.69	0.67	0.74
$1/128$	0.97	1.00	1.00	1.07	1.07	1.06	1.00	0.99	0.98
	L_∞ Smoothed color function								
$1/32$	0.41	0.33	0.82	0.35	0.38	0.79	1.25	1.02	1.67
$1/64$	1.02	0.93	0.74	0.96	0.94	0.74	0.58	0.67	0.56
$1/128$	0.99	0.97	0.98	1.06	1.00	1.01	0.95	1.02	1.05

4.1.4.2 Normal analysis: sphere. Comparison between the Least Squares, the Young and the Shirani methods

Figures 4.19 and 4.20 present the L_2 and L_∞ error norms, respectively, as a function of the mesh refinement for the sphere ($r = 0.25[m]$) with the Young, Shirani and Least Squares methods. The L_2 error norm decreases with the mesh refinement, but the L_∞ error norm remains almost unchanged on the finer meshes.

The use of the smoothed color function improves the accuracy of the normal computation in all methods; both the L_2 and L_∞ error norms presents smaller values when compared with the non-smoothed results.

The convergence ration is presented in Tab. 4.9. For the L_2 and L_∞ error norms and $\Delta = 1/32$, the Young method has the lower order and the Least Squares the bigger; for $\Delta = 1/128$ and non-smoothed color function, all method presents order 1.4 for the L_2 error and order 1.0 for the L_∞ error. For the smoothed color function the L_2 error is of order 1.35 for the Young and Shirani and 1.29 for the Least Squares, and order 1.0 for the L_∞ .

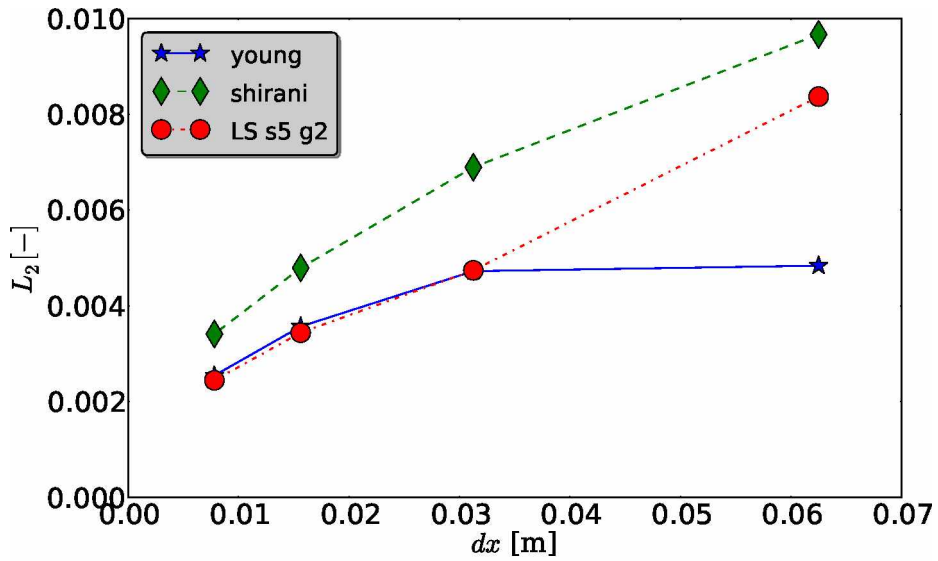
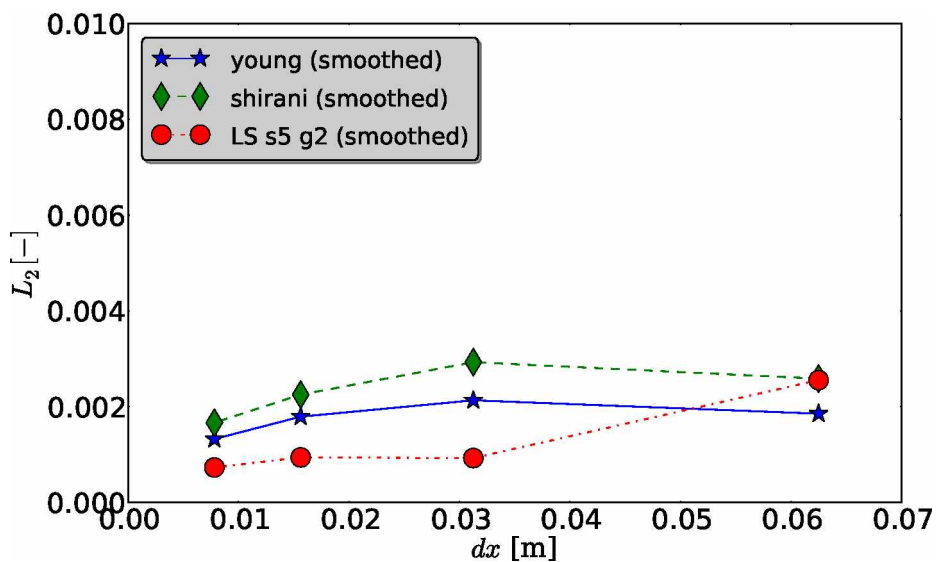
(a) L_2 error norm with the non-smoothed color function.(b) L_2 error norm with the smoothed color function.

Figure 4.19: L_2 error norm as a function of the mesh refinement for the normal vector of a sphere with $r = 0.25[m]$, computed with the Least Squares, the Young and the Shirani methods.

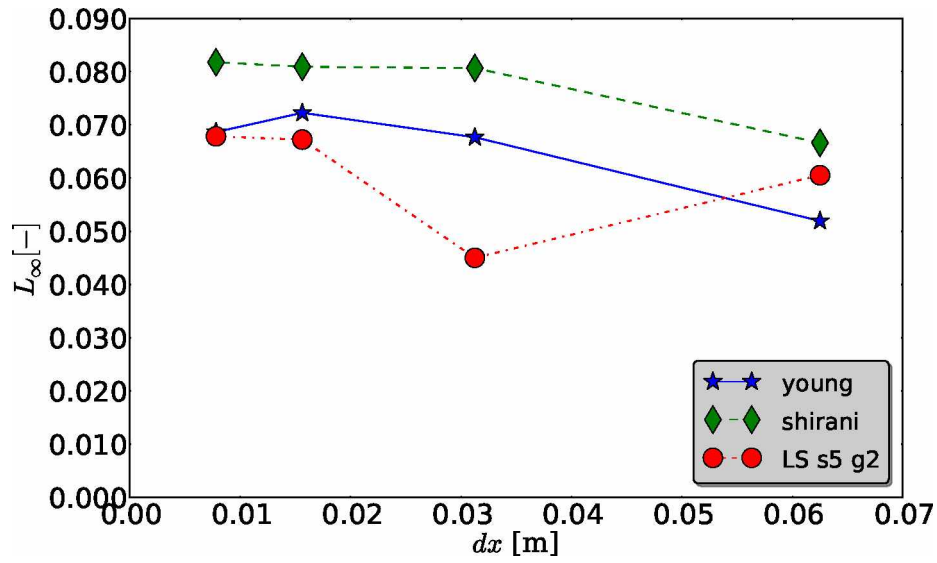
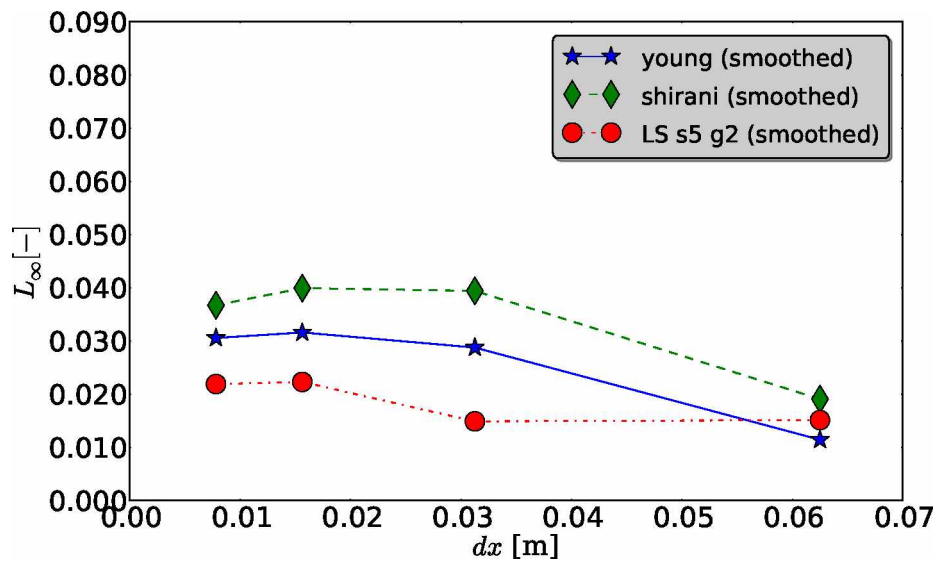
(a) L_∞ error norm with the non-smoothed color function.(b) L_∞ error norm with the smoothed color function.

Figure 4.20: L_∞ error norm as a function of the mesh refinement for the normal vector of a sphere with $r = 0.25[m]$, computed with the Least Squares, the Young and the Shirani methods.

Table 4.9: Convergence ratio for the normal vector of a sphere computed with the Least Squares, the Young and the Shirani methods.

Δ	L_2			L_∞		
	Young	Shirani	LS	Young	Shirani	LS
$1/32$	1.02	1.40	1.77	0.77	0.83	1.35
$1/64$	1.32	1.44	1.38	0.94	1.00	0.67
$1/128$	1.40	1.40	1.41	1.05	0.99	0.99
Smoothed color function						
$1/32$	0.87	0.88	2.75	0.40	0.48	1.02
$1/64$	1.19	1.30	0.99	0.91	0.99	0.67
$1/128$	1.35	1.35	1.29	1.03	1.09	1.02

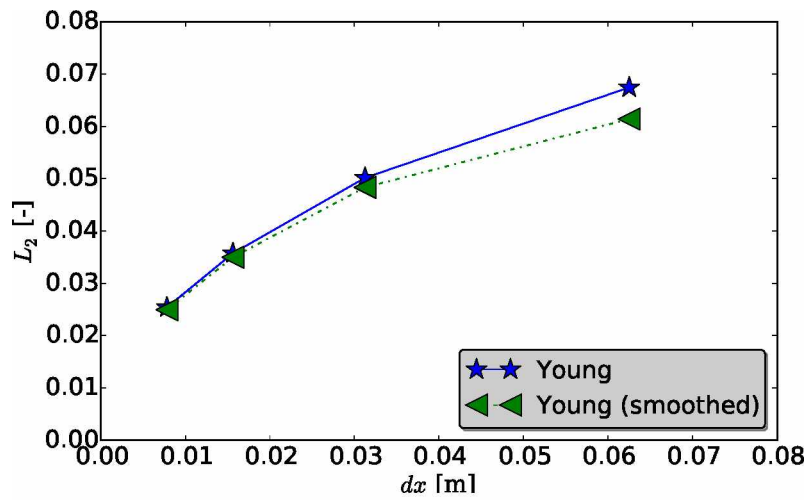
4.1.4.3 Normal analysis: ellipsoid. Comparison between the Least Squares, the Young and the Shirani methods.

For the ellipsoid, Fig. 4.21 and 4.22 shows that smoothing the color function also decreases the L_2 and the L_∞ error norms for the three methods tested, but with much less impact than in the case of the sphere. Figures 4.23 and 4.24 compares the L_2 and L_∞ error norms, respectively, between the three methods tested.

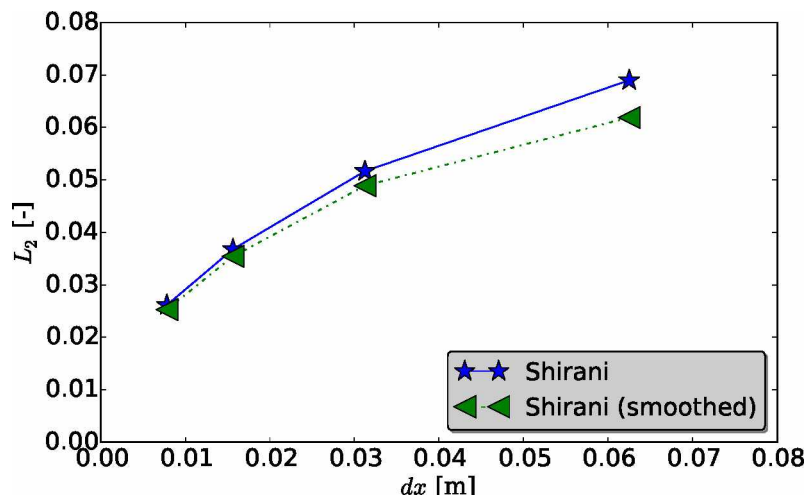
The convergence ratio is presented in Tab. 4.10, being of order 1.4 for the L_2 error norm and $\Delta = 1/128$ and order 1 for the L_∞ error norm, both for the non-smoothed and the smoothed color function.

Table 4.10: Convergence ratio for normal vector of an ellipsoid, computed with the Least Squares, the Young and the Shirani methods.

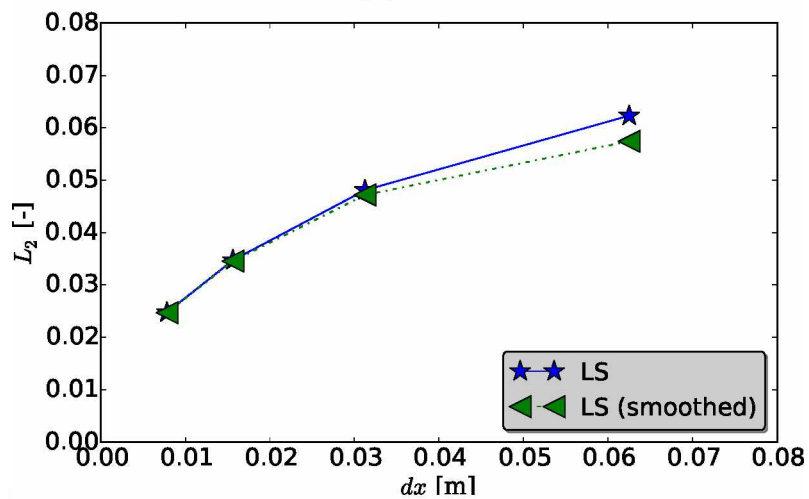
Δ	L_2			L_∞		
	Young	Shirani	LS	Young	Shirani	LS
$1/32$	1.34	1.34	1.30	0.93	0.90	0.89
$1/64$	1.40	1.40	1.38	1.05	1.01	1.02
$1/128$	1.41	1.41	1.40	1.03	1.01	1.02
Smoothed color function						
$1/32$	1.27	1.27	1.22	0.87	0.85	0.83
$1/64$	1.38	1.38	1.37	1.03	1.03	1.01
$1/128$	1.40	1.40	1.40	1.01	1.00	1.00



(a) Young

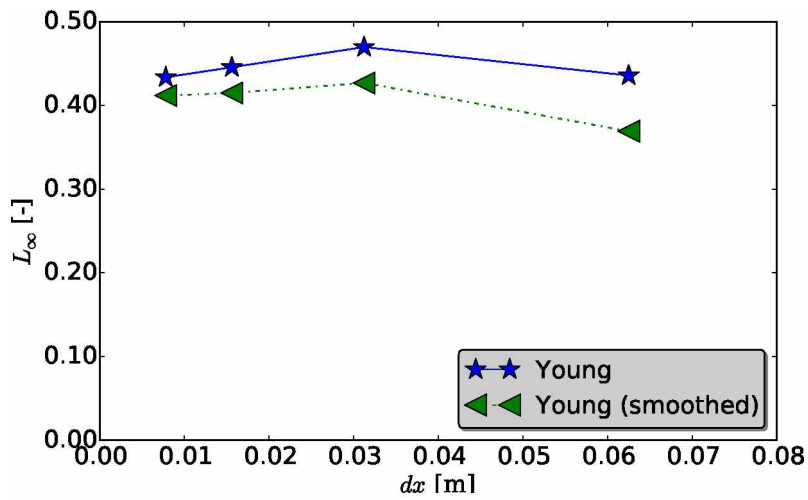


(b) Shirani

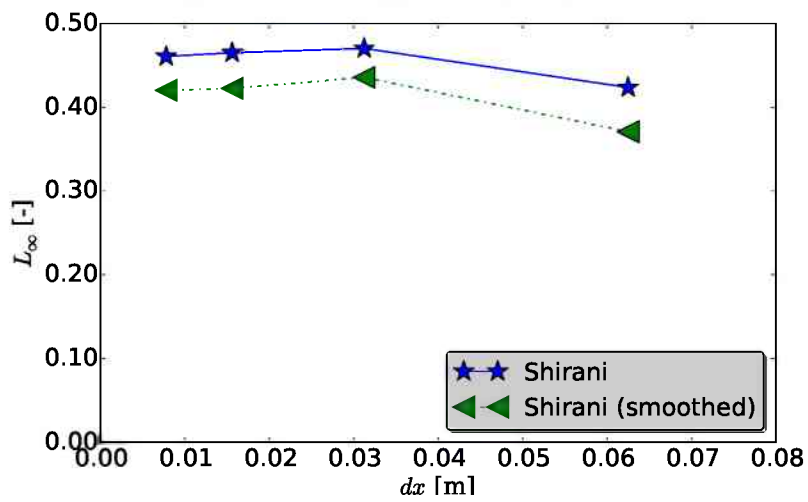


(c) Least Squares

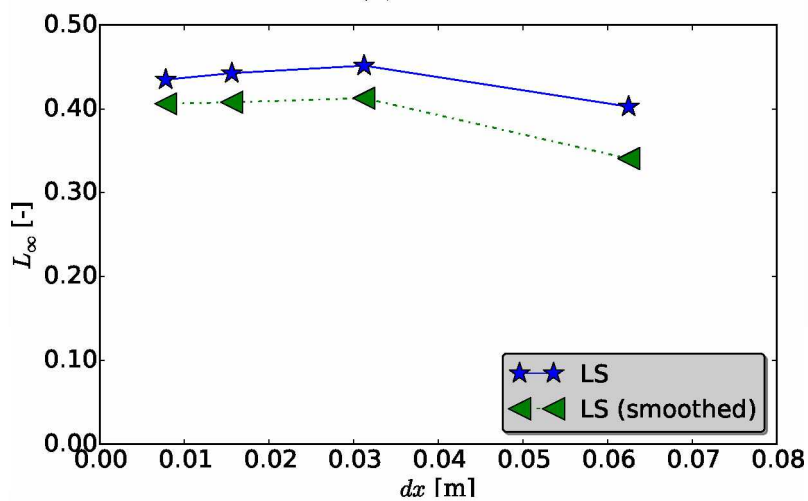
Figure 4.21: L_2 error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3$ [m^3]. Comparison between the non-smoothed and the smoothed color function.



(a) Young

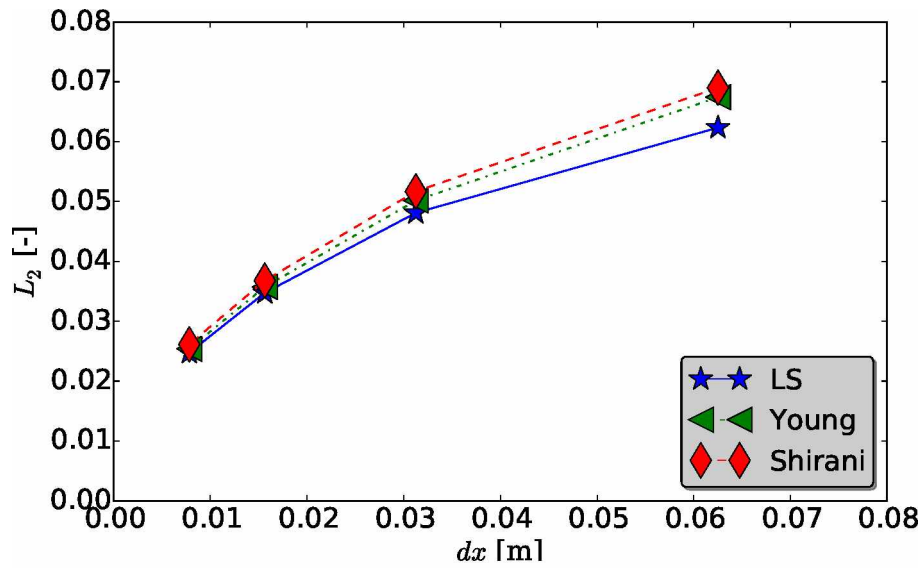


(b) Shirani

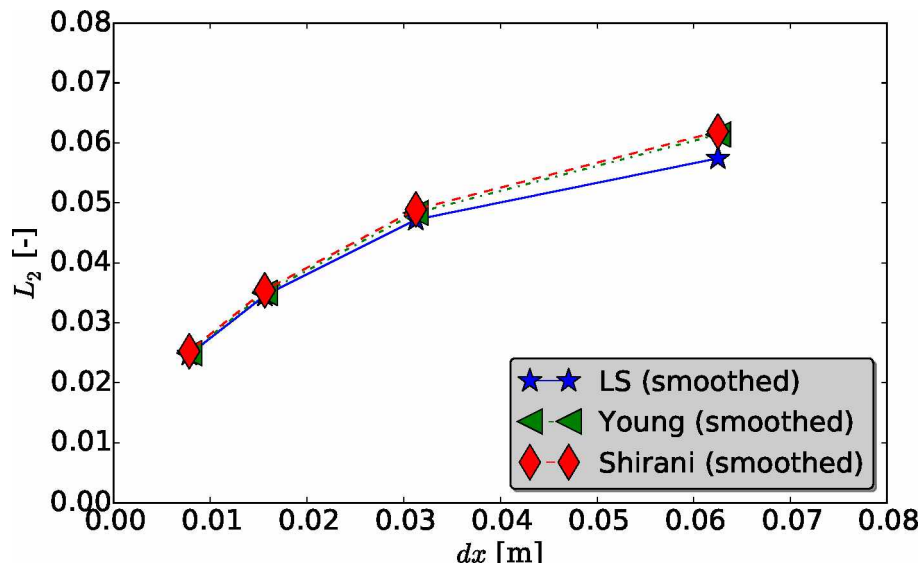


(c) Least Squares

Figure 4.22: L_∞ error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3$ [m^3]. Comparison between the non-smoothed and the smoothed color function.

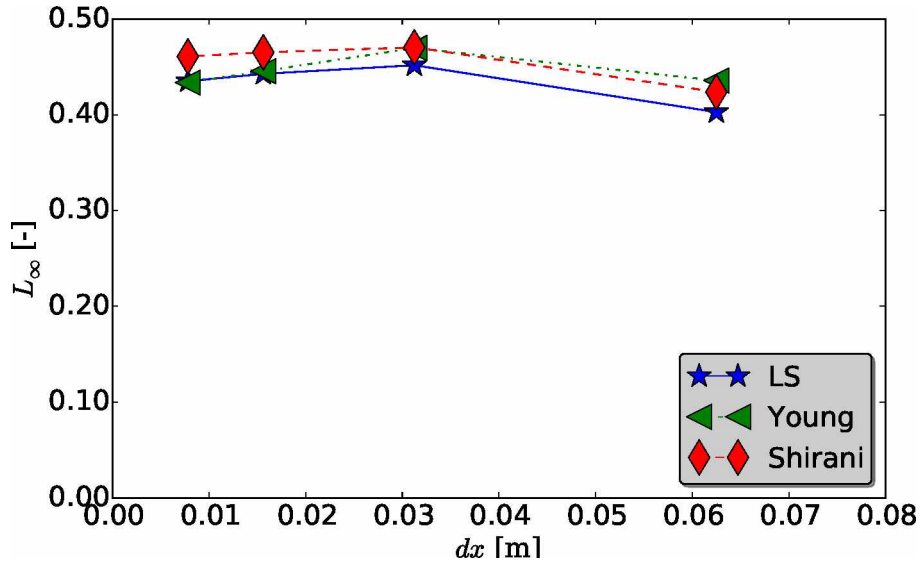


(a) Non-smoothed color function.

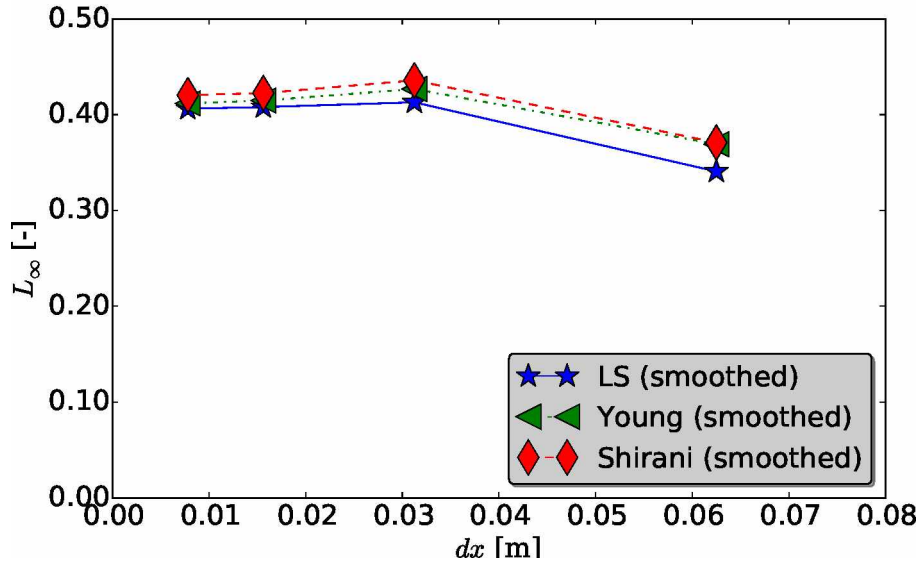


(b) Smoothed color function.

Figure 4.23: L_2 error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3 [m^3]$. Comparison between the Young, Shirani and Least Squares methods.



(a) Non-smoothed color function.



(b) Smoothed color function.

Figure 4.24: L_∞ error norm as a function of the mesh refinement for the normal vector computation of an ellipsoid with parameters $a = 0.3$, $b = 0.25$ and $c = 0.20$, initialized in a domain of size $\Omega = [1]^3$ [m^3]. Comparison between the Young, Shirani and Least Squares methods.

4.1.4.4 Curvature analysis: sphere. Least Squares Method

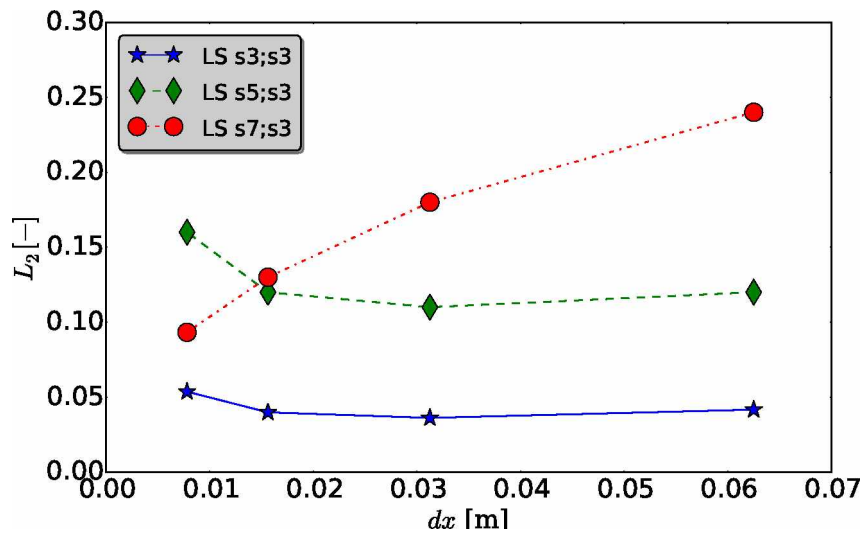
For the curvature computation with the Least Squares method, both the stencil size for the normal and the curvature can be adjusted, as well as the geometrical weighting factor and the use or not of the smoothed color function. The nomenclature adopted to indicate each test case is as follows: LS $s_\kappa s_n$, where the subscript κ refers to the stencil size used for the curvature and the subscript n the stencil size for the normal.

The first analysis is to relate the stencil size for the normal computation and the stencil size for the curvature computation. These test cases are tabulated in Tab. 4.11 and

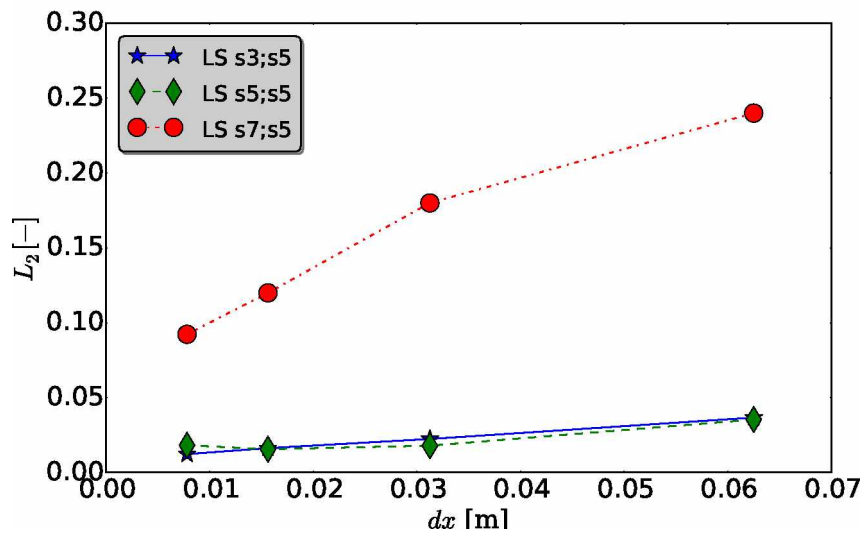
the results are presented in Fig. 4.25; from the results, the stencil for the normal should be equal or bigger than the stencil for the curvature. All cases where the stencil for the curvature is bigger than the stencil for the normal presents bigger L_2 error norms. The cases that satisfies the criteria $s_\kappa \leq s_n$ in Fig. 4.25 are compared in Fig. 4.26.

Table 4.11: Test cases relating the stencil size for the normal and for the curvature, for the curvature computation of a sphere with $r = 0.25[m]$ with the Least Squares method.

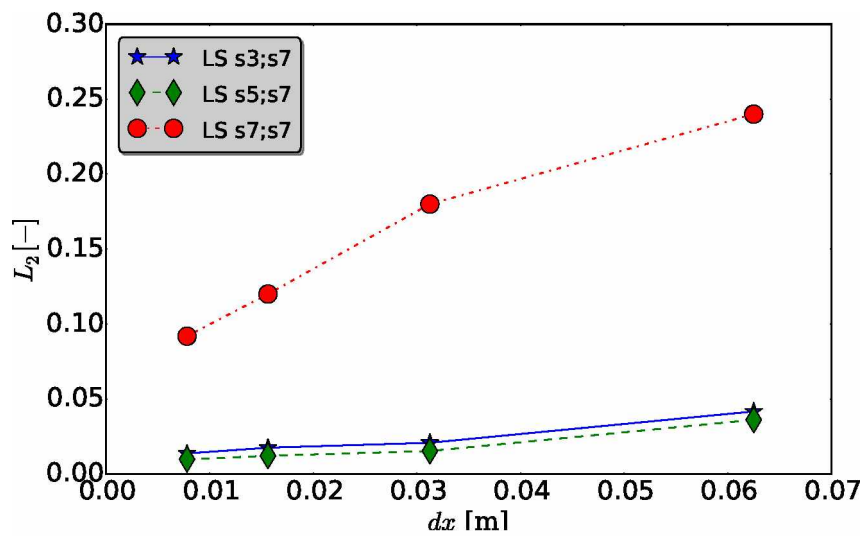
Normal	Curvature		
	s3	s5	s7
s3	Fig. 4.25a		
s5	Fig. 4.25b		
s7	Fig. 4.25c		



(a)



(b)



(c)

Figure 4.25: L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m], computed with the Least Squares method. Analysis of the relation between the stencil sizes of the normal and the curvature.

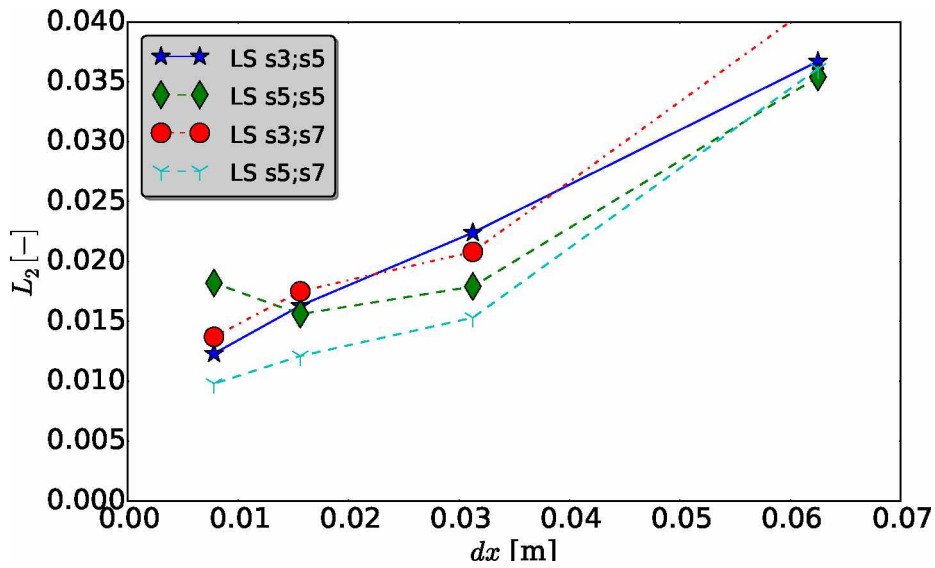
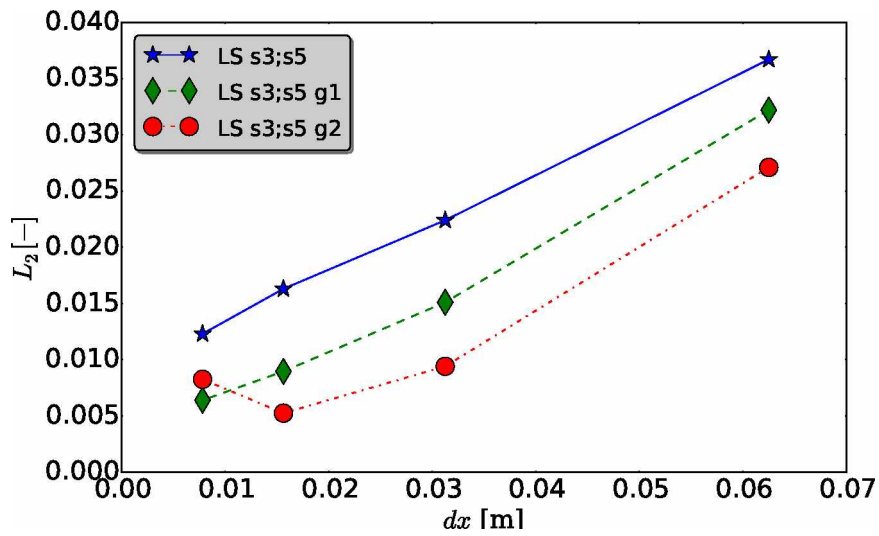


Figure 4.26: L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[m]$ computed with the Least Squares method for the cases: LS s3 s5, LS s5 s5, LS s3 s7 and LS s5 s7.

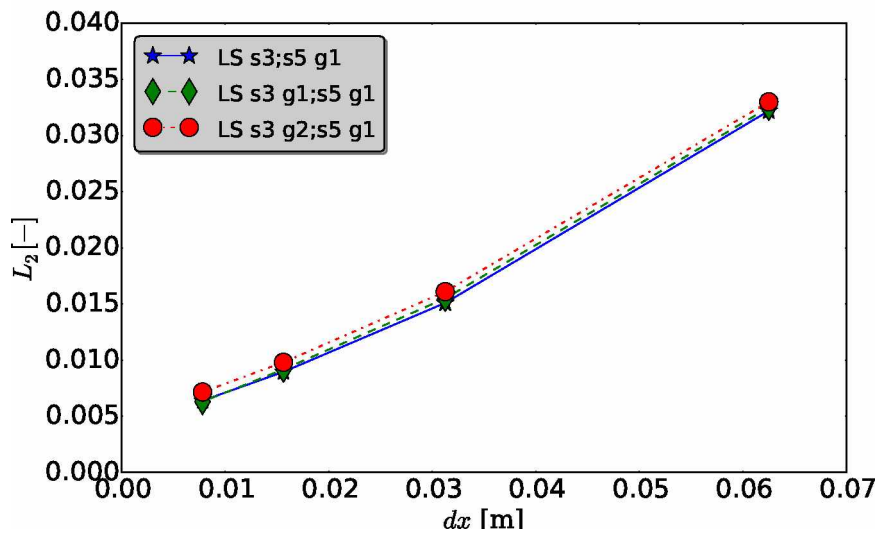
An analysis of the geometrical weighting factor is performed for LS s3 s5. The cases are presented in Tab. 4.12 and Fig. 4.27. From Fig. 4.27a, the geometrical weighting factor in the normal computation reduces the L_2 error norm. For the normal with $g = 1$, the geometrical weighting factor applied to the curvature computation does not affect the results, as presented in Fig. 4.27b. For the normal with $g = 2$ the L_2 error norm increase for the more refined meshes, as presented in Fig. 4.27c. From the results, the normal with geometrical weighting factor $g = 1$ reduces the L_2 error norm, and with $g = 2$ on the normal computation, $g = 1$ on the curvature computation leads to a smaller L_2 error norm.

Table 4.12: Test cases for Least Squares curvature computation of a sphere with radius $r = 0.25[m]$, with stencil size 5 for the normal and 3 for the curvature, with and without geometrical weighting factor.

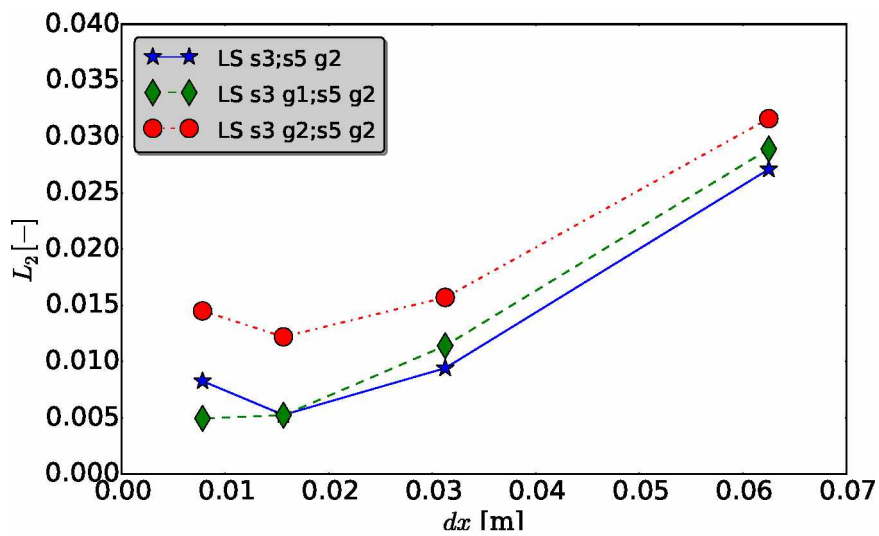
Normal	Curvature		
	s3	s3g1	s3g2
s5	Fig. 4.27a	X	X
s5g1	Fig. 4.27a/Fig. 4.27b	Fig. 4.27b	Fig. 4.27b
s5g2	Fig. 4.27a and Fig. 4.27c	Fig. 4.27c	Fig. 4.27c



(a)



(b)



(c)

Figure 4.27: L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m], computed with the Least Squares method. Analysis of the use of the geometrical weighting factor on the case LS s3 s5.

For the case LS s3 s5g1, the results with the use or not of the smoothed color function on the normal is presented in Fig. 4.28. It shows that with the smoothed color function the L_2 and the L_∞ error norms are smaller.

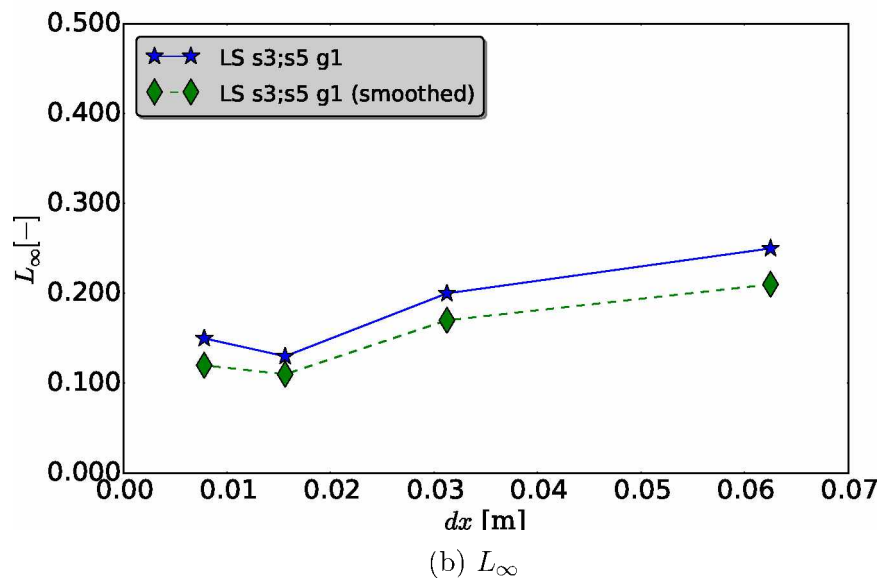
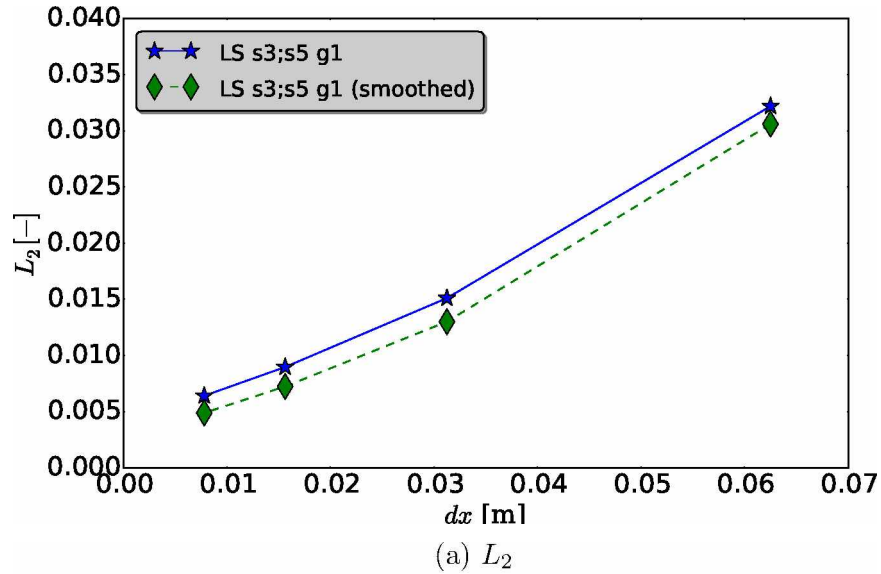


Figure 4.28: error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m] computed with the Least Squares method. Analysis of the use of the smoothed color function on the case LS s3 s5g1.

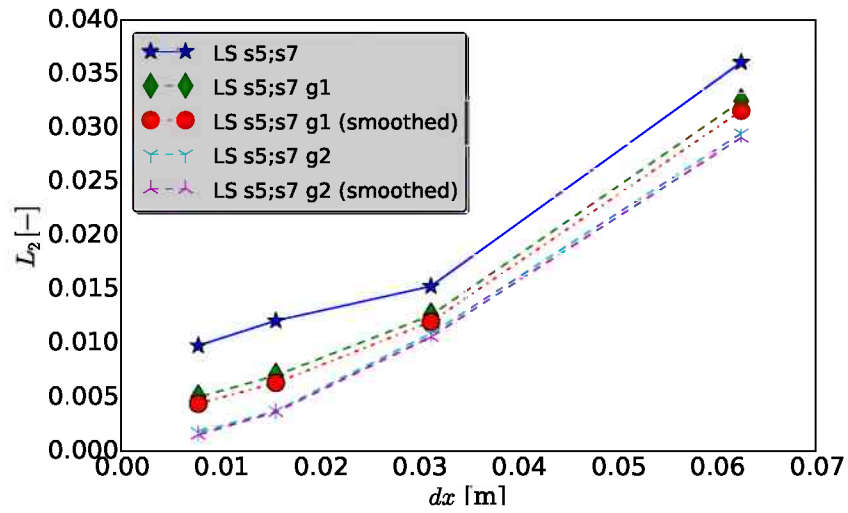
An analysis for the case LS s5 s7 is also performed. The cases are presented in Tab. 4.13 and Fig. 4.29. From Fig. 4.29a, for the normal computation with $g_n = 1$ the L_2 error norm is smaller than without the geometrical weighting factor, but smoothing the color function has no effect. For $g_n = 2$ the L_2 error norm is lower than with $g_n = 1$, and again smoothing the color function does not affect the error.

For the cases where the curvature is computed with $g_\kappa = 1$, presented in Fig. 4.29b, and the normal is computed with $g_n = 1$, the smoothed color function does not change the error norm. For normal computed with $g_n = 2$ ($g_n > g_\kappa$) the error norm is smaller but

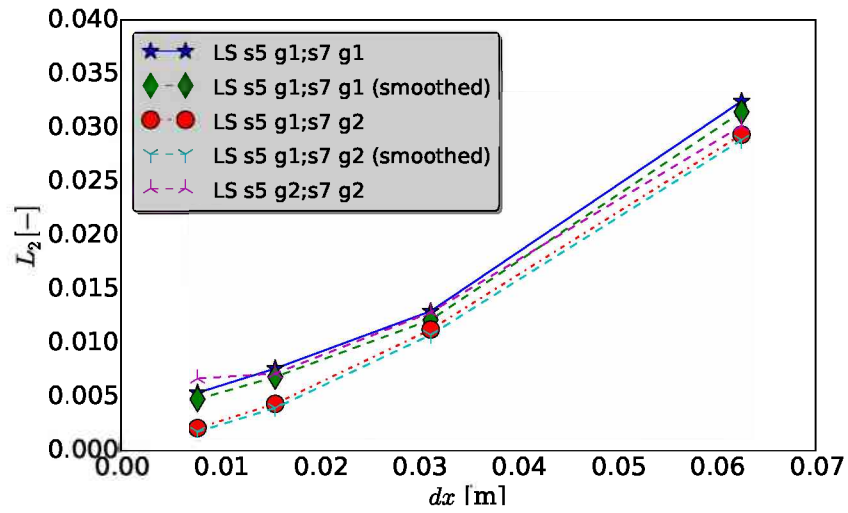
does not change with geometrical weighting factor on the curvature nor with the use of the smoothed color function.

Table 4.13: Test cases for Least Squares curvature computation of a sphere with radius $r = 0.25[m]$, with stencil size 7 for the normal and 5 for the curvature, with and without geometrical weighting factor.

Normal	Curvature		
	s5	s5g1	s5g2
s7	Fig. 4.29a	X	X
s7g1	Fig. 4.29a	Fig. 4.29b	X
s7g1 smoothed	Fig. 4.29a	Fig. 4.29b	X
s7g2	Fig. 4.29a	Fig. 4.29b	Fig. 4.29c
s7g2 smoothed	Fig. 4.29a	Fig. 4.29b	X



(a)



(b)

Figure 4.29: L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m] computed with the Least Squares method, computational stencil size $s = 7$ for the normal and $s = 5$ for the curvature, and geometrical weighting factor $0 \leq g \leq 2$.

Figure 4.30 presents the L_2 and the L_∞ error norms for selected cases Ls s5 s7 (the cases with lower L_2 error norms from Fig. 4.29). It shows that the cases with $g_\kappa < g_n$ have lower errors. For this stencil size, smoothening the color function does not affect the error.

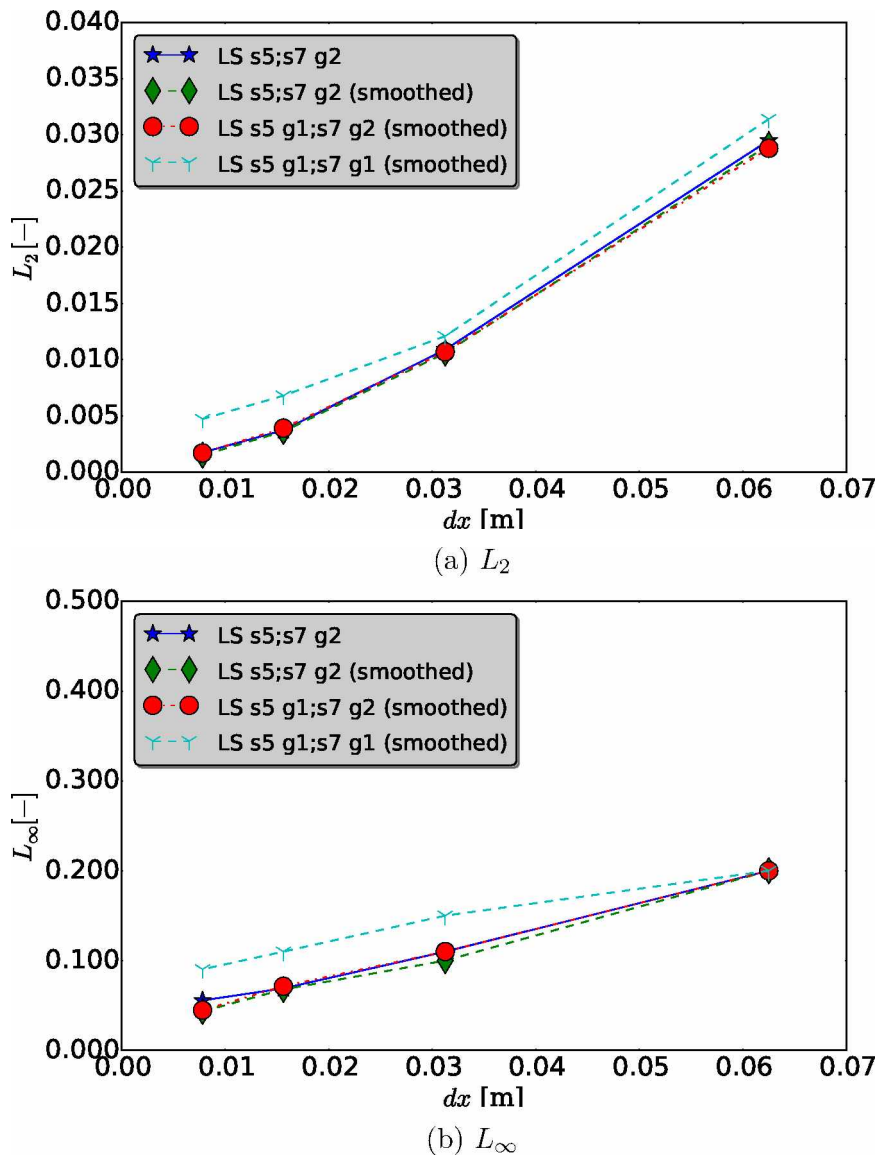


Figure 4.30: error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25[\text{m}]$ computed with the Least Squares method and cases LS s5 s7.

Some of the cases with lower L_2 error norm for LS s3;s5 and LS s5;s7 are compared in Fig. 4.31 and the convergence ratio is presented in Tab. 4.14. For the L_2 error norm the convergence ratio has order bigger than 2 for the coarse meshes. For LS s3;s5 the order drops to 1.5 for the more refined mesh, and stay bigger than 2 for LS s5;s7. For the L_∞ again the convergence ratio order drops with mesh refinement, being 0.92 for LS s3;s5 and $\Delta = 1/128$, 1.24 for LS s5;s7g2 and 1.6 for LS s5g1;s7g2 smoothed.

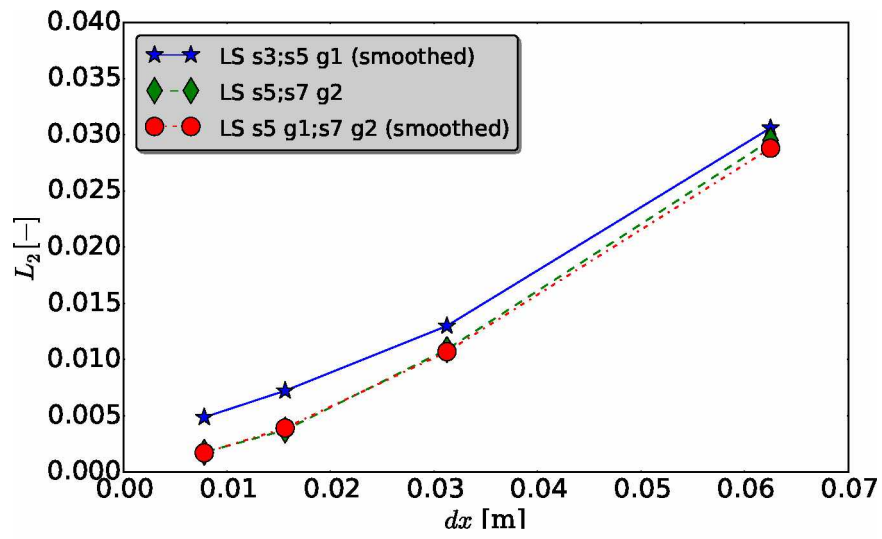
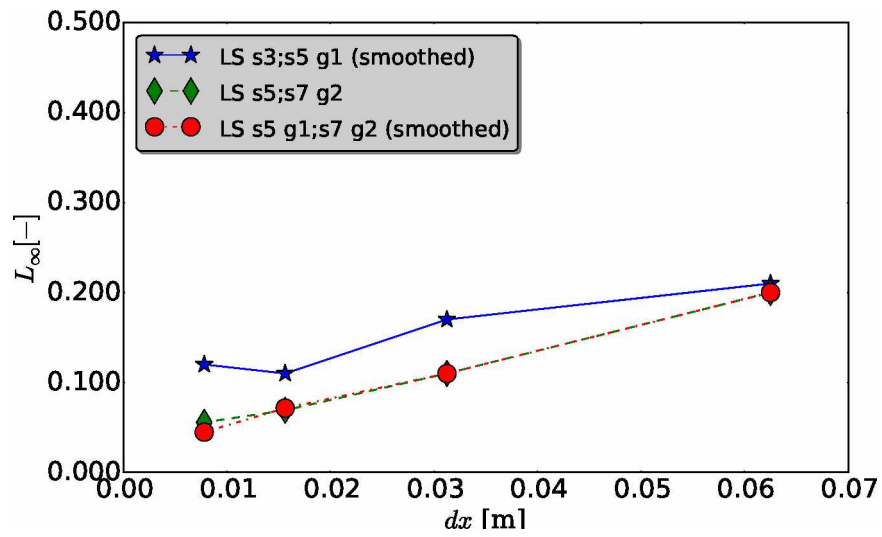
(a) L_2 (b) L_∞

Figure 4.31: error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m] computed with the Least Squares method. Comparison between cases LS s3;s5 g1 (smoothed), LS s5;s7g2 and LS s5g1;s7 g2 (smoothed).

Table 4.14: Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Least Squares method and stencil sizes LS s3;s5 and LS s5;s7

	s3s5g1c	s5s7g2	s5g1s7g2c
Δ	L_2		
$1/32$	2.35	2.71	2.69
$1/64$	1.80	2.92	2.74
$1/128$	1.48	2.13	2.29
	L_∞		
$1/32$	1.24	1.82	1.82
$1/64$	1.55	1.59	1.54
$1/128$	0.92	1.24	1.60

4.1.4.5 Curvature analysis: sphere. Paraboloid method

For the Paraboloid method, three computational stencil sizes are analyzed, $s = 3$, $s = 5$ and $s = 7$, and the use of the smoothed color function is evaluated for the stencil of size $s = 5$. Figure 4.32 presents the results of the L_2 and L_∞ error norms and the mean curvature. It shows that the use of the smoothed color function harms the results, and there is no significant difference concerning the stencil size for the more refined meshes.

The L_2 convergence ratio, presented in Tab. 4.15, has order 1.4, 4.2 and 5.6 for $\Delta = 1/128$ and stencil sizes 3, 5 and 7, respectively; for the L_∞ the convergence ratio, for $\Delta = 1/128$, has order 0.9, 1.6 and 2.9 for stencil sizes 3, 5 and 7, respectively. The convergence ratio is smaller than 1 for the case with the smoothed color function.

Table 4.15: Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Paraboloid method and stencil sizes 3, 5 and 7.

	s_3	s_5	s_7	$s_5 \bar{F}$	s_3	s_5	s_7	$s_5 \bar{F}$
Δ	L_2				L_∞			
$1/32$	3.59	6.38	6.74	0.80	1.80	5.50	6.55	0.58
$1/64$	2.06	5.69	5.91	0.77	1.24	3.25	4.17	0.62
$1/128$	1.42	4.21	5.63	0.72	0.92	1.62	2.96	0.51

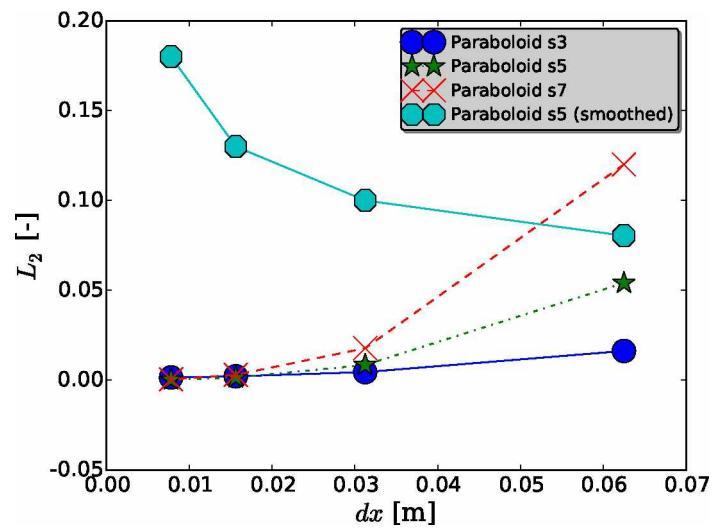
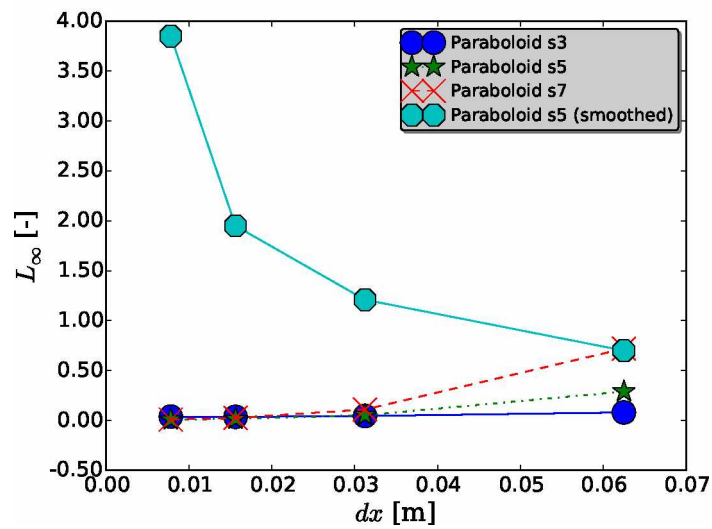
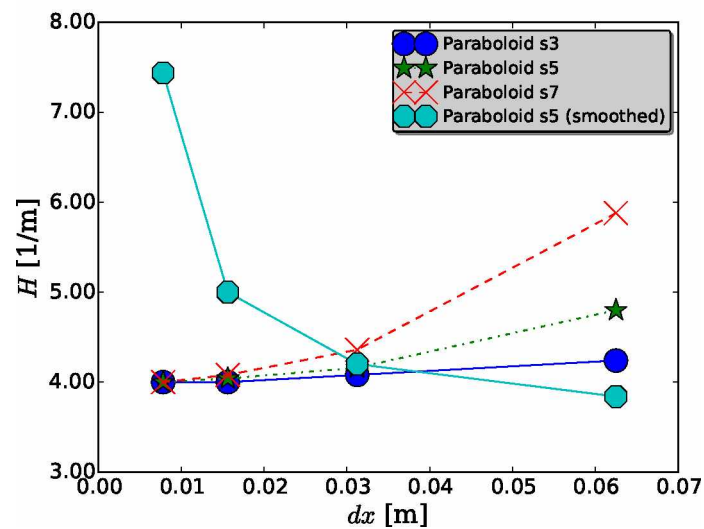
(a) L_2 (b) L_∞ (c) H

Figure 4.32: L_2 and L_∞ error norms and mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m] computed with the Paraboloid method, computational stencil sizes 3, 5 and 7, and smoothed color function for the stencil size 5.

4.1.4.6 Curvature analysis: sphere. Shirani and 27 Cells

For the Shirani based discretization and the 27 Cells discretization, the comparison between the use or not of the smoothed color function is presented in Fig. 4.33. For both methods the smaller errors are with the use of the smoothed color function.

The L_2 and L_∞ convergence errors are presented in Tab. 4.16. For $\Delta = 1/128$, the Shirani method with the smoothed color function has L_2 convergence order bigger than 1 and smaller than 1 for the L_∞ , and the 27 Cells method has order smaller than 1.

Table 4.16: Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Shirani and the 27 Cells methods with and without the smoothed color function

	Shirani	27 Cells	Shirani	27 Cells
Δ	L_2		L_∞	
$1/32$	1.32	0.86	0.53	0.34
$1/64$	0.93	0.85	0.36	0.40
$1/128$	0.72	0.67	0.43	0.45
	Smoothed color function			
$1/32$	2.37	2.14	1.64	1.50
$1/64$	1.96	1.38	1.40	0.70
$1/128$	1.18	0.81	0.43	0.44

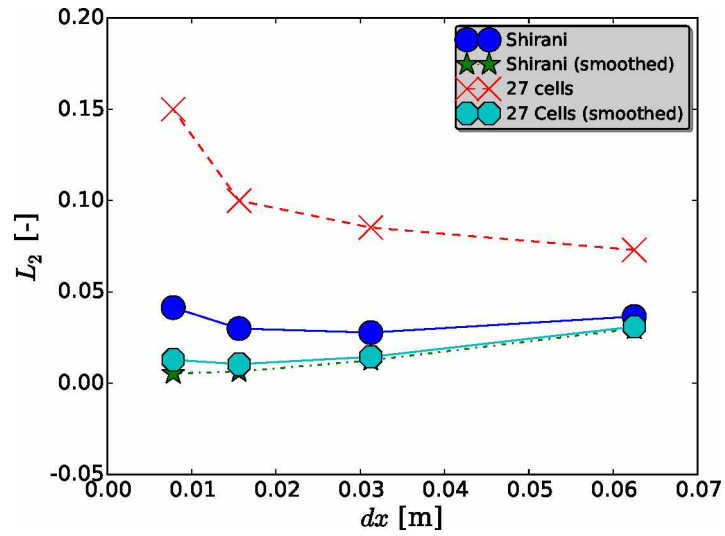
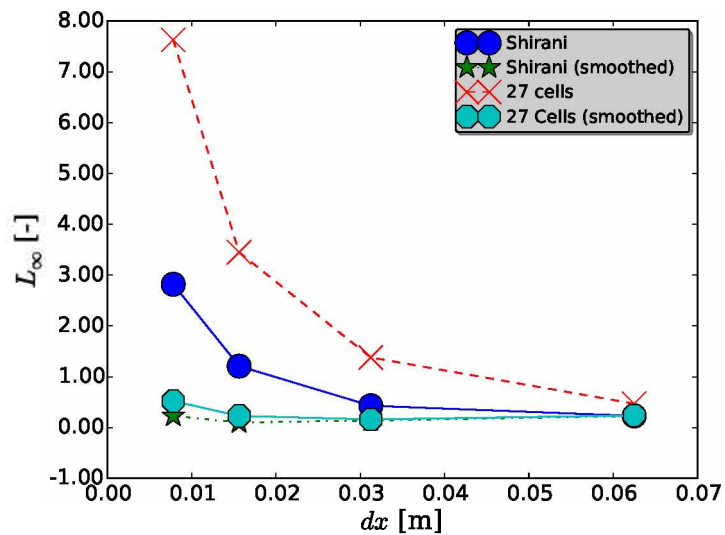
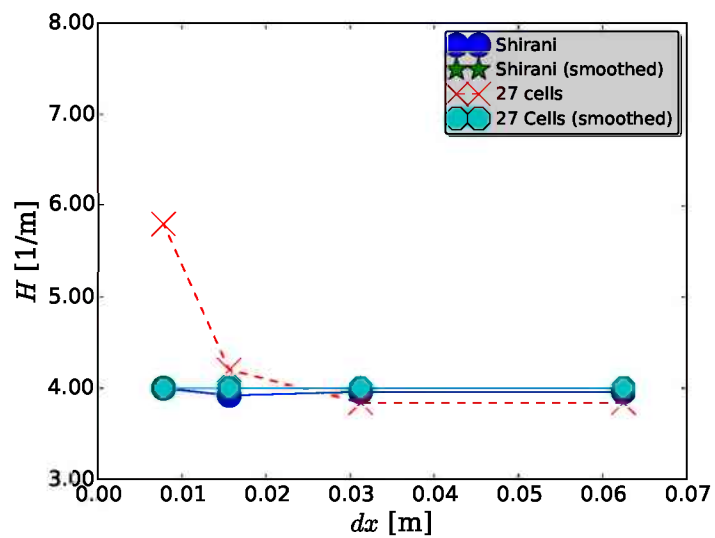
(a) L_2 (b) L_∞ (c) H

Figure 4.33: L_2 and L_∞ error norms and mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m] computed with the Shirani and 27 Cells methods with and without the smoothed color function.

4.1.4.7 Curvature analysis: sphere. Height function method

The Height Function method is coupled with the Shirani, the Paraboloid and the 27 Cells curvature computation methods, with the results presented in Fig. 4.34. The couple with the Paraboloid method leads to smaller errors for the finer meshes. The convergence error is presented in Tab. 4.18.

The quantity of inconsistent points on the Height Function method (the points that requires an auxiliary method) for each mesh resolution are presented on Tab 4.17. For the coarse mesh used most of the points are inconsistent.

Table 4.17: Quantity of inconsistent points in relation to the mesh size for the Height Function method in a sphere with $r = 0.25$ [m]

Δ	1/16	1/32	1/64	1/128
n [%]	72	17	11	10

Table 4.18: Convergence ratio for the L_2 and the L_∞ error norms for the curvature of a sphere with radius $r = 0.25$, computed with the Height Function method coupled with the Shirani, the Paraboloid and the 27 Cells methods.

		%	HF Shirani	HF Paraboloid	HF 27 Cells
Δ			L_2		
$1/32$	17		4.24	12.54	4.19
$1/64$	11		2.73	6.17	2.96
$1/128$	10		2.33	4.72	2.55
			L_∞		
$1/32$	17		1.61	5.35	1.50
$1/64$	11		1.76	3.33	1.79
$1/128$	10		1.33	2.42	1.32

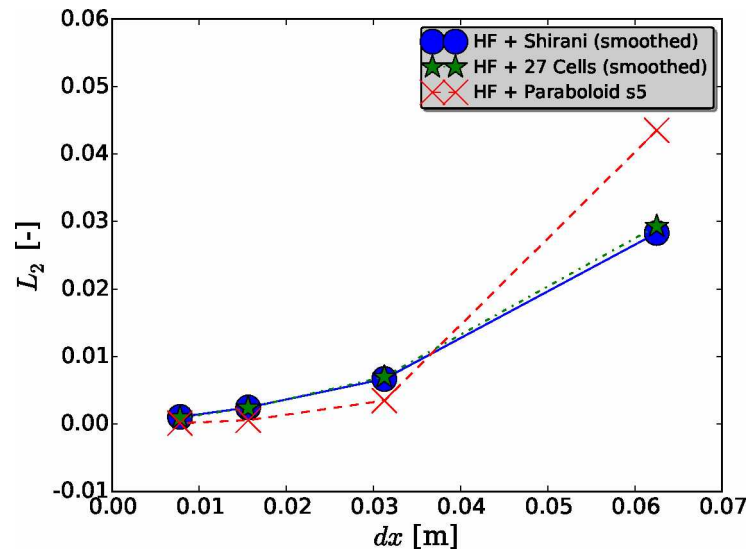
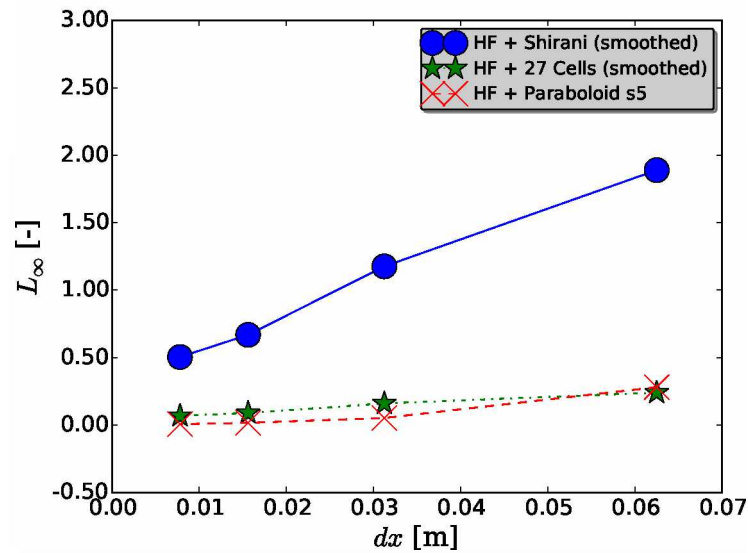
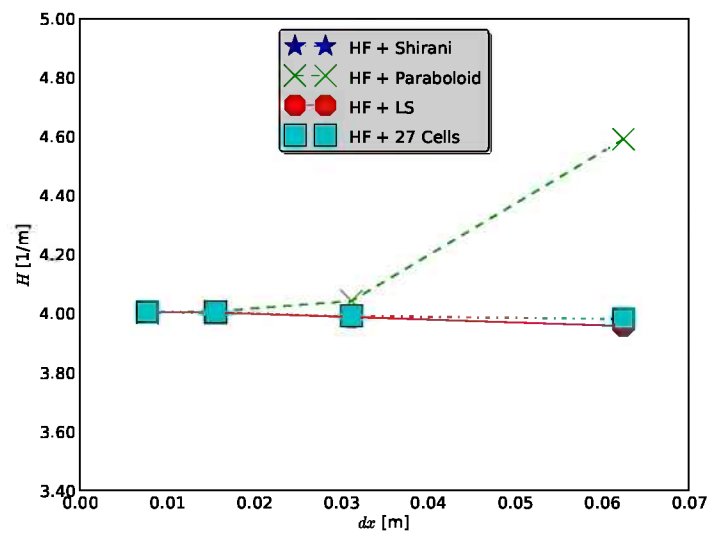
(a) L_2 (b) L_∞ (c) H

Figure 4.34: L_2 and L_∞ error norms and mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m] computed with the Height Function method.

4.1.4.8 Curvature analysis: sphere. Comparison between different curvature computation methods

After the separate analysis of the curvature computation methods and the selection of a proper configuration for each one of them, they are compared between each other. The results are presented in Fig. 4.35 to 4.37.

For the L_2 error norm, the 27 Cells discretization (smoothed) presents bigger errors and starts to diverge with the finer mesh and the Paraboloid method presents smaller errors for the more refined meshes. For the L_∞ , the error with the Shirani method increase for $\Delta = 1/128$, and the Height Function coupled with the Shirani presents the bigger errors. The Height Function method suffers from the influence of the method it is coupled with, particularly in the L_∞ error norm.

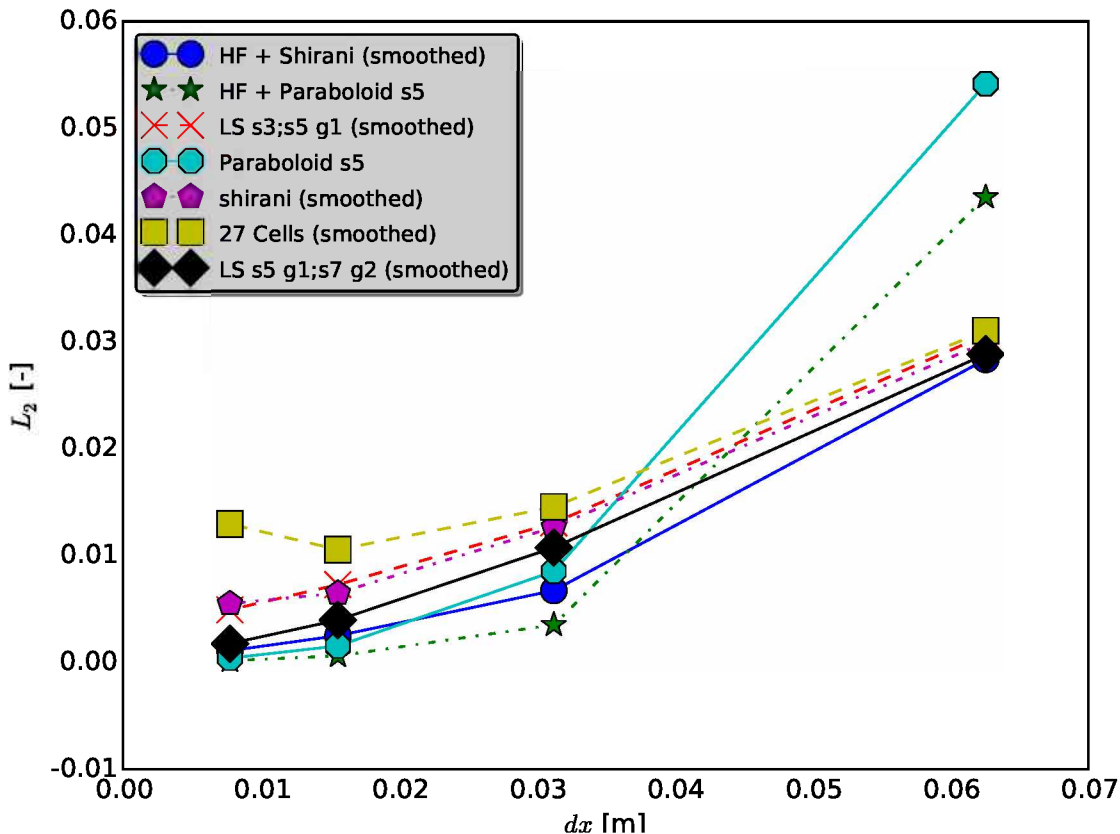


Figure 4.35: L_2 error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m]. Comparison between different curvature computation methods.

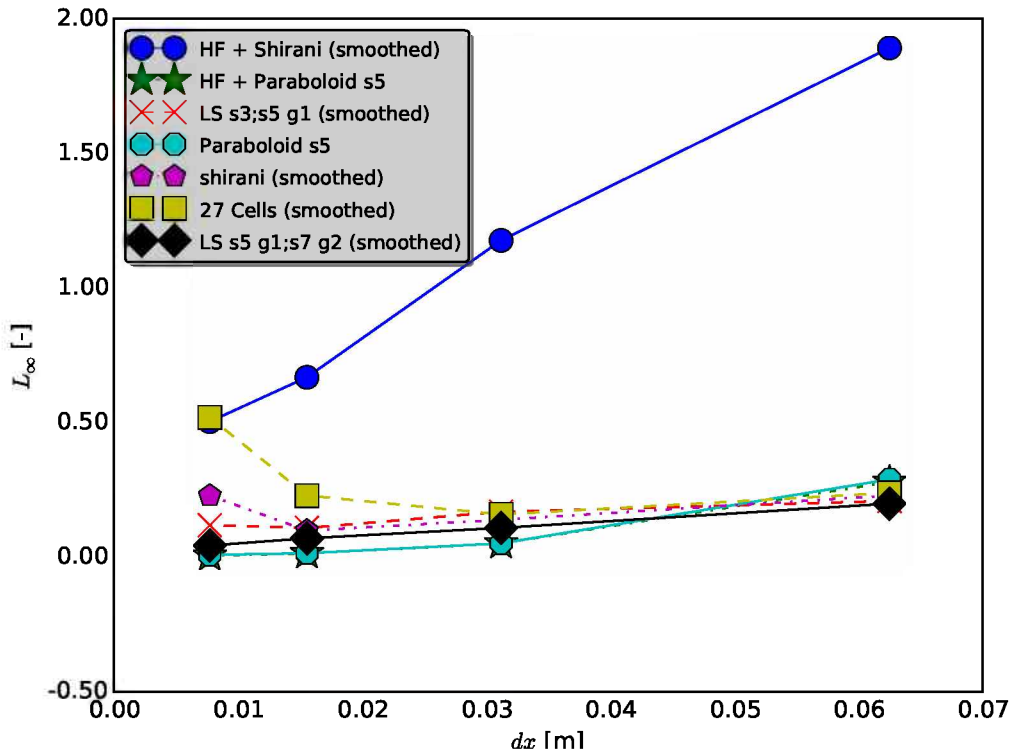


Figure 4.36: L_∞ error norm as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m]. Comparison between different curvature computation methods.

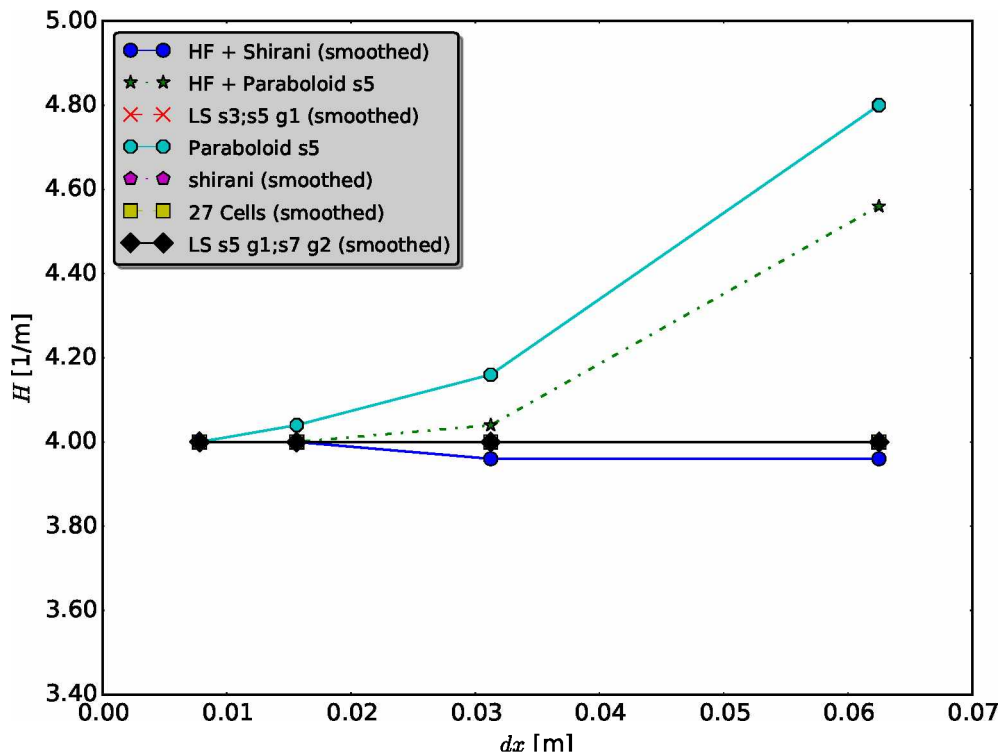


Figure 4.37: Mean curvature H as a function of the mesh refinement for the curvature of a sphere with $r = 0.25$ [m]. Comparison between different curvature computation methods.

4.1.4.9 Curvature analysis: ellipsoid

An analysis of the Height function method coupled with the other methods is performed, with the results presented in Fig. 4.38. The Height Function coupled with the Paraboloid presents smaller errors. The number of inconsistent points with the mesh refinement is presented in Tab. 4.19. Its behavior is similar to the sphere test case but, interestingly, the number of inconsistent point is smaller for the refined meshes used.

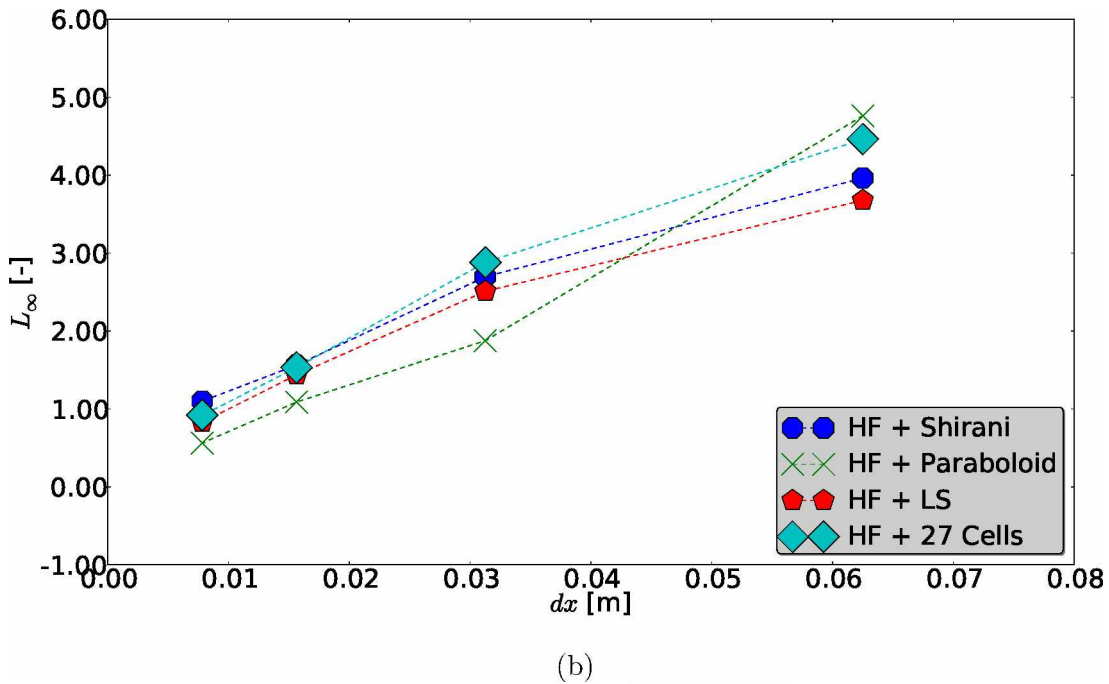
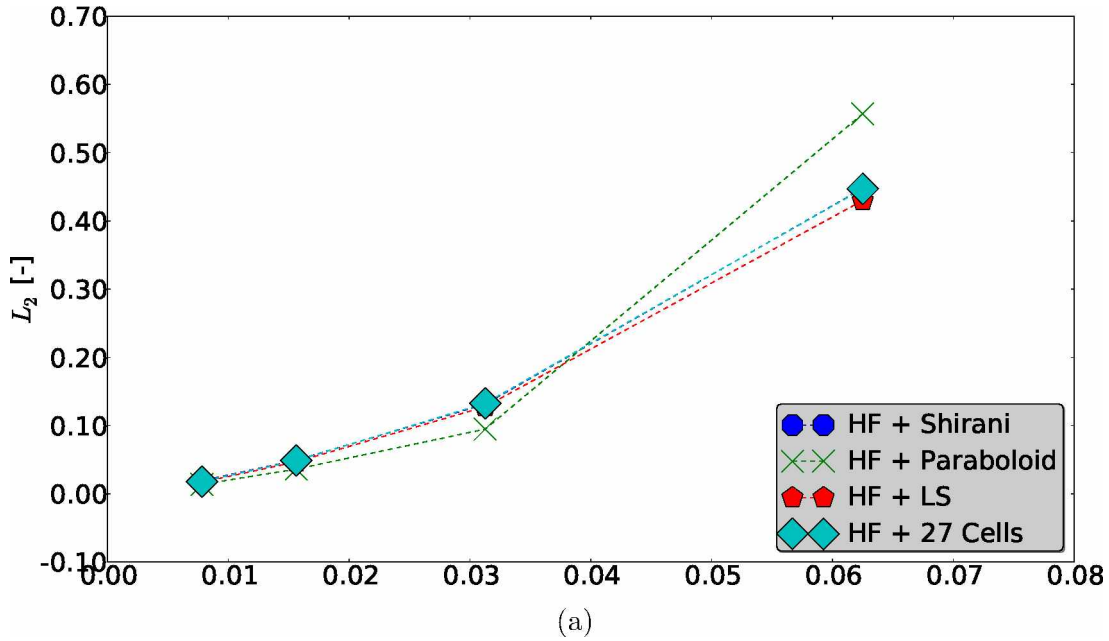


Figure 4.38: a) L_2 error norm x mesh refinement and b) L_∞ error norm x mesh refinement of the curvature for an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$. Comparison between the Height Function coupled with different methods

Table 4.19: Quantity of inconsistent points as a function of the mesh refinement for the Height Function method. Curvature computation on an ellipsoid

Δ	1/16	1/32	1/64	1/128
n [%]	72	17	10	8

From the previous analysis on the sphere, the tests with the ellipsoid are performed with the following configurations for each method. For the Paraboloid method, it is chosen the stencil of size 5^3 and non-smoothed color function. The Shirani and the 27 Cells discretization are tested with the smoothed color function. The Least Squares is tested with a stencil of size $s = 5$ for the normal and smoothed color function and $s = 3$ for the curvature.

Figure 4.39 and 4.40 presents the results and Tab. 4.20 the convergence error. The 27 Cells discretization alone presents bigger errors for the curvature computation on the ellipsoid, as in the case for the sphere, but the couple with the Height Function method leads to errors similar to the other methods. The Shirani and the Least Squares performed very similar, and presents an increase of the L_∞ error for the finer mesh. The Paraboloid and the Height Function coupled with the Paraboloid, as for the sphere, presents the smaller errors for the more refined mesh.

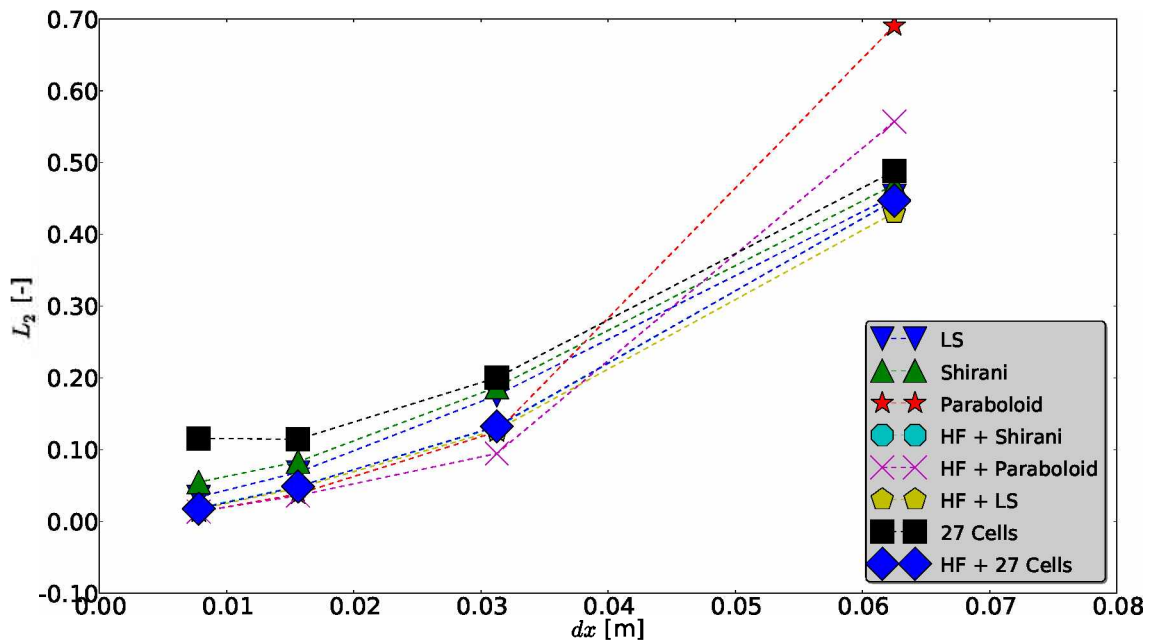


Figure 4.39: L_2 error norm as a function of the mesh refinement for the curvature computation of an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$. Comparison between different curvature computation methods

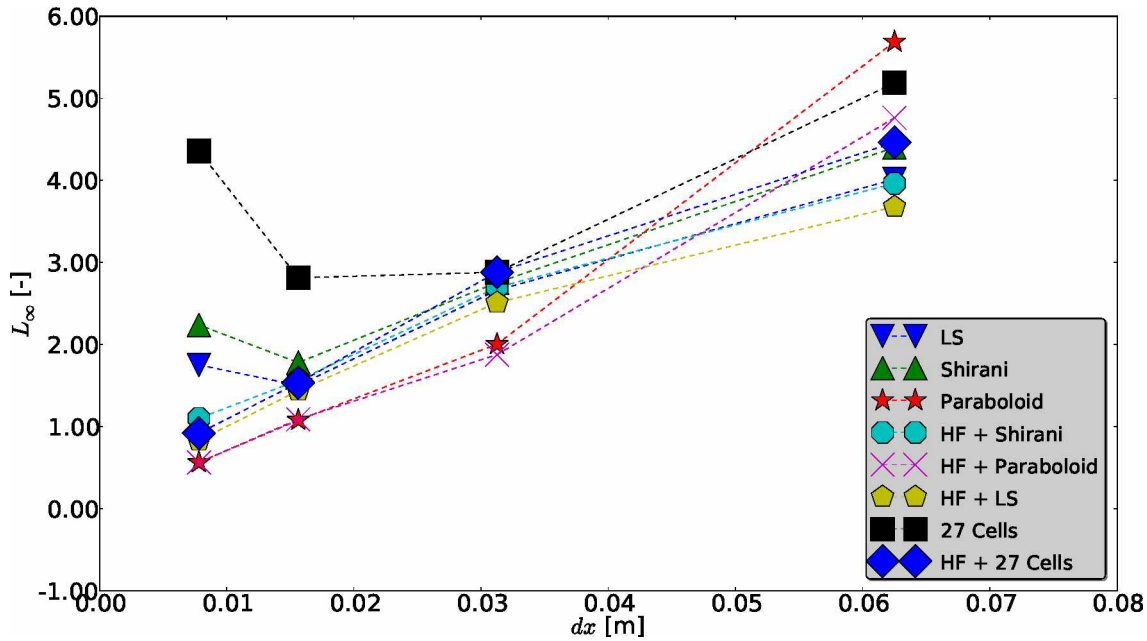


Figure 4.40: L_∞ error norm as a function of the mesh refinement for the curvature computation of an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$. Comparison between different curvature computation methods

Table 4.20: Convergence ratio for the L_2 and the L_∞ error norms for the curvature of an ellipsoid with parameters $a = 0.30$, $b = 0.25$ and $c = 0.20$.

	Paraboloid	LS	Shirani	27 Cells	HF+ Shirani	HF+ Paraboloid	HF+ LS	HF+ 27 Cells
Δ	L_2							
$1/32$	5.52	2.58	1.71	2.44	3.41	5.88	3.37	3.37
$1/64$	3.25	2.57	1.11	1.75	2.67	2.62	2.72	2.72
$1/128$	2.87	2.00	0.80	0.99	2.59	2.61	2.75	2.72
	L_∞							
$1/32$	2.84	1.51	1.12	1.80	1.47	2.54	1.47	1.55
$1/64$	1.86	1.77	0.37	1.02	1.73	1.72	1.75	1.88
$1/128$	1.91	0.86	0.47	0.65	1.42	1.93	1.73	1.67

4.1.5 Analysis of a 3D static drop in equilibrium

Following Francois et al. (2006), a series of tests for a 3D inviscid static drop in equilibrium without gravity are performed. The setup is a drop of radius $r = 2[m]$ centered in a domain $\Omega = [8]^3[m^3]$; velocity boundary conditions are free-slip (normal component of velocity is zero) and the mesh resolution is $R/h = 10$.

Francois et al. (2006) presents the results with the convolution technique and the Height function method for curvatures, and compares with results of Williams, Kothe and

Puckett (1998): BKZ method (which represents the results of the original CSF paper of Brackbill, Kothe and Zemach (1992) with a smooth volume fraction); Method I (results with convolved curvatures and step delta function); Method II (results with finite-difference normals and parabolic delta function).

In the AMR3D code, for the Height Function coupled with the Shirani method, the number of inconsistent points is 17.44%.

The exact jump in pressure across the drop is given by:

$$\Delta P_{exact} = \sigma \kappa \quad (4.13)$$

with $\kappa = 2/R$ in 3D.

The numerical jump in pressure is evaluated as:

$$\Delta P_{max} = P_{max} - P_{min} \quad (4.14)$$

The relative pressure jump error is evaluated as:

$$E(\Delta P)_{max} = \frac{|\Delta P_{max} - \Delta P_{exact}|}{\Delta P_{exact}} \quad (4.15)$$

Table 4.21 compares the error in maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ for a 3D inviscid static drop in equilibrium without gravity after one and 50 time steps with different methods to evaluate the curvature. The setup is a drop of radius $R = 2[m]$ centered in a domain $\Omega = [8]^3[m^3]$, with density ratio 10, surface tension coefficient $\sigma = 73[N/m]$ and time step constant and equal to $10^{-3}[s]$. The exact pressure difference for this case is 73.

The smaller parasitic velocity presented by Francois et al. (2006) for the first time step is with the Height Function method ($|u|_{max} = 4.02E - 3$) and with the AMR3D code the smaller parasitic velocity is with the Paraboloid method ($|u|_{max} = 1.49E - 3$). The Least Squares, the Height Function coupled with the Shirani, the Shirani and the Paraboloid curvature computation methods presents parasitic velocities on the same order of magnitude as the Height Function and the Convolution methods presented by Francois et al. (2006) for the first time step. For 50 time steps only the Height Function and the Least Squares are on the same order as the Height Function from Francois et al. (2006). For the errors in the relative pressure jump, the Least Squares method presents smaller errors for this case.

With the density scaling, $|\mathbf{u}|_{max}$ is bigger than without the density scaling, but the

relative pressure jump error $E(\Delta P)_{max}$ is smaller. It is interesting to note that the density scaling affect significantly the result for the exact curvature.

Table 4.21: Error in maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after one and 50 time steps for a 3D inviscid static drop in equilibrium with different curvature estimates.(Uniform mesh)

Curvature Method	$ \mathbf{u} _{max}$		$E(\Delta P)_{max}$	
	$(t = \Delta t)$	$(t = 50\Delta t)$	$(t = \Delta t)$	$(t = 50\Delta t)$
Francois et al. (2006)				
Convolution	$4.87E - 03$	$1.63E - 01$	–	–
Heigh Function	$4.0E - 03$	$4.02E - 02$	–	–
Williams, Kothe and Puckett (apud FRANCOIS et al., 2006)				
BKZ	$3.49E - 01$	2.55	–	–
Method I	$1.03E - 01$	$8.46E - 01$	–	–
Method II	$8.55E - 02$	$3.86E - 01$	–	–
AMR3D code with density scaling				
Least Squares	$1.89E - 03$	$7.86E - 02$	$4.12E - 03$	$3.60E - 03$
HFS	$1.50E - 03$	$6.53E - 02$	$2.43E - 02$	$2.28E - 02$
Shirani	$8.24E - 03$	$2.95E - 01$	$6.86E - 02$	$2.54E - 02$
27 Cells	$2.25E - 02$	$4.57E - 01$	$2.98E - 01$	$6.50E - 02$
Paraboloid	$1.49E - 03$	$4.57E - 01$	$1.96E - 02$	$6.50E - 02$
Exact	$1.46E - 03$	$7.16E - 02$	$9.72E - 03$	$9.67E - 03$
AMR3D code without density scaling				
Least Squares	$1.69E - 03$	$6.92E - 02$	$1.51E - 02$	$1.32E - 02$
HFS	$1.34E - 03$	$5.55E - 02$	$3.16E - 02$	$2.95E - 02$
Shirani	$8.43E - 03$	$2.98E - 01$	$1.08E - 01$	$3.61E - 02$
27 Cells	$2.35E - 02$	$4.62E - 01$	$3.42E - 01$	$7.07E - 02$
Paraboloid	$2.61E - 04$	$1.18E - 02$	$3.02E - 02$	$3.02E - 02$
Exact	$1.35E - 06$	$1.12E - 06$	$5.55E - 05$	$3.25E - 10$

4.1.5.1 Effect of fluid properties and integration time step

Table 4.22 presents the effect of the viscosity ratio with and without the density scaling method. Similar to Francois et al. (2006), there is no influence of the viscosity ratio neither in the parasitic currents nor in the relative pressure jump error. For all results with the computed curvature, the density scaling method leads to worse results for $|\mathbf{u}|_{max}$ and better for $E(\Delta P)_{max}$.

The parasitic current increases after 100 time steps for the computed curvature methods with and without the density scaling and for the exact curvature with the density scaling. There is no significant difference between the values obtained with the Least Squares and the

Heigh Function + Shirani. For the error in pressure, the density scaling improves the results for the cases with the computed curvature and no effect is observed between the first and the last time step. The density scaling harms the result for the exact curvature considerably; again no difference is observed for the pressure error between the two curvature computation methods used.

Table 4.22: Effect of the viscosity ratio on the error of the maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 and 100 time steps for the viscous drop in equilibrium. The fluid density ration is 10^3 , with density inside ($\rho_1 = 1[kg/m^3]$) and density outside ($\rho_2 = 10^{-3}[kg/m^3]$) and $\sigma = 73.0[N/m]$. The drop has radius $R = 2[m]$ and is centered in a domain $\Omega = [8]^3[m^3]$ with mesh 40^3 ($R/h = 10$). The time step is constant and equal to $10^{-6}[s]$. The fluid viscosity inside the drop ($\mu_1 = 10^{-2}[Pa.s]$) and outside the viscosity is allowed to vary. (Uniform mesh)

μ_1/μ_2	Curvature Method					
	HFS		LS		Exact	
	(ts=1)	(ts=100)	(ts=1)	(ts=100)	(ts=1)	(ts=100)
	$ \mathbf{u} _{max}$ (With density scaling)					
1	$2.29E - 05$	$2.27E - 03$	$2.28E - 05$	$2.25E - 03$	$2.20E - 05$	$2.18E - 03$
10	$2.29E - 05$	$2.27E - 03$	$2.28E - 05$	$2.25E - 03$	$2.20E - 05$	$2.18E - 03$
100	$2.28E - 05$	$2.27E - 03$	$2.28E - 05$	$2.25E - 03$	$2.20E - 05$	$2.18E - 03$
	$ \mathbf{u} _{max}$ (Without density scaling)					
1	$1.46E - 05$	$1.44E - 03$	$1.89E - 05$	$1.87E - 03$	$3.60E - 09$	$1.86E - 09$
10	$1.46E - 05$	$1.44E - 03$	$1.89E - 05$	$1.87E - 03$	$3.60E - 09$	$1.86E - 09$
100	$1.46E - 05$	$1.44E - 03$	$1.89E - 05$	$1.87E - 03$	$3.60E - 09$	$1.86E - 09$
	$E(\Delta P)_{max}$ (With density scaling)					
1	$2.08E - 02$	$2.08E - 02$	$2.21E - 03$	$2.21E - 03$	$1.43E - 02$	$1.43E - 02$
10	$2.08E - 02$	$2.08E - 02$	$2.21E - 03$	$2.21E - 03$	$1.43E - 02$	$1.43E - 02$
100	$2.08E - 02$	$2.08E - 02$	$2.21E - 03$	$2.21E - 03$	$1.43E - 02$	$1.43E - 02$
	$E(\Delta P)_{max}$ (Without density scaling)					
1	$3.13E - 02$	$3.13E - 02$	$1.35E - 02$	$1.35E - 03$	$2.44E - 07$	$2.09E - 12$
10	$3.13E - 02$	$3.13E - 02$	$1.35E - 02$	$1.35E - 03$	$2.44E - 07$	$2.09E - 12$
100	$3.13E - 02$	$3.13E - 02$	$1.35E - 02$	$1.35E - 03$	$2.44E - 07$	$2.09E - 12$

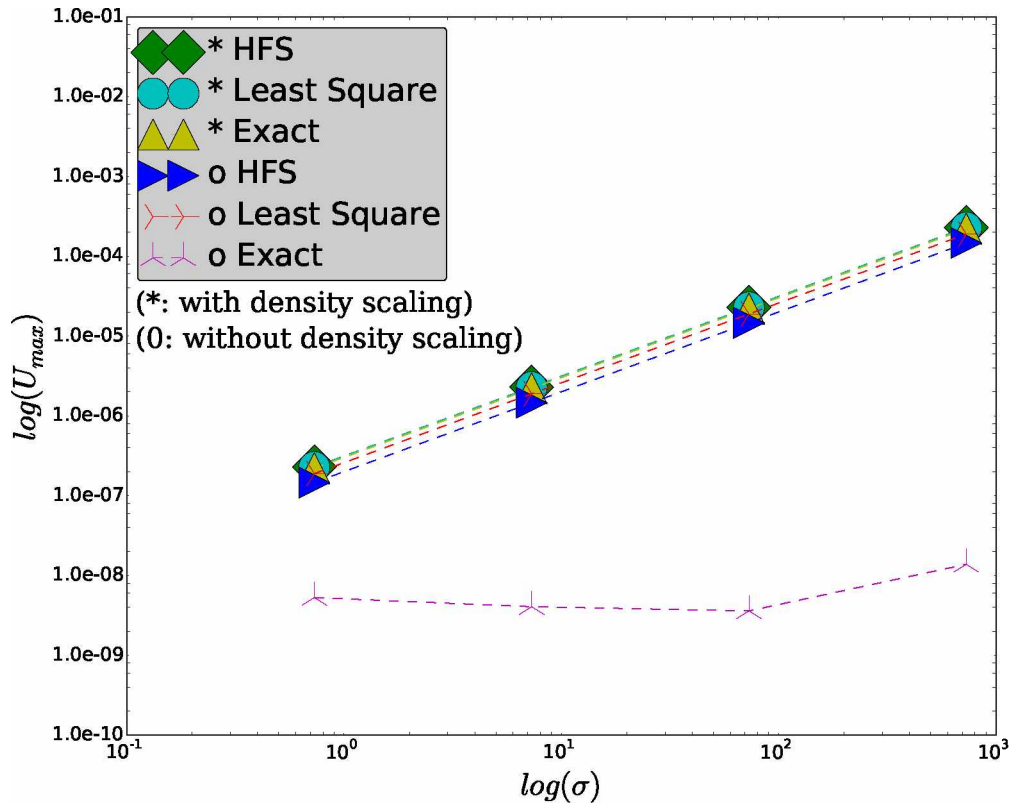
The effect of the surface tension coefficient is presented in Tab. 4.23 and illustrated in Fig. 4.41. The effect of the time step is presented in Tab. 4.24 and illustrated in Fig. 4.42. Parasitic velocities are directly proportional to the surface tension coefficient and the time step, but the pressure jump error suffers no influence. The Height Function method and the Least Squares presents similar results for the parasitic currents; for the relative pressure jump, the Least Squares method presents errors that are one order lower. Again the density scaling method lead to worse results for $|\mathbf{u}|_{max}$ and better for $E(\Delta P)_{max}$.

Table 4.23: Effect of the surface tension coefficient on the error of the maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. The fluid density ration is 10^3 , with density inside ($\rho_1 = 1[kg/m^3]$) and density outside ($\rho_2 = 10^{-3}[kg/m^3]$). The drop has radius $R = 2[m]$ and is centered in a domain $\Omega = [8]^3[m^3]$ with mesh 40^3 ($R/h = 10$). The time step is constant and equal to $10^{-6}[s]$. The viscosity ratio is 10, with $\mu_1 = 10^{-2}[Pa.s]$ and $\mu_2 = 10^{-3}[Pa.s]$. (Uniform mesh)

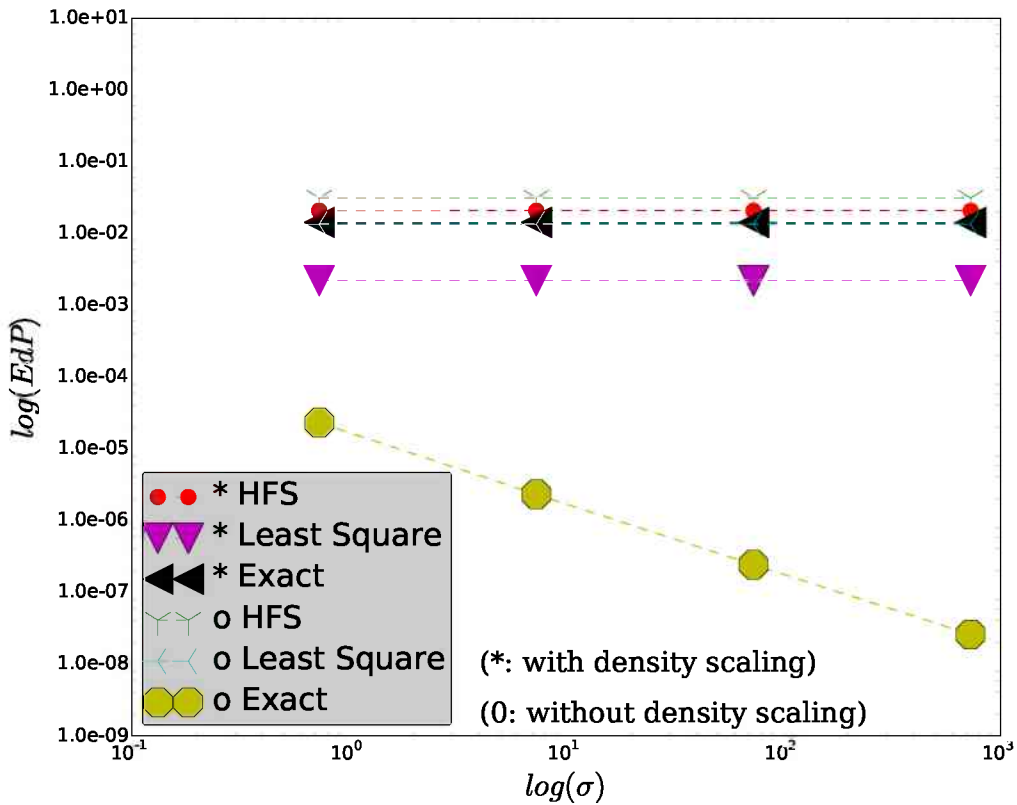
Curvature Method						
HFS			LS		Exact	
σ	$ \mathbf{u} _{max}$	$E(\Delta P)_{max}$	$ \mathbf{u} _{max}$	$E(\Delta P)_{max}$	$ \mathbf{u} _{max}$	$E(\Delta P)_{max}$
With density scaling						
0.73	$2.29E - 07$	$2.09E - 02$	$2.28E - 07$	$2.21E - 03$	$2.20E - 07$	$1.43E - 02$
7.3	$2.29E - 06$	$2.08E - 02$	$2.28E - 06$	$2.21E - 03$	$2.20E - 06$	$1.43E - 02$
73.0	$2.29E - 05$	$2.08E - 02$	$2.28E - 05$	$2.21E - 03$	$2.20E - 05$	$1.43E - 02$
730.0	$2.29E - 04$	$2.08E - 02$	$2.28E - 04$	$2.21E - 03$	$2.20E - 04$	$1.43E - 02$
Without density scaling						
0.73	$1.46E - 07$	$3.13E - 02$	$1.89E - 07$	$1.35E - 02$	$5.29E - 09$	$2.32E - 05$
7.3	$1.46E - 06$	$3.13E - 02$	$1.89E - 06$	$1.35E - 02$	$4.06E - 09$	$2.33E - 06$
73.0	$1.46E - 05$	$3.13E - 02$	$1.89E - 05$	$1.35E - 02$	$3.60E - 09$	$2.44E - 07$
730.0	$1.46E - 04$	$3.13E - 02$	$1.89E - 04$	$1.35E - 02$	$1.38E - 08$	$2.61E - 08$

Table 4.24: Effect of the time step magnitude on the error of the maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. The fluid density ration is 10^3 , with density inside ($\rho_1 = 1[kg/m^3]$) and density outside ($\rho_2 = 10^{-3}[kg/m^3]$). The drop has radius $R = 2[m]$ and is centered in a domain $\Omega = [8]^3[m^3]$ with mesh 40^3 ($R/h = 10$). The viscosity ratio is 10, with $\mu_1 = 10^{-2}[Pa.s]$ and $\mu_2 = 10^{-3}[Pa.s]$ and $\sigma = 730.0[N/m]$. (Uniform mesh)

Curvature Method						
HFS			LS		Exact	
Δt	$ \mathbf{u} _{max}$	$E(\Delta P)_{max}$	$ \mathbf{u} _{max}$	$E(\Delta P)_{max}$	$ \mathbf{u} _{max}$	$E(\Delta P)_{max}$
With density scaling						
10^{-3}	$1.07E - 01$	$2.07E - 02$	$1.06E - 01$	$2.06E - 03$	$1.03E - 01$	$1.44E - 02$
10^{-4}	$2.29E - 02$	$2.08E - 02$	$2.28E - 02$	$2.17E - 03$	$2.20E - 02$	$1.43E - 02$
10^{-5}	$2.29E - 03$	$2.08E - 02$	$2.28E - 03$	$2.20E - 03$	$2.20E - 03$	$1.43E - 02$
10^{-6}	$2.29E - 04$	$2.08E - 02$	$2.28E - 04$	$2.21E - 03$	$2.20E - 04$	$1.43E - 02$
Without density scaling						
10^{-3}	$6.81E - 02$	$3.12E - 02$	$8.85E - 02$	$1.33E - 02$	$2.48E - 03$	$2.26E - 06$
10^{-4}	$1.46E - 02$	$3.12E - 02$	$1.89E - 02$	$1.35E - 02$	$1.26E - 04$	$5.51E - 07$
10^{-5}	$1.46E - 03$	$3.13E - 02$	$1.89E - 03$	$1.35E - 02$	$1.40E - 06$	$6.65E - 08$
10^{-6}	$1.46E - 04$	$3.13E - 02$	$1.89E - 04$	$1.35E - 02$	$1.38E - 08$	$2.61E - 08$

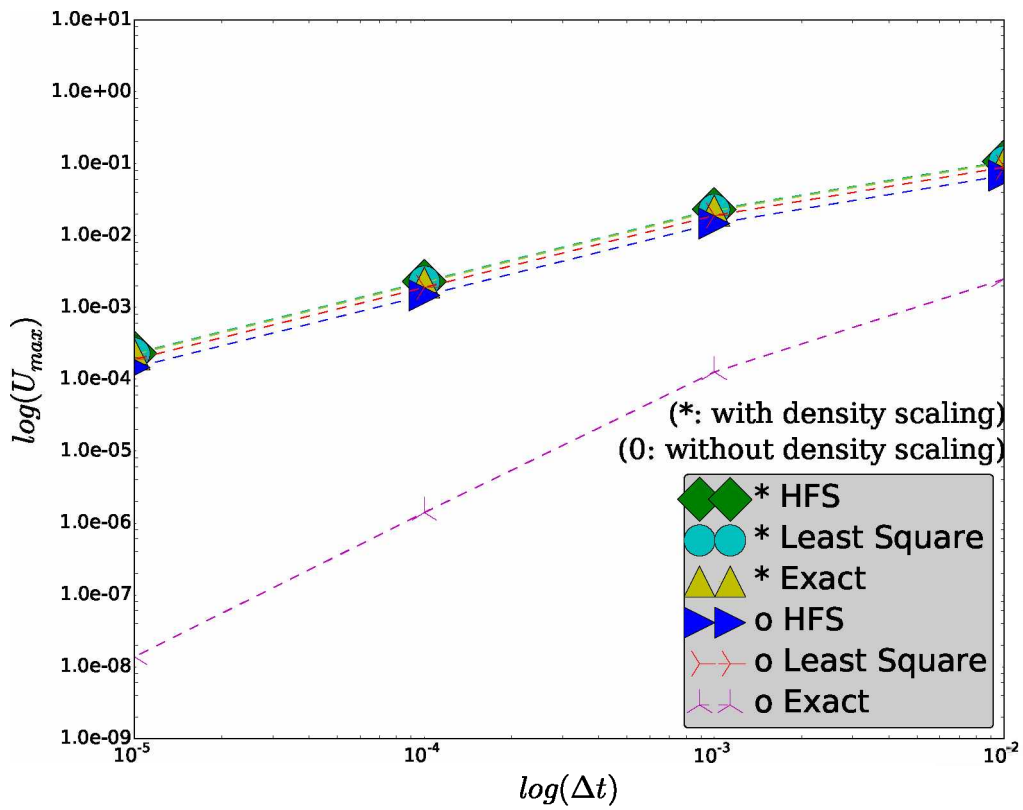


(a) $\log(\sigma)$ vs $\log(|\mathbf{u}|_{max})$

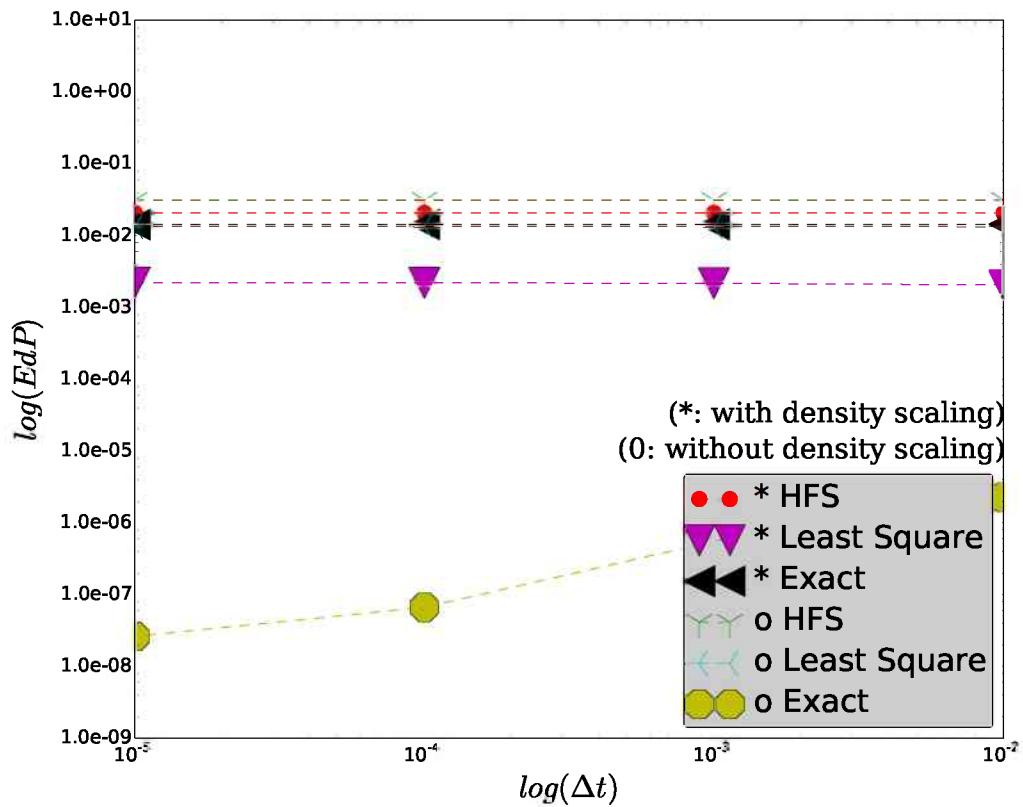


(b) $\log(\sigma)$ vs $\log(E(\Delta P)_{max})$

Figure 4.41: Effect of the surface tension coefficient on the error of the maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. Data from Tab. 4.23



(a) $\log(\Delta t)$ vs $\log(|\mathbf{u}|_{max})$



(b) $\log(\Delta t)$ vs $\log(E(\Delta P)_{max})$

Figure 4.42: Effect of the time step magnitude on the error of the maximum velocity $|\mathbf{u}|_{max}$ and relative pressure jump $E(\Delta P)_{max}$ after 1 time step for the viscous drop in equilibrium. Data from Tab. 4.24

4.1.6 Parasitic current analysis

A parasitic current analysis, proposed by Popinet (2009), is performed with the following curvature computation methods: i) Paraboloid (stencil $s = 5$), ii) Least Squares (stencil $s = 5$ for the normal and $s = 3$ for the curvature, with smoothed color function to compute the normal vector), iii) Height Function coupled with the Shirani (smoothed) and iv) Height Function coupled with the Paraboloid (stencil $s = 5$).

A circular interface centred on the top-left-south corner of a unit cubic domain is considered. Symmetry conditions are applied on the top, left and south boundaries so that only a quarter of the droplet is simulated on a $32 \times 32 \times 32$ grid. The diameter of the droplet is $d = 0.8[m]$. The velocity scale for the inviscid problem is (POPINET, 2009)

$$U_\sigma = \sqrt{\frac{\sigma}{\rho d}} \quad (4.16)$$

with ρ the constant density. For a viscous fluid the timescale is defined as

$$T_\mu = \frac{d^2}{\mu} \quad (4.17)$$

with μ the kinematic viscosity.

The parameters of the simulation are:

- $g = 0[m/s^2]$.

- $\rho_c = \rho_d = 1.0[kg/m^3]$.

where: ρ_c = density at continuous phase; ρ_d =density at disperse phase

- $\mu_c = \mu_d = 1.0[kg/m.s]$

where: μ_c = viscosity at continuous phase; μ_d =viscosity at disperse phase

- $La = \frac{\sigma \rho d}{\mu^2}$

Three tests are performed:

1. $La = 120$ ($\sigma = 150[N/m]$)
2. $La = 1200$ ($\sigma = 1500[N/m]$)
3. $La = 12000$ ($\sigma = 15000[N/m]$, $d_t = 1.0E-6[s]$)

Figures 4.43 to 4.45 illustrates the evolution of the root-mean-square (RMS) velocity with time for the range of Laplace number indicated in the legend. Time and velocity are made non-dimensional using T_μ and U_σ as reference scales, respectively.

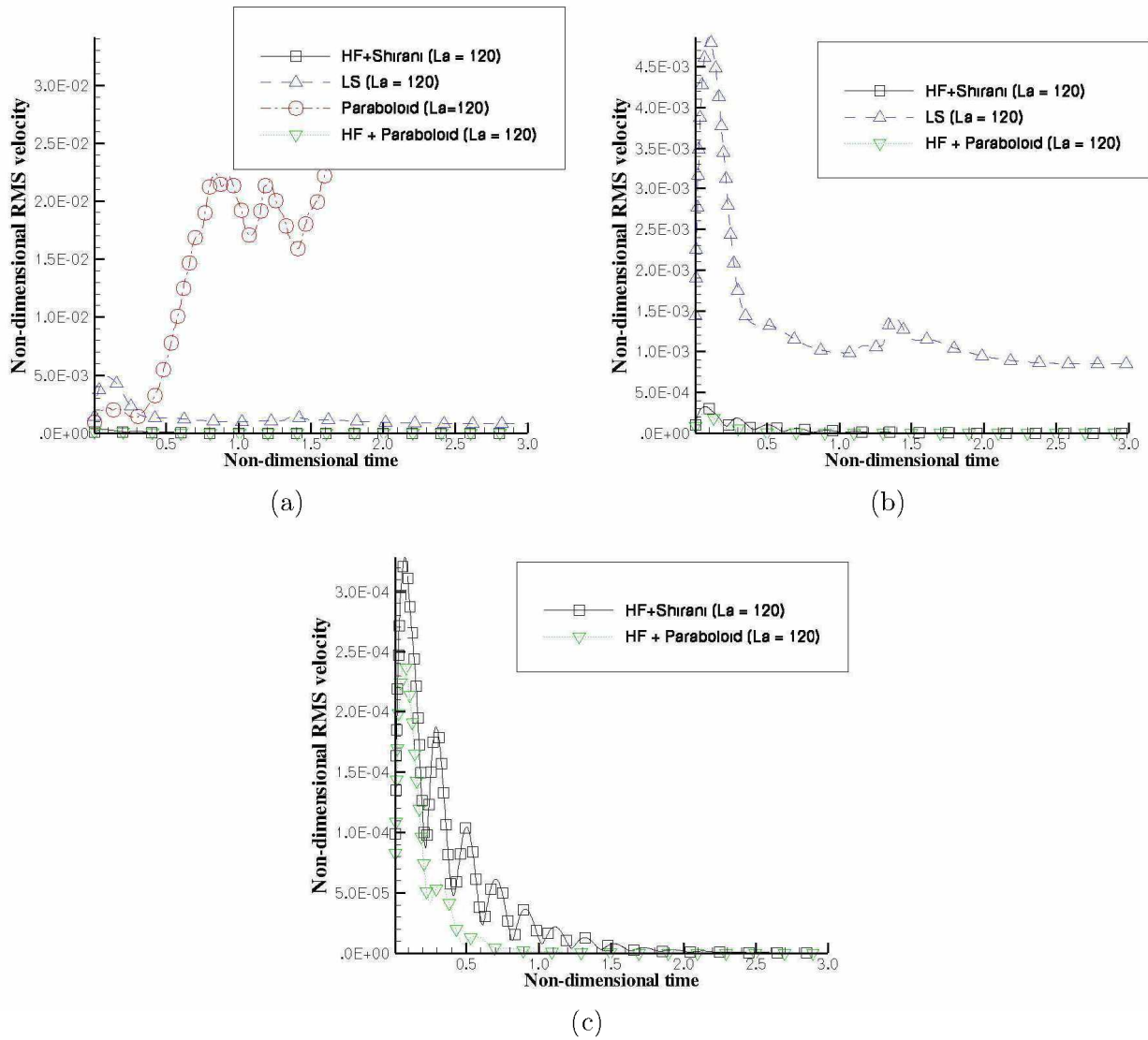


Figure 4.43: Evolution of the RMS velocity around a circular droplet in theoretical equilibrium for $La = 120$. a) Result comparison with different curvature computation methods b) Result comparison excluding the Paraboloid method c) Result comparison between the Height function coupled with the Shirani and coupled with the Paraboloid method. Time and velocity are made non-dimensional using T_μ and U_σ as reference scales, respectively.

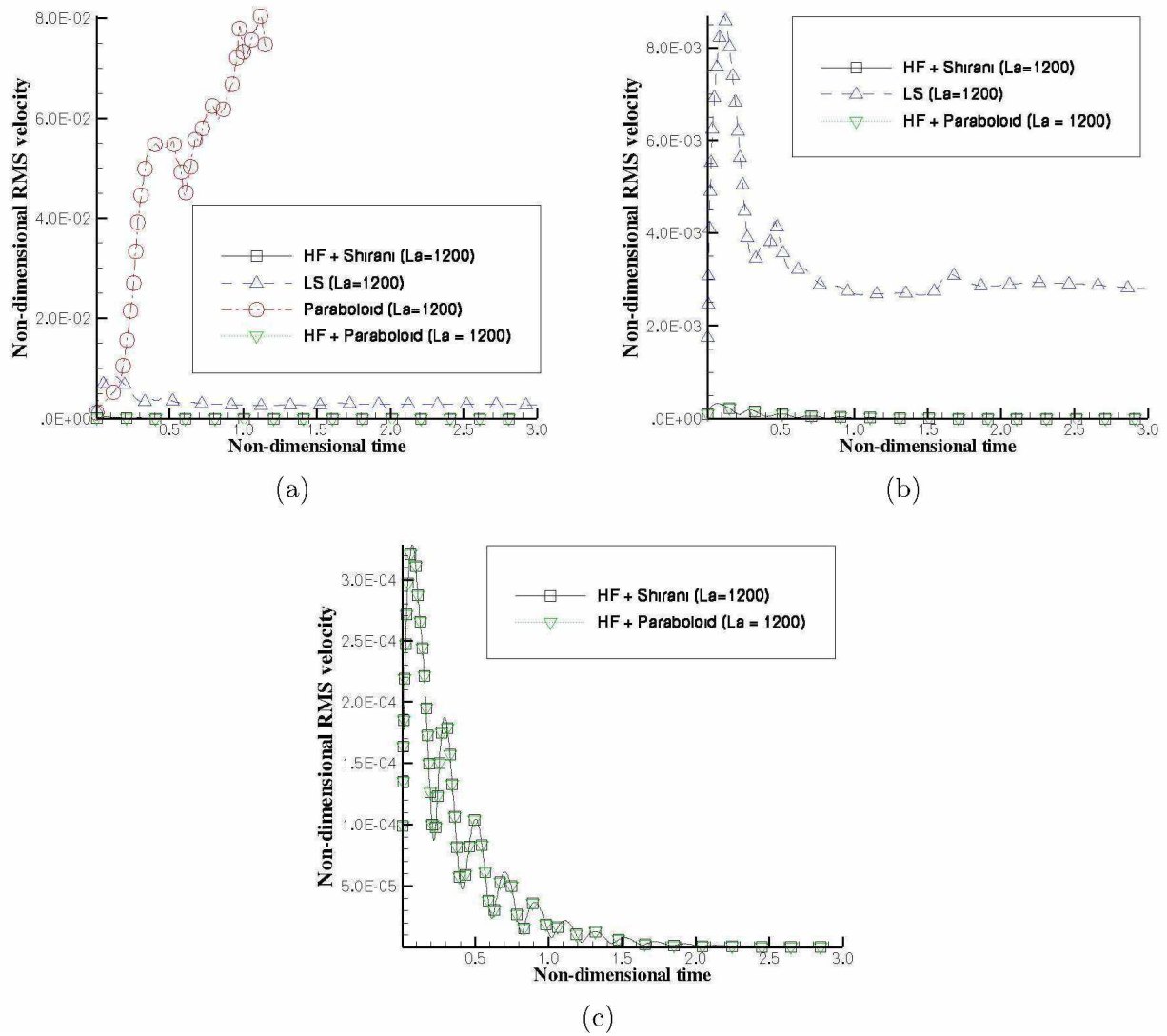


Figure 4.44: Evolution of the RMS velocity around a circular droplet in theoretical equilibrium for $La = 1200$. a) Result comparison with different curvature computation methods b) Result comparison excluding the Paraboloid method c) Result comparison between the Height function coupled with the Shirani and coupled with the Paraboloid method. Time and velocity are made non-dimensional using T_μ and U_σ as reference scales, respectively.

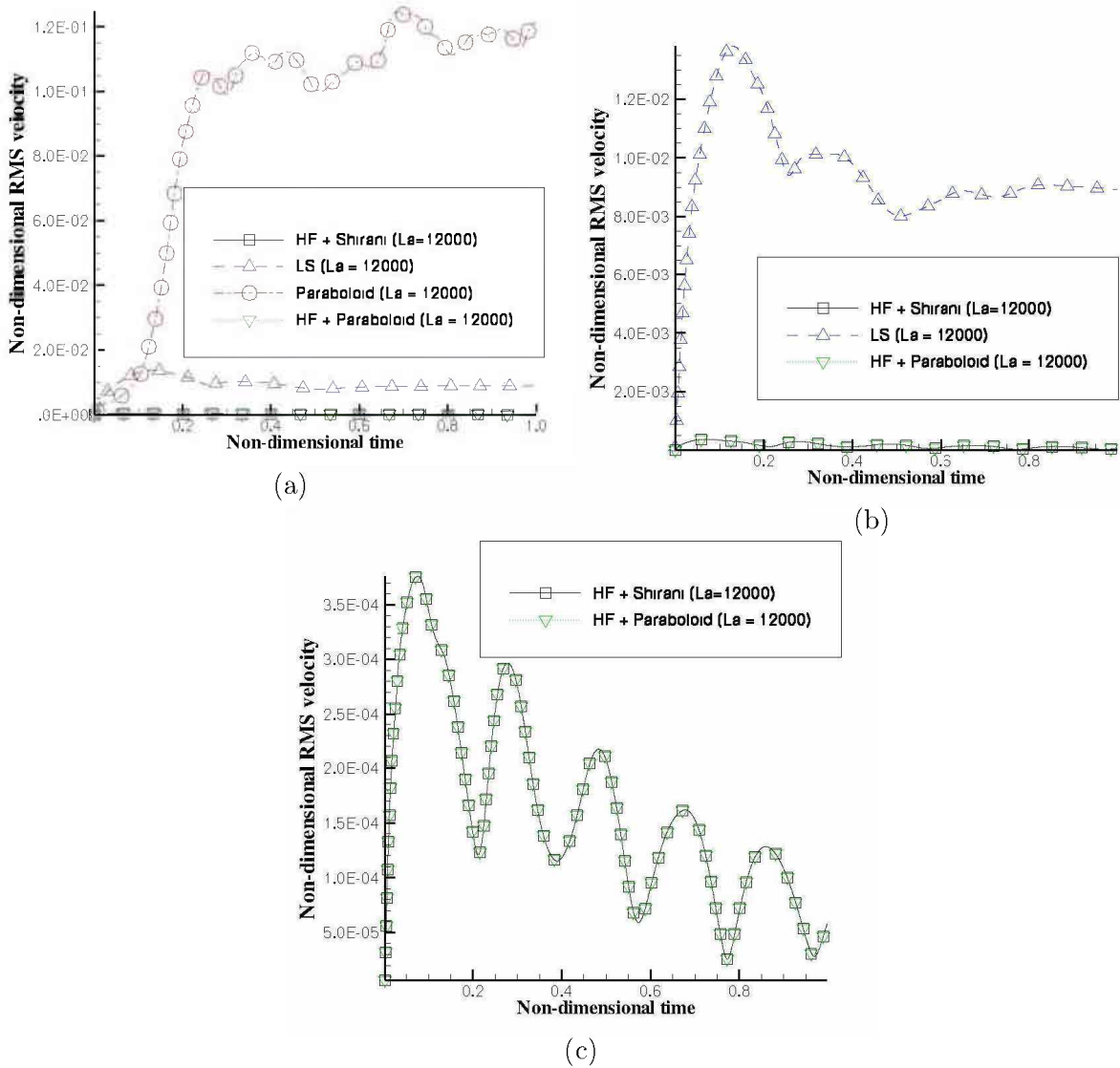


Figure 4.45: Evolution of the RMS velocity around a circular droplet in theoretical equilibrium for $La = 12000$. a) Result comparison with different curvature computation methods b) Result comparison excluding the Paraboloid method c) Result comparison between the Height function coupled with the Shirani and coupled with the Paraboloid method. Time and velocity are made non-dimensional using T_μ and U_σ as reference scales, respectively.

The results shows that with the use of the paraboloid method alone to compute the curvature, the method becomes very unstable, with parasitic velocity, that should be close to zero, oscillating considerably. Interesting, when coupled with the Height Function method, the results convergences, similar to what is observed with the Height Function coupled with the Shirani method. Comparing the Least Squares and the Height Function coupled with the Shirani/Paraboloid methods, the Least Squares presents bigger parasitic velocities for this test case.

4.1.7 Ascending bubble

The rising of a bubble in quiescent viscous liquid is simulated by Hua and Lou (2007), and the numerical predicted terminal bubble shape and Reynolds number are compared with experimental results of Bhaga and Weber (1981), for different conditions according to various Reynolds (Re), Eötvös (Eo) and Morton (M) numbers.

Some of the bubble regimes presented by Hua and Lou (2007), ranging from low Re - low Eo spherical bubbles to moderate Re - moderate Eo skirted bubbles, are compared with the AMR3D code using the Front-Tracking advection scheme from Pivello (2012) (AMR3D-FT), and the VOF method (varying the method to compute the curvature). The curvature is computed by the Height Function coupled with the Shirani method (AMR3D-HFS) and by the Least Squares method (AMR3D-LS), with stencil size of 5^3 for the normal and 3^3 for the curvature.

The cases simulated, illustrated in Fig. 4.46, are for the following bubble regimes:

a) Case A2: spherical; b) Case A3: oblate ellipsoidal (disk); c) Case A4: oblate ellipsoidal (cap); d) Case A5: spherical cap (closed wake); e) Case A7: skirted (smooth); f) Case A8: skirted (wavy).

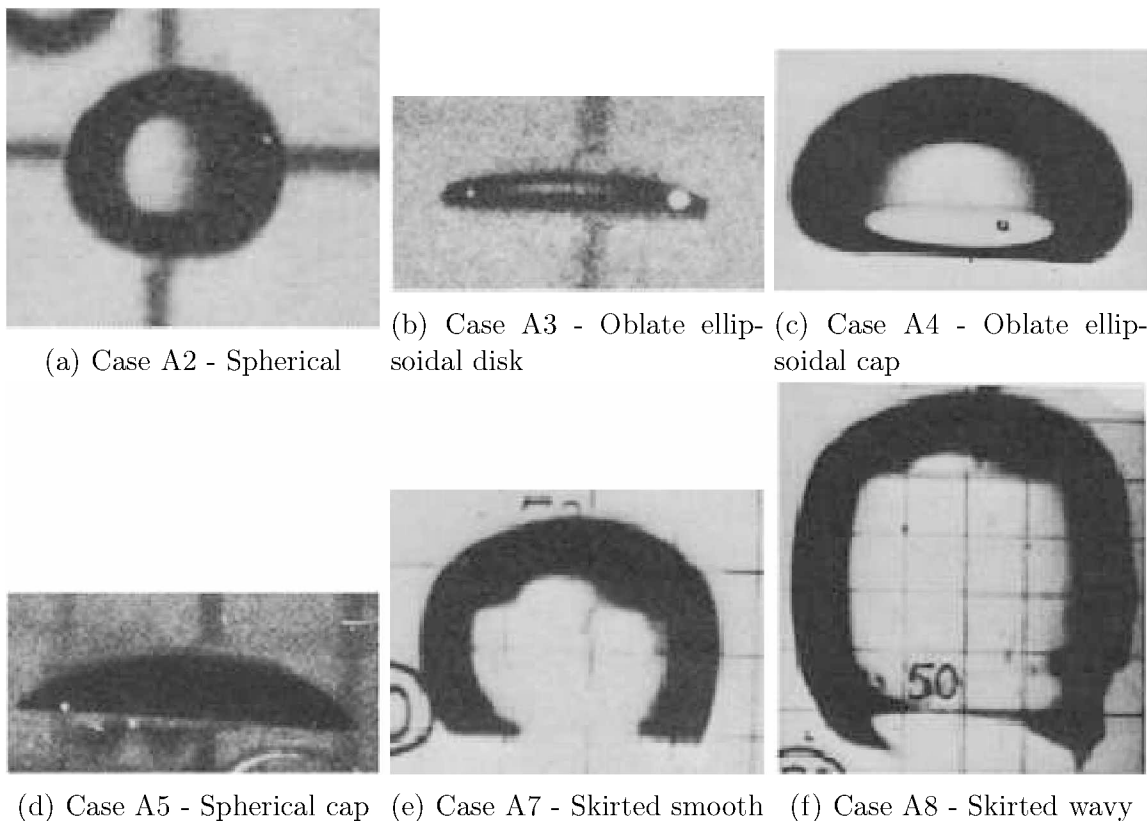


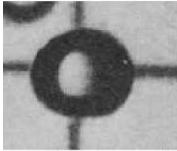
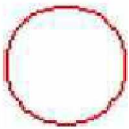

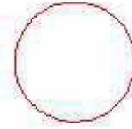

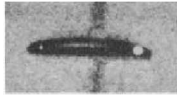




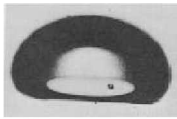




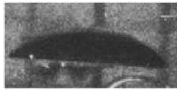





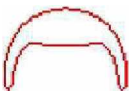




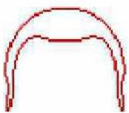



Figure 4.46: Observed bubble terminal shapes for the rising bubble experiment. Source: Hua and Lou (2007)

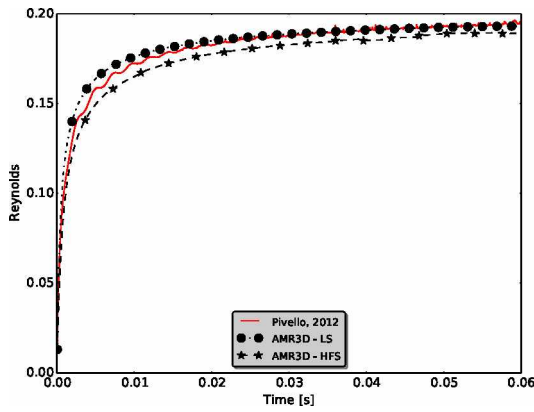
Table 4.25 and Fig. 4.47 presents the comparison results between the terminal bubble shape and the final Reynolds number, and the time evolution of the Reynolds number, respectively, computed with the Front-Tracking scheme from Pivello (2012), the VOF method with the curvature computed by the Least Squares and the Height Function coupled with the Shirani methods, and the reference works of Bhaga and Weber (1981) and Hua and Lou (2007).

On the work of Hua and Lou (2007) and Pivello (2012), the maximum error for the terminal Reynolds number occurred for the creeping flow regime (case A2). In the AMR3D-VOF method the bigger error is in case A5. For this case, fragmentation of the bubble occurs in the VOF method. The terminal shape is thus very dependent on the time step chosen to extract the values. This is clearly observed in Fig. 4.47d, with oscillation in the Re number along time occurring in the VOF methods. The Front-tracking method does not suffer this oscillation as it does not have a fragmentation model.

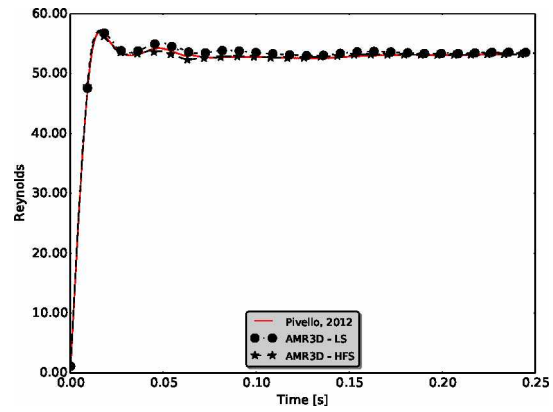
In the present work the terminal shape is in good agreement with the reference works; concerning the terminal Re number, the AMR3D-LS presents smaller errors than the AMR3D-HFS (with the exception being case A7), and in agreement with the results obtained by the AMR3D-FT.

Table 4.25: Comparison of terminal shapes and Reynolds number reported by Hua and Lou (2007), Pivello (2012) and the present work.

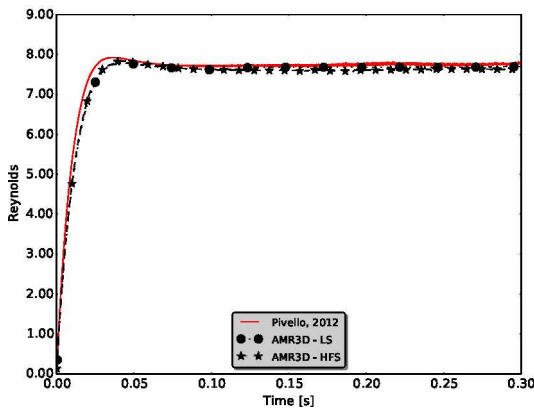
Bhaga and Weber (1981)	Hua and Lou (2007)	Pivello (2012)	AMR3D-HFS	AMR3D-LS
Case A2: $Eo=17.7$; $M=711$				
				
Re=0.232	Re = 0.211 $\varepsilon = 9.05\%$	Re = 0.212 $\varepsilon = 8.62\%$	Re = 0.189 $\varepsilon = 18.57\%$	Re = 0.194 $\varepsilon = 16.46\%$
Case A3: $Eo=32.2$; $M=8.2e-4$				
				
Re=55.3	Re = 52.9 $\varepsilon = 4.23\%$	Re = 53.2 $\varepsilon = 3.80\%$	Re = 53.2 $\varepsilon = 3.79\%$	Re = 53.5 $\varepsilon = 3.25\%$
Case A4: $Eo = 243$; $M=266$				
				
Re=7.77	Re = 8.4 $\varepsilon = 8.07\%$	Re = 7.61 $\varepsilon = 2.06\%$	Re = 7.62 $\varepsilon = 1.93\%$	Re = 7.68 $\varepsilon = 1.15\%$
Case A5: $Eo = 115$; $M=4.63e-3$				
				
Re=94.0	Re = 88.70 $\varepsilon = 5.64\%$	Re = 90.05 $\varepsilon = 4.20\%$	Re = 73.45 $\varepsilon = 21.86\%$	Re = 75.78 $\varepsilon = 19.38\%$
Case A7: $Eo = 339$; $M=43.1$				
				
Re=18.3	Re = 17.91 $\varepsilon = 2.13\%$	Re = 17.06 $\varepsilon = 6.77\%$	Re = 17.19 $\varepsilon = 6.06\%$	Re = 17.11 $\varepsilon = 6.50\%$
Case A8: $Eo = 641$; $M=43.1$				
				
Re=30.3	Re = 28.54 $\varepsilon = 1.22\%$	Re = 31.47 $\varepsilon = 3.86\%$	Re = 27.31 $\varepsilon = 9.86\%$	Re = 28.86 $\varepsilon = 4.75\%$



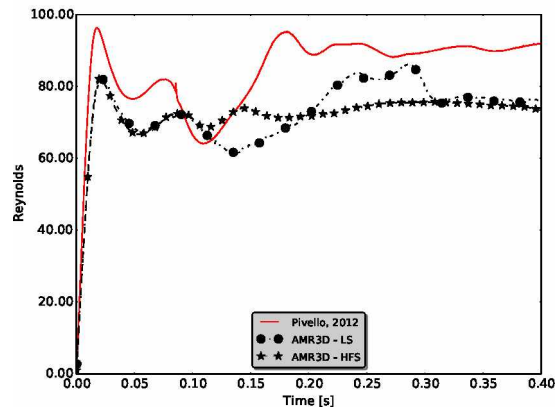
(a) Case A2



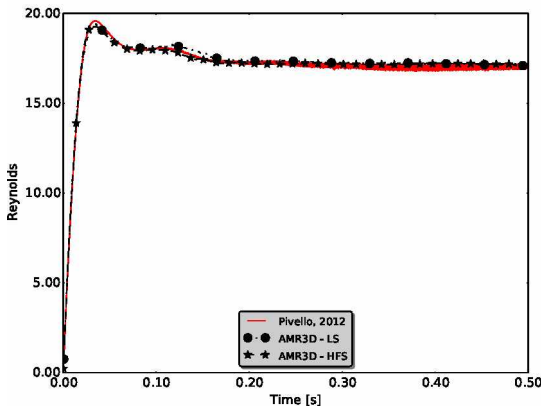
(b) Case A3



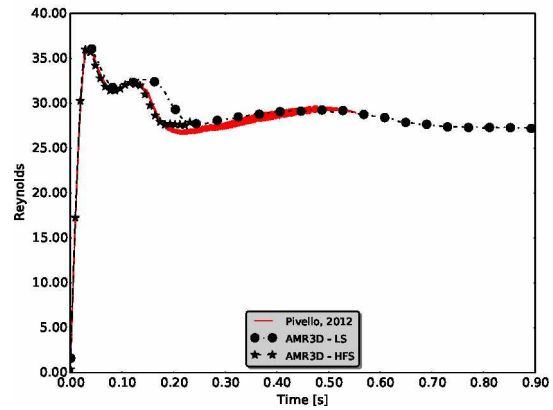
(c) Case A4



(d) Case A5



(e) Case A7



(f) Case A8

Figure 4.47: Time evolution of the Reynolds number for simulations with the Front-Tracking scheme from Pivello (2012) and with the VOF method with the Height Function and the Least Squares curvature computation methods.

4.1.8 Analysis of normal, curvature and surface tension force at the contact point

An analysis of the normal, curvature and surface tension force computation is performed at the contact point for the first time step and a comparison between the conventional wall and the Immersed Boundary wall is evaluated. The geometrical parameters of this analysis are:

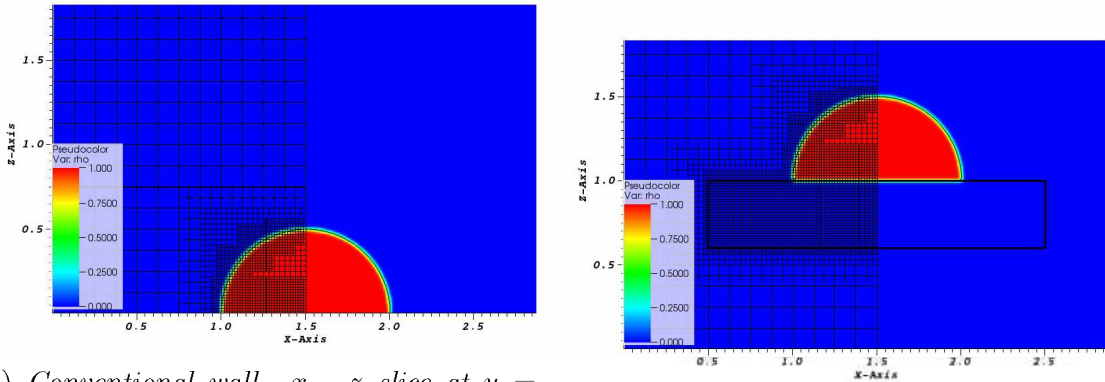
- Domain $\Omega = \{3, 3, 2\}[m^3]$;
- $n_x = n_y = 24$, $n_z = 16$ and $l = 4$ levels of refinement;
- Mesh resolution at the top level is $\Delta = \frac{1}{64}$
- $r = 0.5[m] \rightarrow \kappa_E = 4.0$
- Half droplet initialized

for the conventional wall: $x_c = 1.5[m]$, $y_c = 1.5[m]$; $z_c = 0.0[m]$

for the immersed boundary wall: $x_c = 1.5[m]$, $y_c = 1.5[m]$; $z_c = 1.0[m]$

The initialization of the color function is performed with the Points method, using 100 points (np) and 10 sub-domains (nsc). This means that each Eulerian computational cell is subdivided in nsc sub-cells. For these, the implicit function is evaluated at its vertices's to identify if the sub-cell is completely inside or outside of the implicit equation, which means that they are full or empty, respectively. For each sub-cells that has vertices inside and outside, the volume fraction is computed with np points. The error in the initialized vof volume is $\epsilon_{vol} = 2.34E - 4$ for both cases. The volume fraction initialized, the mesh used and the Immersed Boundary are presented in Fig. 4.48

4.1.8.1 Color function



(a) Conventional wall. $x - z$ slice at $y = 1.5[m]$

(b) IB wall. $x - z$ slice at $y = 1.5[m]$

Figure 4.48: Color function. $x - z$ slice at $y = 1.5[m]$

In the conventional wall, Newman boundary condition is applied to the color function. A boundary at the conventional wall is represented by Dirichlet boundary condition to the velocity normal to it, to ensure no-penetration, and by Navier-Slip to the velocities parallel to it. At the Immersed Boundary, the presence of the wall is represented by a source term at the right hand side of the momentum equation; This difference in how the boundary condition is applied in the conventional and in the Immersed boundary wall leads to a slightly different values obtained, as explained in the following sections.

4.1.8.2 Normal component parallel to the wall

The normal x component is presented in Fig. 4.49. A comparison between the values obtained for the normal component in x -direction at the conventional and the Immersed Boundary wall is presented in Fig. 4.50. In the Least Squares system, only values of cells considered inside the domain are used. Values from cells considered outside are not used nor computed.

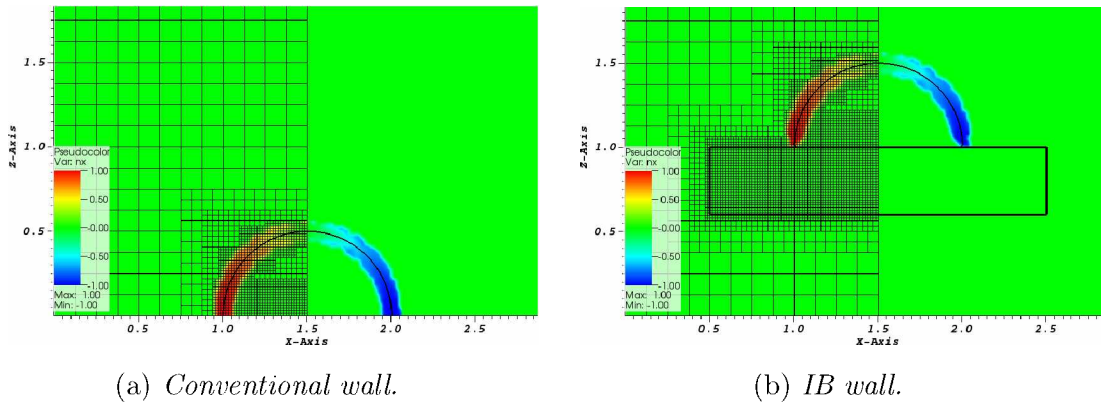


Figure 4.49: Normal x component. $x - z$ slice at $y = 1.5[m]$

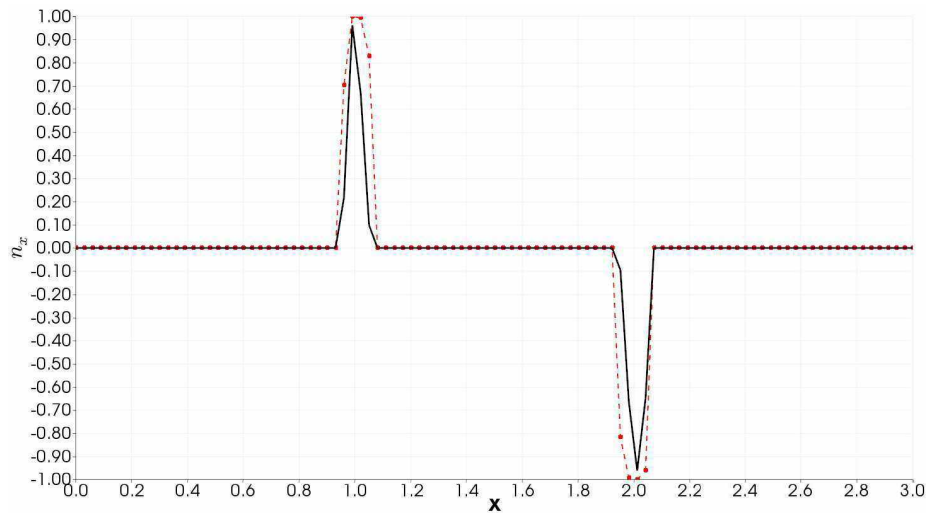


Figure 4.50: Normal x . Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; *Dashed line: Conventional wall; Continuous line: IB wall*

4.1.8.3 Normal component perpendicular to the wall

For the normal perpendicular to the wall (at z_{th} direction), in the case being analyzed, the difference of treatment of the color function at the wall is representative. For the conventional wall, with Newman boundary condition being applied, the normal component inside the bubble, and in contact with the wall, vanishes. This can be seen in Fig. 4.51a. For the

Immersed Boundary, since only values inside the bubble are considered, there is a gradient of the color function through the droplet, which leads to the resultant normal illustrated in Fig. 4.51b. A comparison between the results is presented in Fig. 4.52.

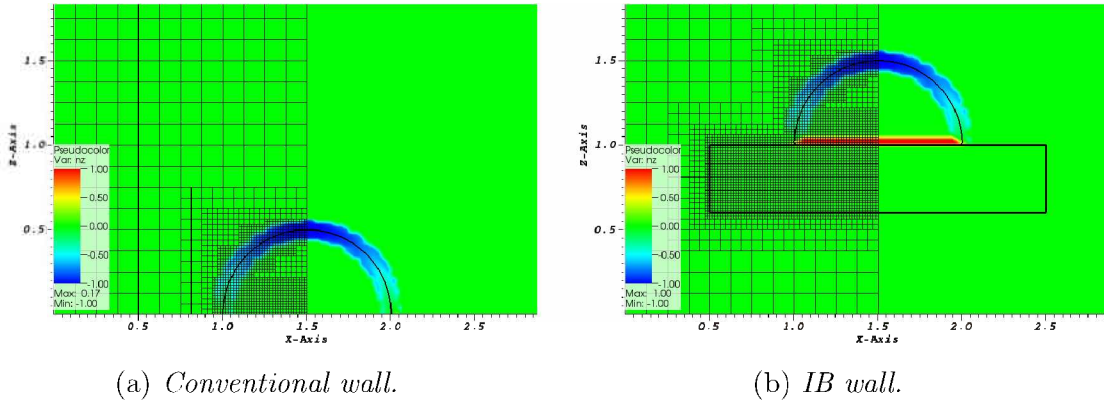


Figure 4.51: Normal z . $x - z$ slice at $y = 1.5[m]$

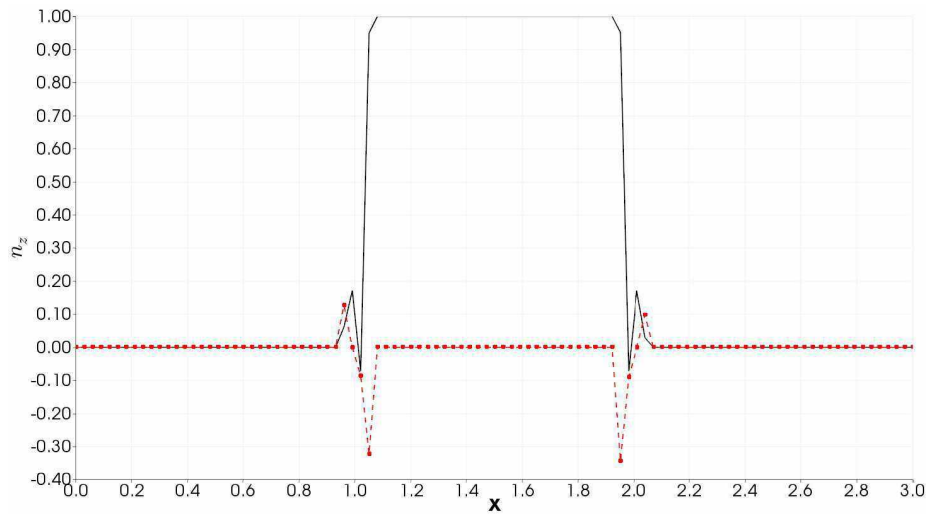


Figure 4.52: Normal z . Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; *Dashed line: Conventional wall; Continuous line: IB wall*

4.1.8.4 Curvature

The curvature is similar along the droplet, with the case with the Immersed Boundary wall presenting a constant value different than zero at its bottom, as illustrated in Fig. 4.54. This is due to the normal- z component. The results for both cases are compared in Fig. 4.55. The value at the contact point is similar, and the main difference between the conventional and the Immersed Boundary wall is inside the droplet. It vanishes inside the droplet at the region in contact with the conventional wall, since the normal z component is absent and Newman boundary condition is applied to the color function. For the Immersed Boundary,

due to the value of the normal z component and the gradient of the color function along the Immersed Boundary, it presents a constant value.

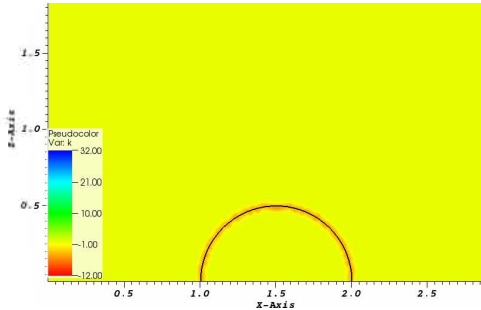


Figure 4.53: Curvature. *Conventional wall. $x - z$ slice at $y = 1.5[m]$*

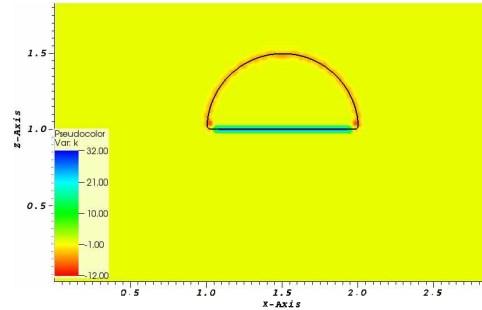


Figure 4.54: Curvature. *IB wall. $x - z$ slice at $y = 1.5[m]$*

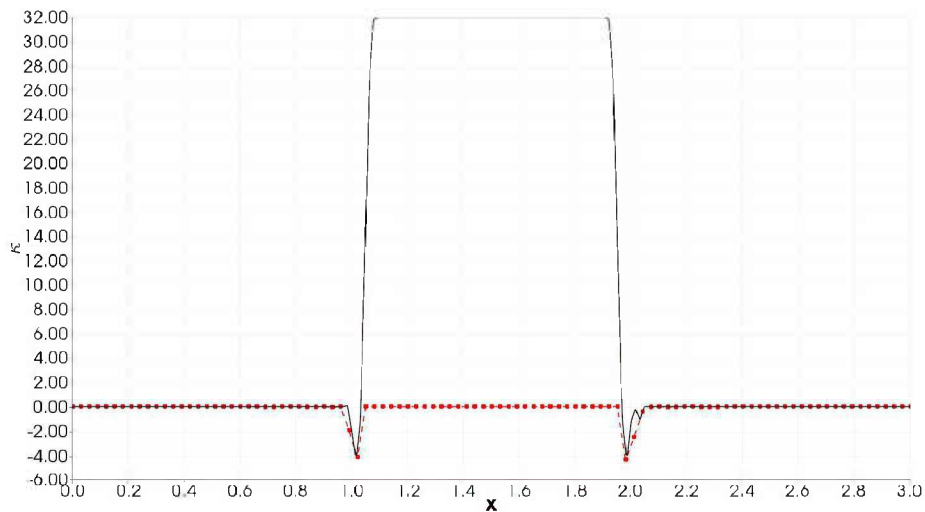


Figure 4.55: Curvature. Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; *Dashed line: Conventional wall; Continuous line: IB wall*

4.1.8.5 Force components

The force parallel to the wall (x component) is presented in Fig. 4.58 and compared in Fig. 4.59. Since the values of the normal vector and the curvature are similar, the force component is also similar. As for the force perpendicular to the wall (z th component), due to the difference in normal and curvature, at the contact point there is a peak force for the Immersed Boundary wall which is not present in the conventional wall. This is illustrated in Fig. 4.62 and Fig. 4.63.

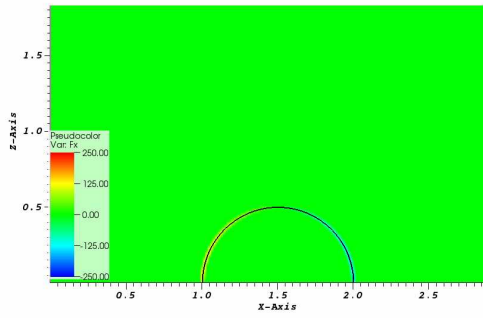


Figure 4.56: *Conventional wall.*

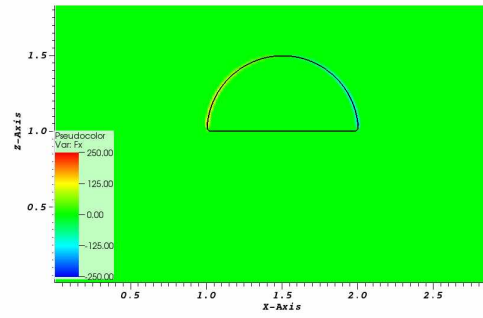


Figure 4.57: *IB wall.*

Figure 4.58: Force x. $x - z$ slice at $y = 1.5[m]$

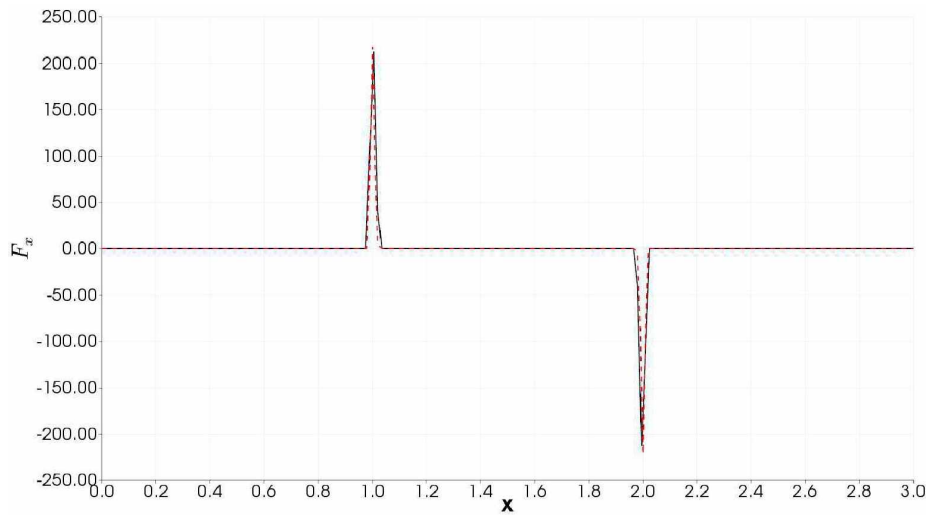


Figure 4.59: Force x. Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5$; *Dashed line: Conventional wall; Continuous line: IB wall*

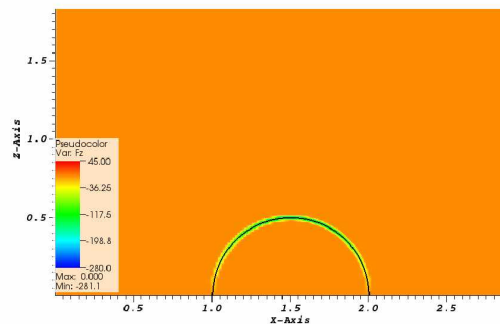


Figure 4.60: *Conventional wall.*

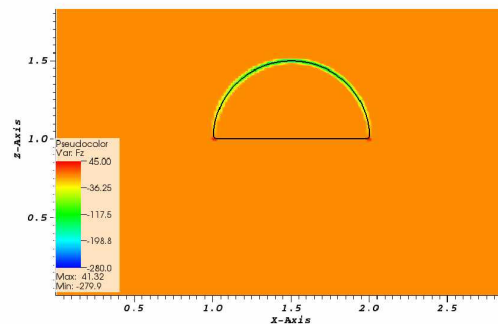


Figure 4.61: *IB wall.*

Figure 4.62: Force z. $x - z$ slice at $y = 1.5[m]$

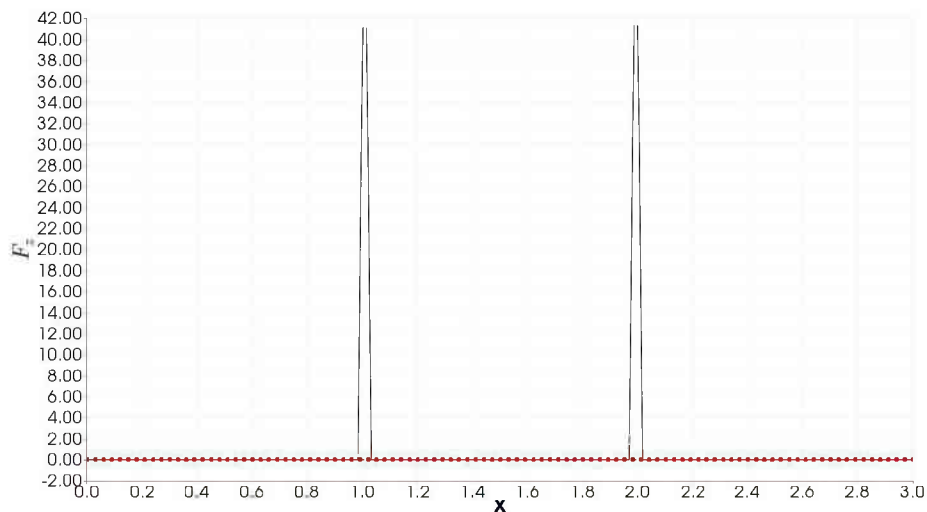


Figure 4.63: Force z . Data extracted over a line crossing the computational cell adjacent to the wall, with coordinates $0 < x < 3$; $y = 1.5[m]$; *Dashed line: Conventional wall*; *Continuous line: IB wall*

The difference in the force at z -component, normal to the wall, should be considered. Either a Newman boundary condition should be applied at the Immersed Boundary, or the method must compute a source term that accounts for this force to avoid no-penetration through the wall (which is currently done in the AMR3D code).

4.2 Applications

4.2.1 3D contact point study with density ratio $\rho_l/\rho_g = 1.0$ and viscosity ratio $\mu_l/\mu_g = 1.0$, based on the 2D work of Lai, Tseng and Huang (2010)

Based on the 2D work of Lai, Tseng and Huang (2010), 3D numerical experiments with hydrophilic and hydrophobic cases are performed. A drop on a solid surface initially at equilibrium is suddenly imposed to a different contact angle; the drop fluid should accelerate toward a steady state defined by the new value of θ_E . The force at the contact point should vanish as the equilibrium contact angle is approximated.

The setup consists of an initially hemispherical drop ($\theta = 90^\circ$) at $t = 0[s]$, of radius $r = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$. The density ratio is $\rho_l/\rho_g = 1$, the viscosity ratio is $\mu_l/\mu_g = 1$, surface tension is $\sigma = 1[N/m]$. Both free-slip and no-slip boundary conditions are used on the bottom of the domain (Lai, Tseng and Huang (2010) used a slip condition $\lambda = \Delta/4$, where Δ is the smallest mesh size), and an open boundary condition is specified elsewhere. The gravity is not considered. For this particular case, the only force responsible for the movement of the interface is the contact point force.

The computations are performed in 4 processors, with an adaptive mesh of base $[12 \times 12 \times 12]$ and 2 to 6 levels of refinement (maximum grid resolution of $\Delta = 1/8, 1/16, 1/32, 1/64$ and $1/128$, respectively). The *Semi-Backward Difference* is used for the temporal

discretization, and the *Central Difference Scheme (CDS)* for the advection model.

The time step is calculated following stability criteria for explicit schemes, taking into account the advective, diffusive and capillary terms, as defined by Eq.(3.11), with safety coefficients $a_1 = b_1 = b_2 = b_3 = 0.5$.

$$\Delta t = \min(0.5\Delta t_{adv}, 0.5\Delta t_{diff}, 0.5\Delta t_{cap}). \quad (4.18a)$$

The error in volume conservation ϵ_{vol} is defined by Eq.(4.19)

$$\epsilon_{vol} = |V_{cur} - V_{prev}| \quad (4.19)$$

where: V_{cur} and V_{prev} are the vof volume at the current and previous time steps, respectively.

The error between the achieved angle and the desired one, ϵ_θ , is computed by Eq.(4.20):

$$\epsilon_\theta = \frac{|\theta_N - \theta_E|}{\theta_E} \times 100[\%] \quad (4.20)$$

where: θ_E is the equilibrium contact angle; θ_N is the numerical contact angle, computed with: 1) the free open-source multi-platform Java image-processing program ImageJ (SCHNEIDER; RASBAND; ELICEIRI, 2012); 2) with the DropSnake plugin for ImageJ, a method based on B-spline snakes (active contours) for measuring high-accuracy contact angles (STALDER et al., 2006). θ_N is the average between three measures from ImageJ ($\bar{\theta}_{IJ}$) and the left (θ_{DSL}) and right (θ_{DSR}) measured contact angles from the DropSnake plugin.

4.2.1.1 *Hydrophilic case: $\theta = 60^\circ$, base mesh [12 x 12 x 4] and 2 to 6 levels of refinement. No-Slip boundary condition.*

A convergence analysis is performed, with maximum grid resolution varying from $\Delta = \frac{1}{8}$ to $\Delta = \frac{1}{128}$. The time step is determined by the capillarity restriction, and is presented in Tab. 4.26 for each mesh resolution tested, as well as the error in the initialized vof volume ϵ , the error in the numerically measured contact angle $\epsilon_\theta\%$, and the ratio of convergence of the numerically measured contact angle from the current to the previous mesh resolution R .

Table 4.26: Mesh resolution, time step, error in the initialized vof volume and contact angle error for $\theta = 60^\circ$ and $\frac{1}{8} \leq \Delta \leq \frac{1}{128}$. (No-Slip boundary condition.)

Δ	$\Delta t[s]$	ϵ	$\epsilon_\theta\%$	R
$\frac{1}{8}$	$1.76E - 2$	$1.93E - 3$	19.56	-
$\frac{1}{16}$	$6.23E - 3$	$9.70E - 4$	11.89	1.64
$\frac{1}{32}$	$2.20E - 3$	$4.87E - 4$	4.7	2.53
$\frac{1}{64}$	$7.79E - 4$	$2.43E - 4$	0.56	8.4
$\frac{1}{128}$	$2.75E - 4$	$1.21E - 4$	0.67	0.83

Figures 4.64 and 4.65 present the interface ($x - z$ plane view and $x - y$ plane view, respectively) at the final time step, and the Eulerian mesh used. The interface advances to the desired angle and the code was able to maintain the interface symmetry even for the coarse meshes tested.

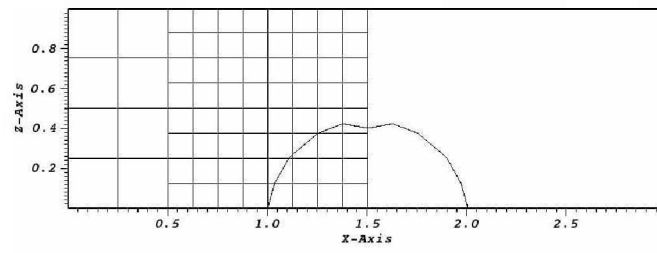
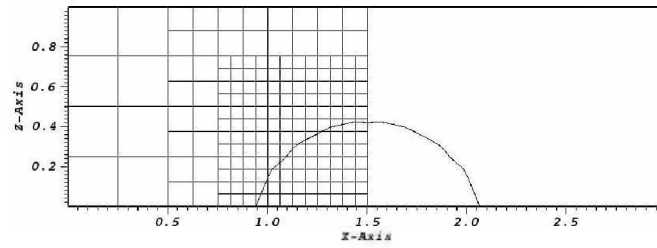
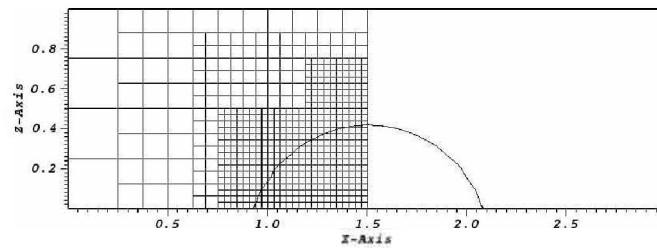
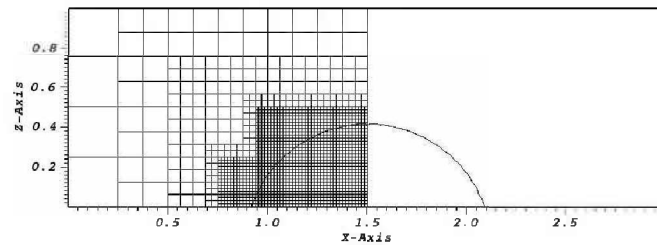
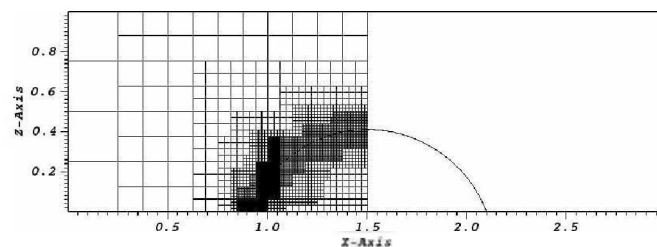
(a) $\Delta = 1/8$ (b) $\Delta = 1/16$ (c) $\Delta = 1/32$ (d) $\Delta = 1/64$ (e) $\Delta = 1/128$

Figure 4.64: x-z plane view ($y = 1.5[m]$) of the final position of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution: a) $\Delta = \frac{1}{8}$; b) $\Delta = \frac{1}{16}$; c) $\Delta = \frac{1}{32}$; d) $\Delta = \frac{1}{64}$; e) $\Delta = \frac{1}{128}$.

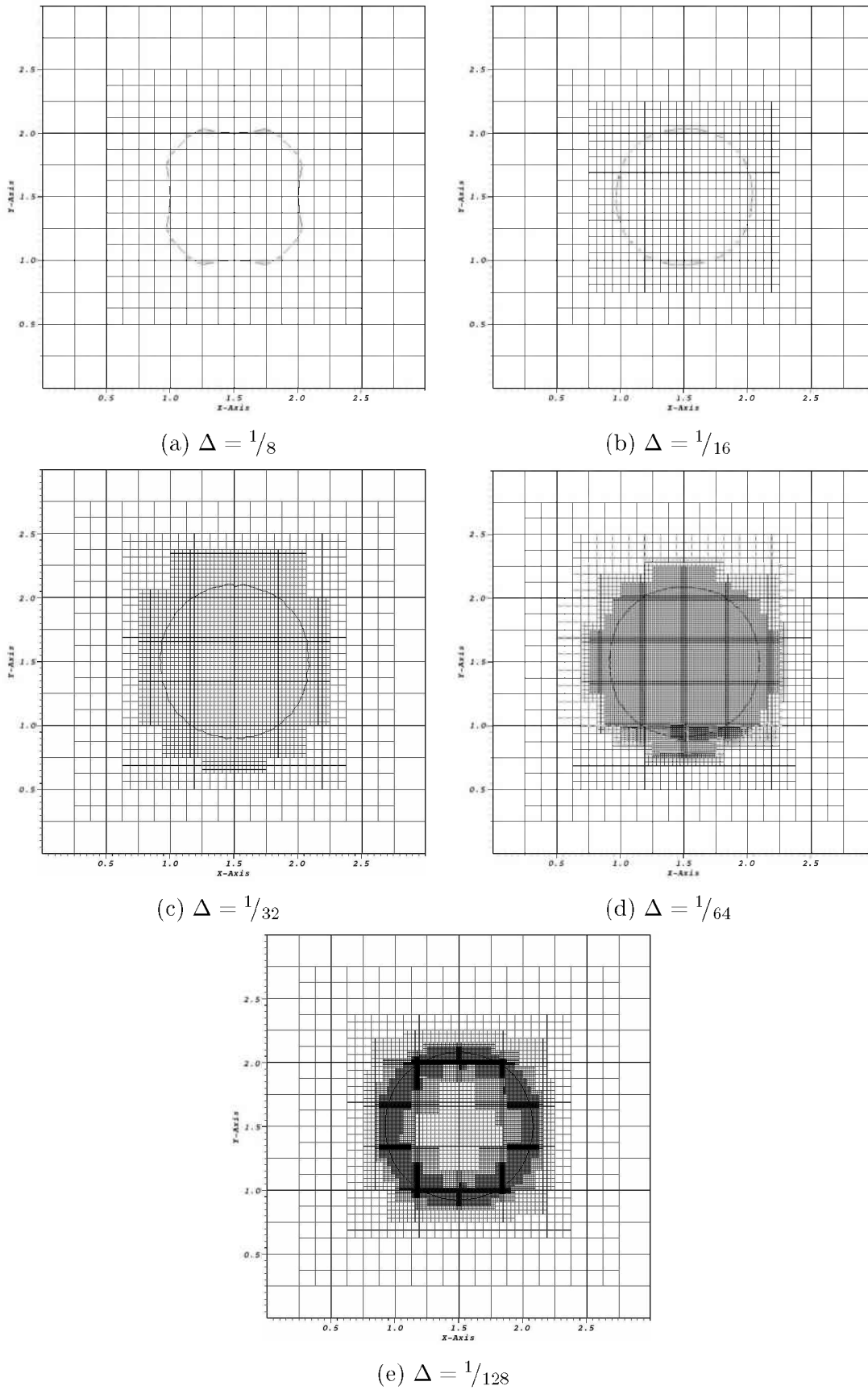


Figure 4.65: x-z plane view ($y = 1.5[m]$) of the final position of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution: a) $\Delta = \frac{1}{8}$; b) $\Delta = \frac{1}{16}$; c) $\Delta = \frac{1}{32}$; d) $\Delta = \frac{1}{64}$; e) $\Delta = \frac{1}{128}$.

The volume conservation is presented in Fig. 4.66, being on machine error order for all simulation.

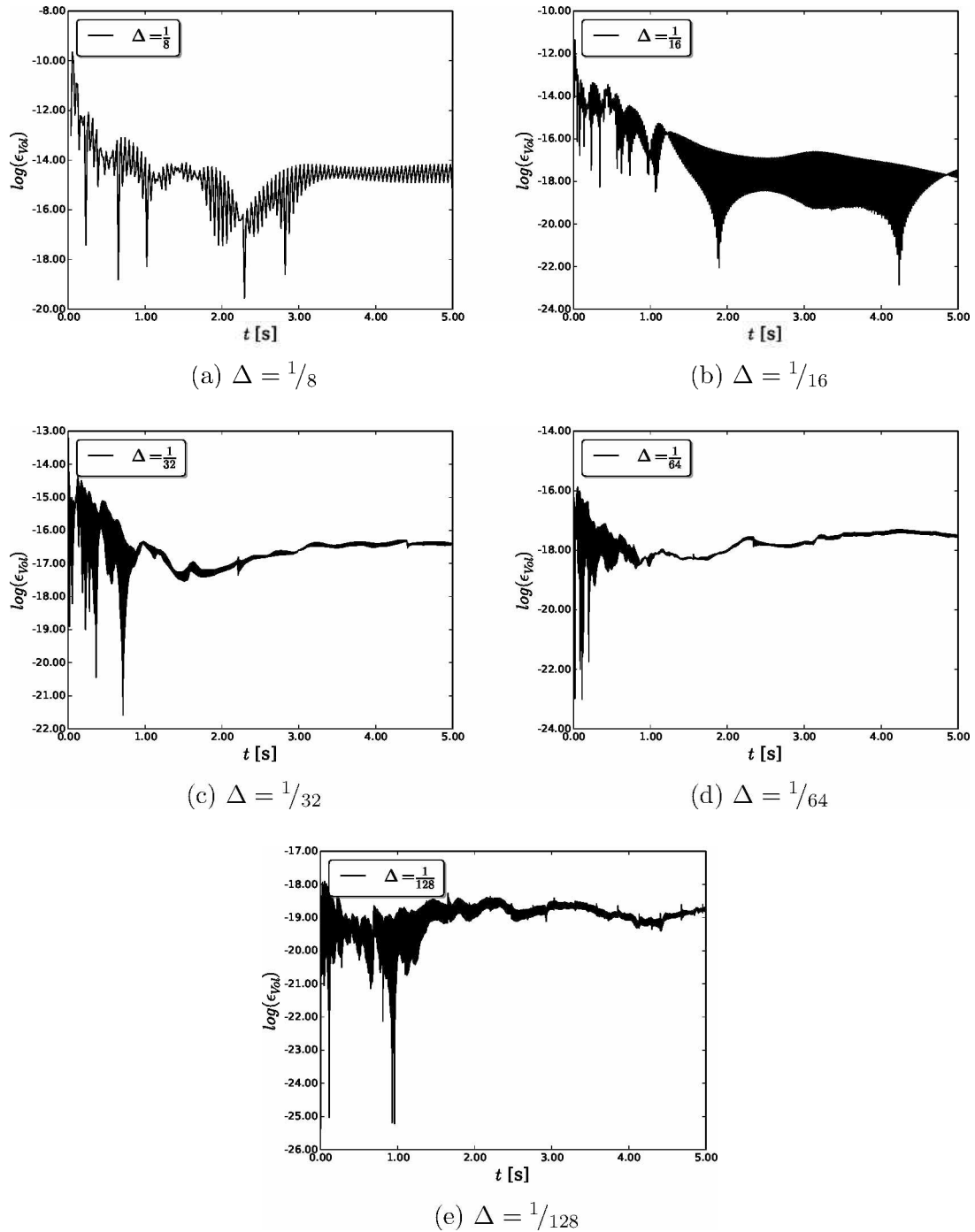


Figure 4.66: log of the error in volume conservation (ϵ_{vol}) vs time.

4.2.1.2 Hydrophilic case: $\theta = 60^\circ$, base mesh [12 x 12 x 4] and 6 levels of refinement. Comparison between Free-Slip and No-Slip boundary conditions

The previous section presented results with only the No-Slip boundary condition. In this section the results with the Free-Slip boundary condition are also presented and com-

pared with the No-Slip boundary condition.

From the convergence analysis performed, the maximum grid resolution of $\Delta = \frac{1}{128}$ is chosen for the comparison between the two boundary conditions. The time step is determined by the capillarity restriction, and is equal to $\Delta t = 2.75E - 4[s]$. The error in the initialized vof volume is $\epsilon = 1.21E - 4$.

The volume conservation is presented in Fig. 4.67, being on machine error order for both simulations.

Figures 4.68 and 4.69 present the time evolution of the interface ($x - z$ slice) and the Eulerian mesh used, for the No-Slip and the Free-Slip cases, respectively. The interface advances to the desired angle, and no deformation on the interface is observed at the end of the simulation. From the top view, presented in Fig. 4.71, the interface spherical shape is maintained at the bottom, as well as its symmetry.

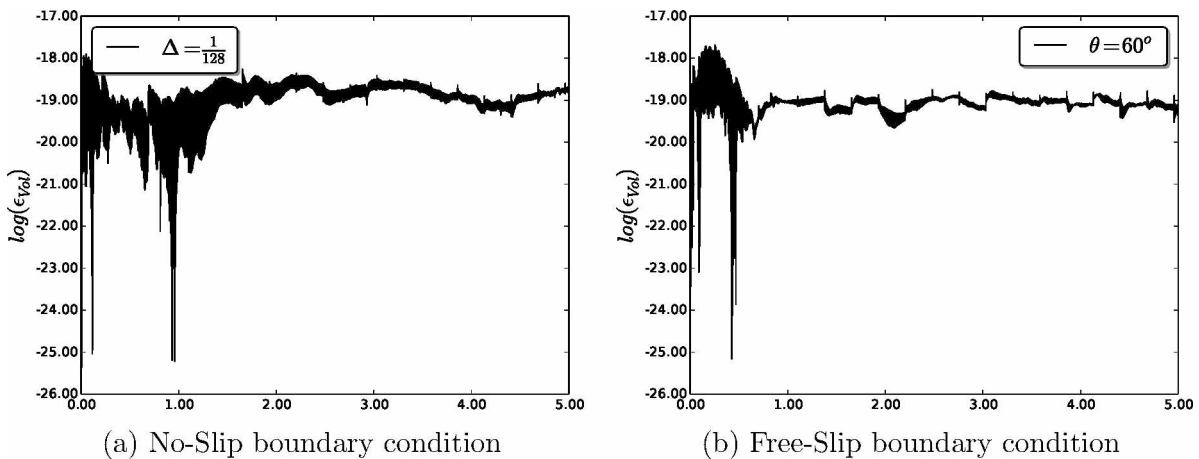


Figure 4.67: \log of the error in volume conservation (ϵ_{vol}) vs *time* for mesh with maximum resolution $\Delta = \frac{1}{128}$

A comparison between the two boundary conditions can be seen at Fig. 4.70 (side view) and 4.71 (Top view). It can be observed that, for this mesh resolution, the final position achieved is the same, but with the Free-Slip the interface advances faster to its final position.

For the No-Slip boundary condition, contact point slip is achieved implicitly, as the advection scheme used to advect volume fractions utilizes face-centered velocities, so that the nearest velocity to the contact point is one half cell width above a solid boundary.

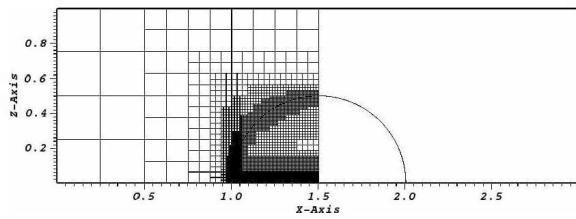
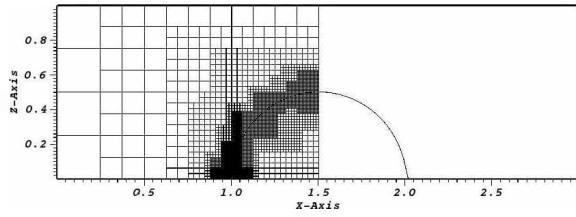
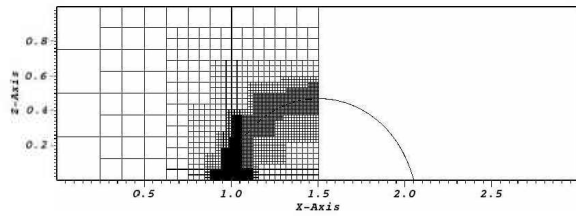
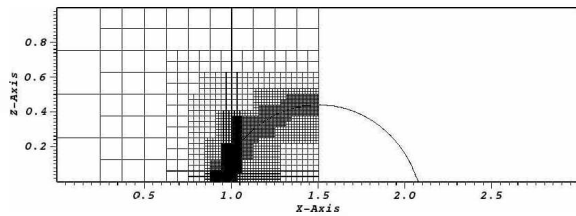
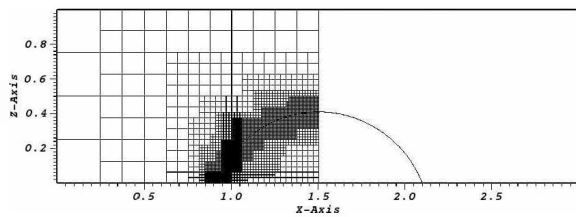
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.68: x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. No-slip boundary condition.

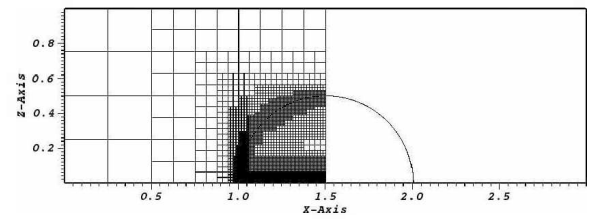
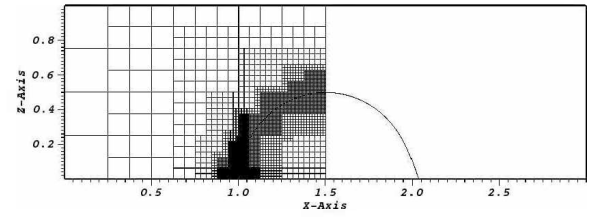
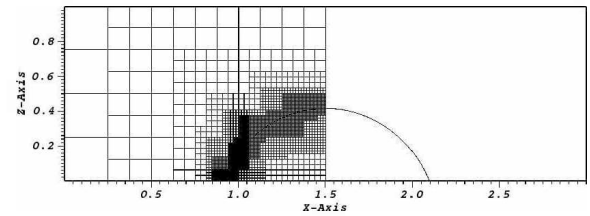
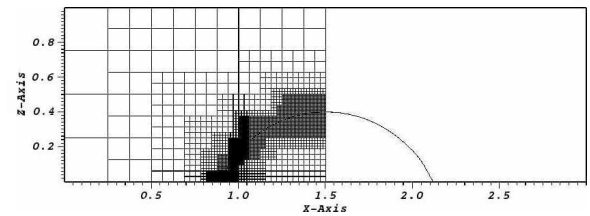
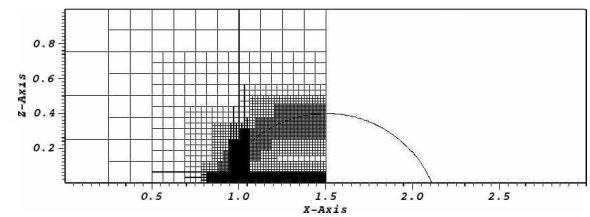
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.69: x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. Free-slip boundary condition.

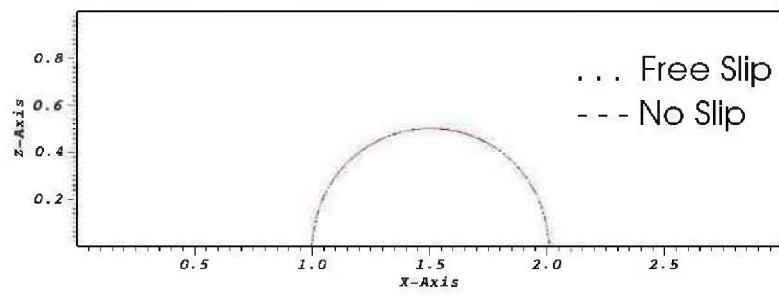
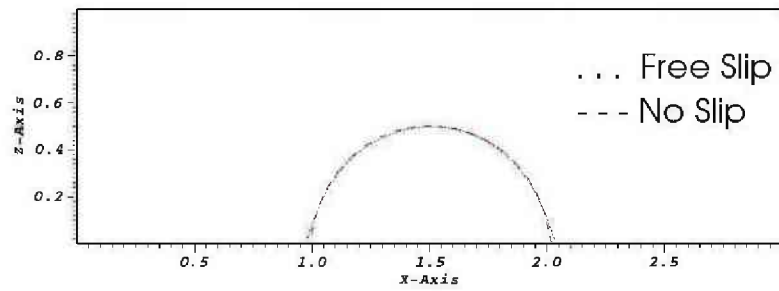
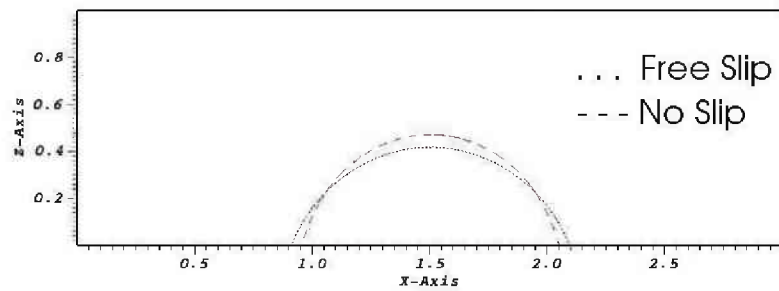
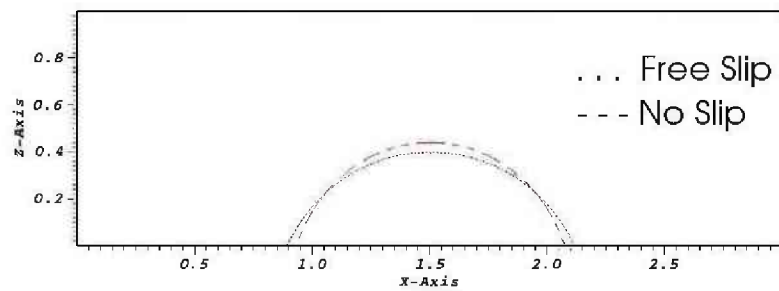
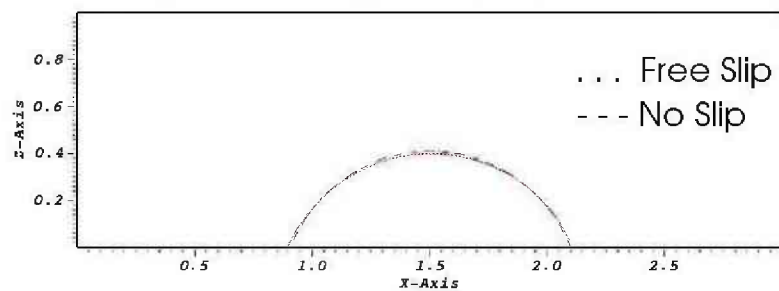
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.70: Side View of the time evolution of the interface with No-Slip and Free-Slip boundary conditions

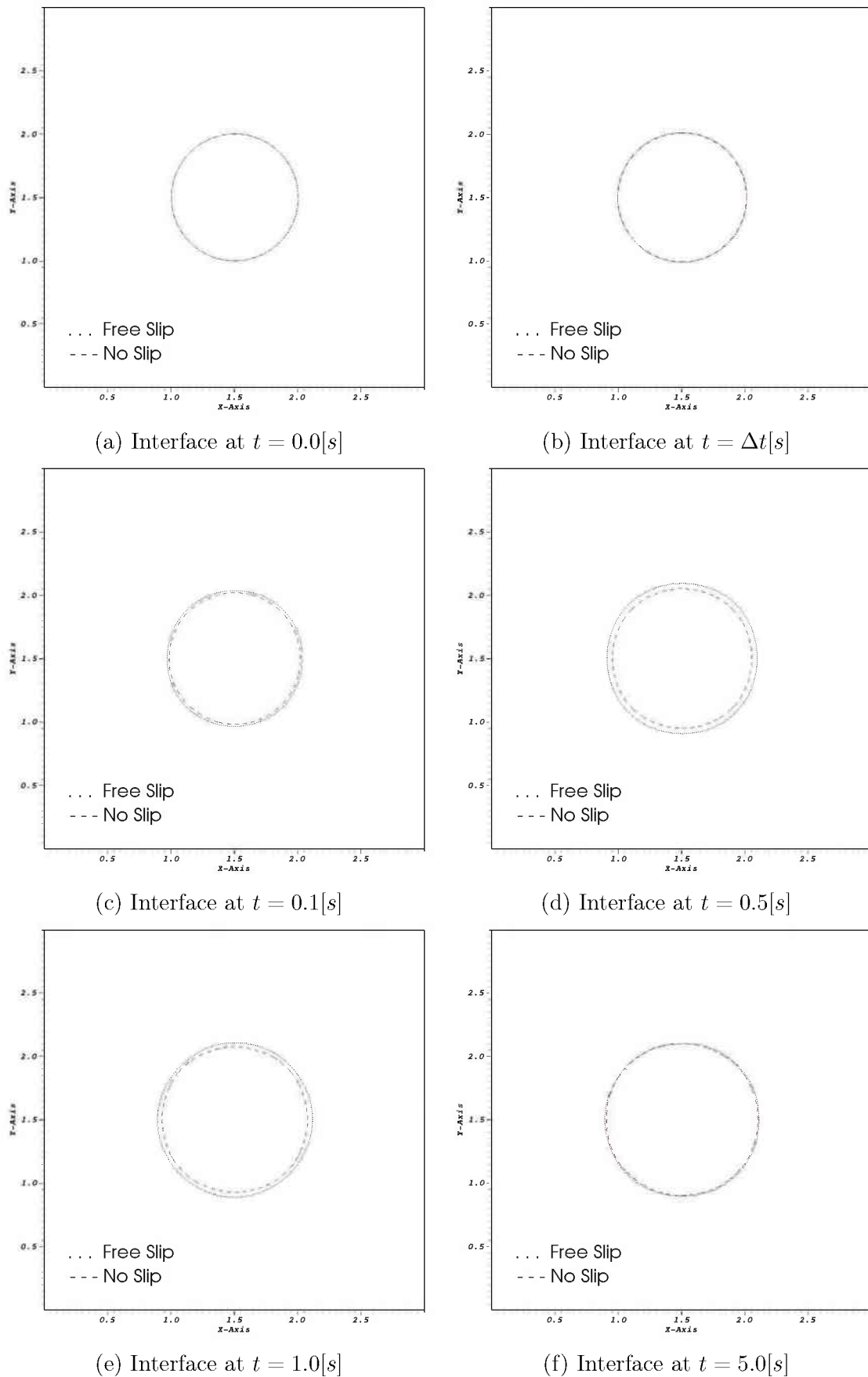
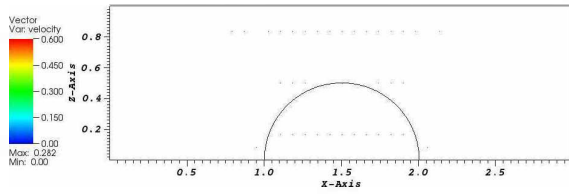
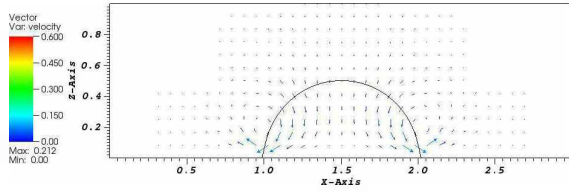


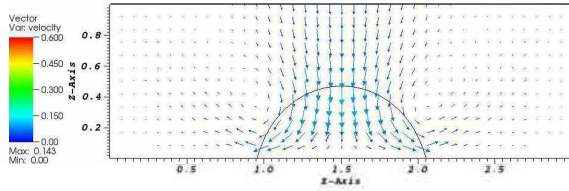
Figure 4.71: Top view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions



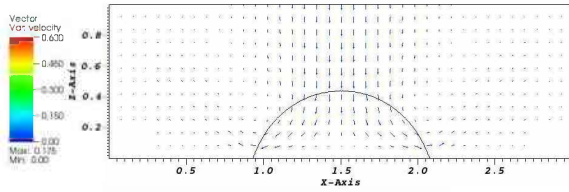
(a) $t = \Delta t[s]$



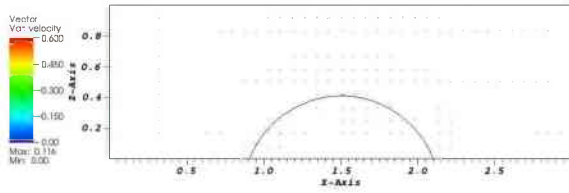
(b) $t = 0.1[s]$



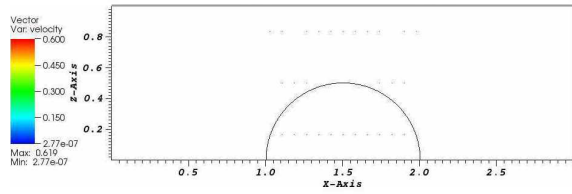
(c) $t = 0.5[s]$



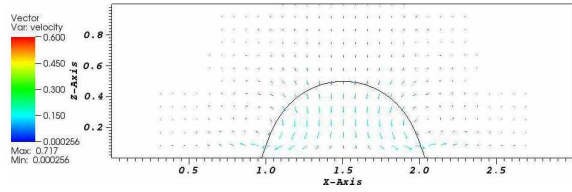
(d) $t = 1.0[s]$



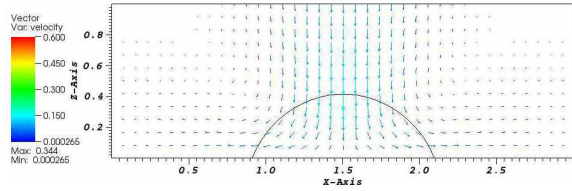
(e) $t = 5.0[s]$



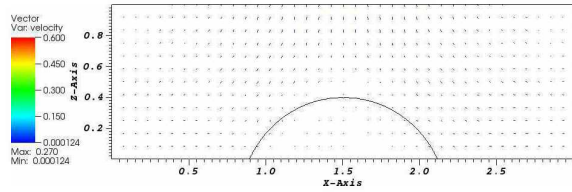
(a) $t = \Delta t[s]$



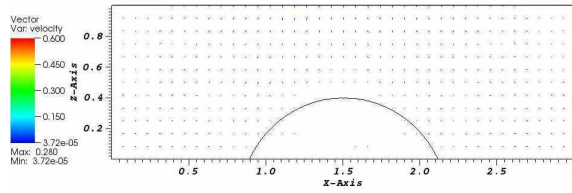
(b) $t = 0.1[s]$



(c) $t = 0.5[s]$



(d) $t = 1.0[s]$



(e) $t = 5.0[s]$

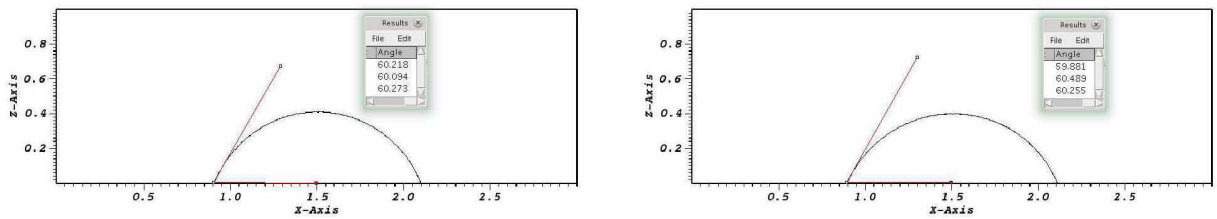
Figure 4.72: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. No-Slip boundary condition.

Figure 4.73: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{128}$. Free-Slip boundary condition.

The velocity field, presented in Figs. 4.72 (No-Slip B.C) and 4.73 (Free-Slip B.C), acts in the expected direction, to spread the droplet over the surface, being very similar to the previously coarse mesh case ($\Delta = 1/64$); at time $T = 1.0[s]$ no recirculation at the contact point is observed. The values obtained with the Free-Slip are two ($t = \Delta t[s]$) to

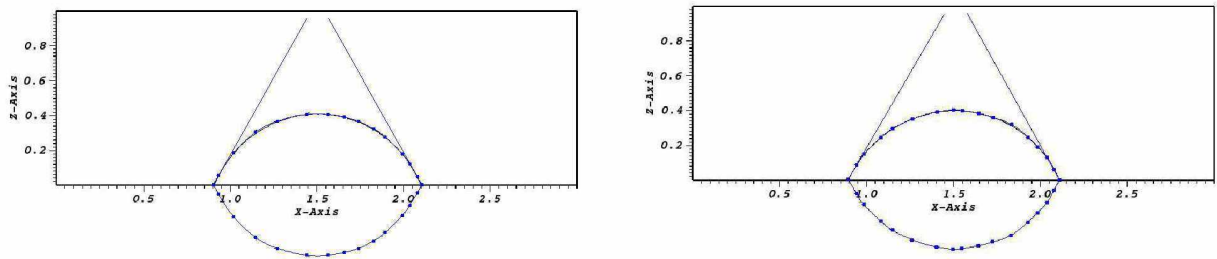
three ($t = 0.1[s]$) times bigger than with the No-Slip, and explains why the interface moves quicker to the desired position. This can be explained by the tangential velocity values at the wall in the Free-Slip case, which is not zero, as it is in the No-Slip case.

The numerically measured contact angles are presented in Figs. 4.74 and 4.75 and are equal to $\theta_N - 60.4^\circ$ for the No-Slip boundary condition and $\theta_N - 60.3^\circ$ for the Free-Slip boundary condition. The error between the numerical and the equilibrium angles are equal to $\epsilon_\theta = 0.67\%$ and $\epsilon_\theta = 0.50\%$ (No-Slip and Free-Slip boundary condition, respectively). This error is on the same order of magnitude of the value observed at the previous coarse mesh resolution $\Delta = 1/64$ ($\epsilon_\theta = 0.56\%$).



(a) ImageJ averaged measured contact angle $\theta_{IJ} = 60.2^\circ$ for the No-Slip B.C. (b) ImageJ averaged measured contact angle $\theta_{IJ} = 60.2^\circ$ for the Free-Slip B.C.

Figure 4.74: contact angle measured at the end of the simulation for $\theta_E = 60^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{128}$.



(a) DropSnake measurement: $\theta_{DSL} = 60.6^\circ$, $\theta_{DSR} = 60.5^\circ$ for the No-Slip B.C. (b) DropSnake measurement: $\theta_{DSL} = 59.9^\circ$, $\theta_{DSR} = 60.7^\circ$ for the Free-Slip B.C.

Figure 4.75: contact angle measured at the end of the simulation for $\theta_E = 60^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{128}$ (DropSnake measurement).

4.2.1.3 Hydrophilic case: $\theta = 45^\circ$, base mesh $[12 \times 12 \times 4]$ and 5 levels of refinement. Comparison between Free-Slip and No-Slip boundary conditions.

For the test cases with $\theta = 45^\circ$, the maximum grid resolution was chosen to be $\Delta = \frac{1}{64}$. From the analysis of the cases with $\theta = 60^\circ$, resumed on Tab. 4.26, the errors between the numerical and the equilibrium angles are on the same order of magnitude for the simulations with 5 and 6 levels of refinement. The time step is determined by the capillarity restriction, and is equal to $\Delta t = 7.79E - 4[s]$. The error in the initialized vof volume is $\epsilon = 2.43E - 4$.

The results for both the Free-Slip and No-Slip boundary condition are presented and compared. The volume conservation is presented in Fig. 4.76, being on machine error order

for both simulations.

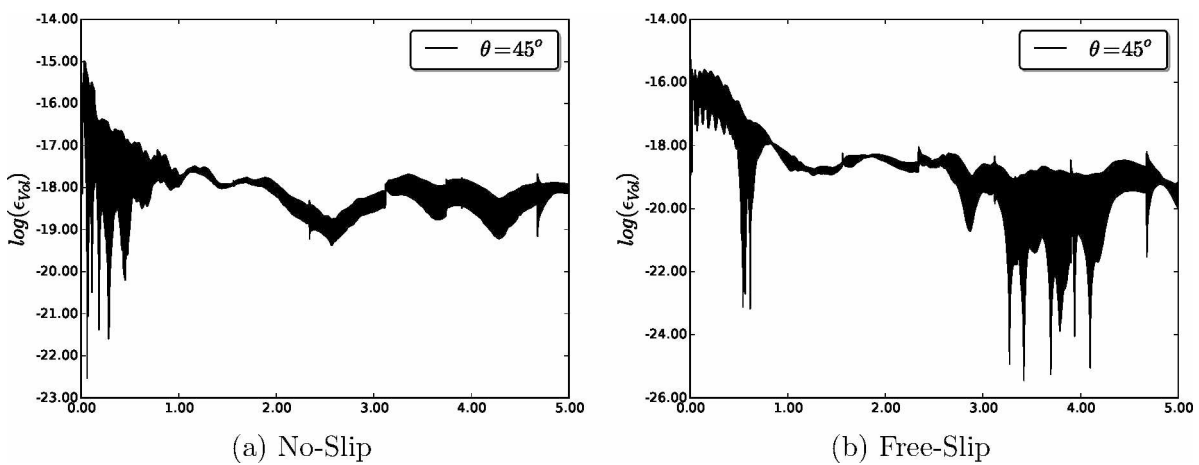


Figure 4.76: log of the error in volume conservation (ϵ_{vol}) vs *time* for mesh with maximum resolution $\Delta = \frac{1}{64}$

Figures 4.77 and 4.78 present the time evolution of the interface ($x - z$ slice) and the Eulerian mesh used, for the No-Slip and the Free-Slip cases, respectively. The interface advances to the desired angle, and no irregularities are observed at the end of the simulation. From the top view, presented in Fig. 4.80, it can be seen that the interface spherical shape is maintained at the bottom, as well as its symmetry.

A comparison between the interface evolution for the two boundary conditions can be seen in Fig. 4.79 (side view) and 4.80 (Top view). With the Free-Slip boundary condition the interface advances faster to the final position, but as presented in Figs. 4.83 and 4.84 the numerical angle observed at the end of the simulation are the same for both conditions.

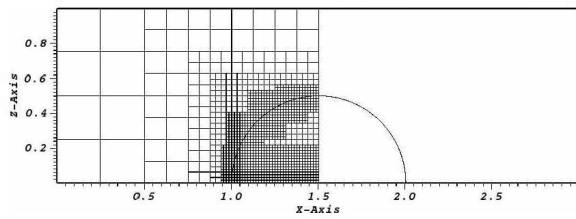
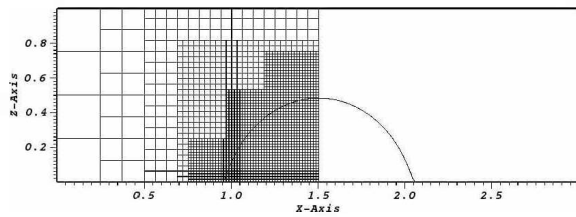
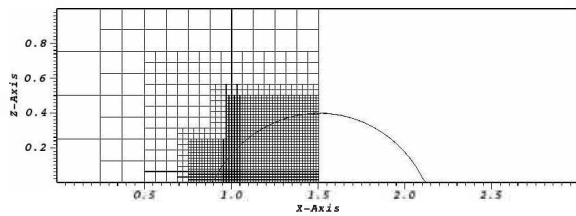
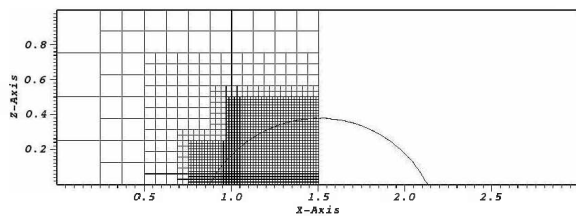
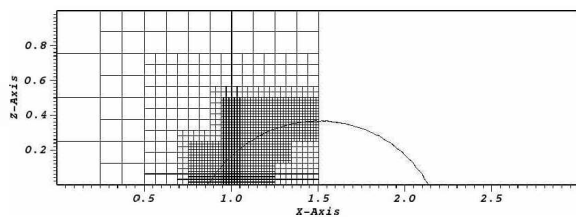
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.77: x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-slip boundary condition.

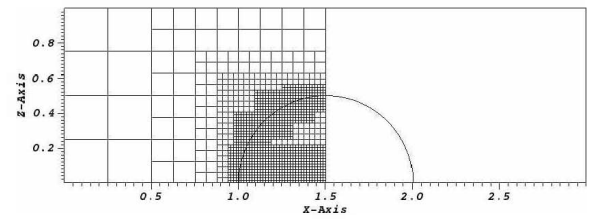
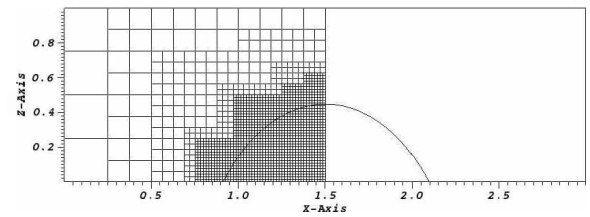
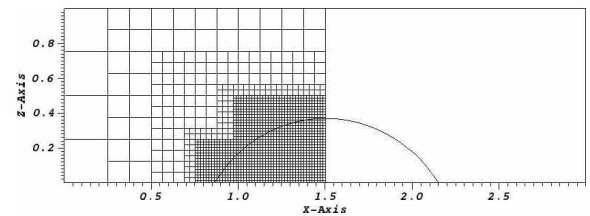
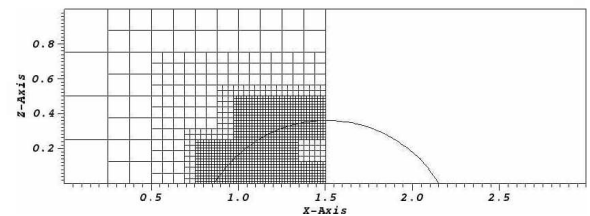
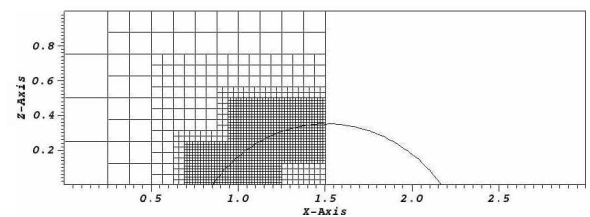
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.78: x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-slip boundary condition.

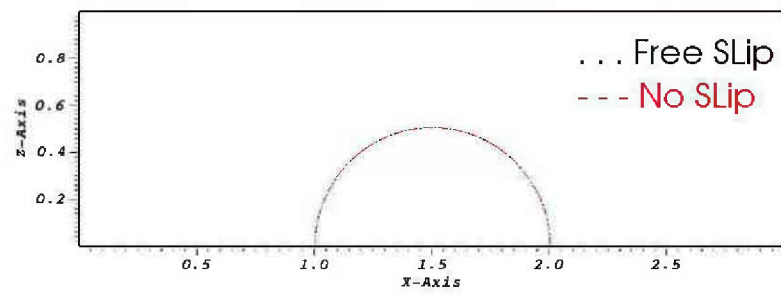
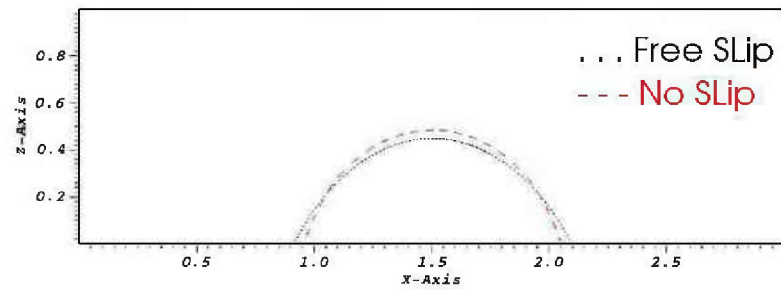
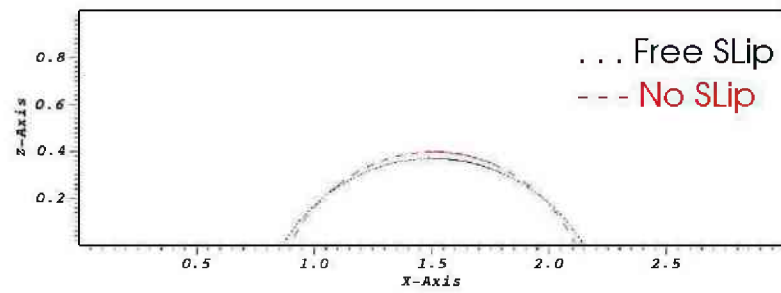
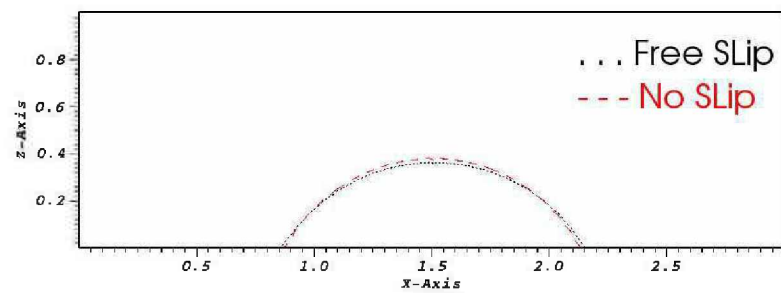
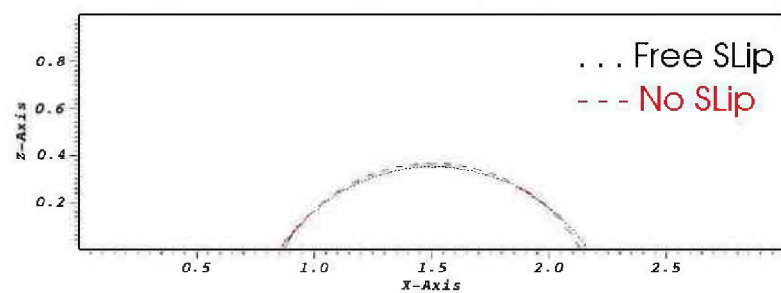
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.79: Side View of the time evolution of the interface with No-Slip and Free-Slip boundary conditions

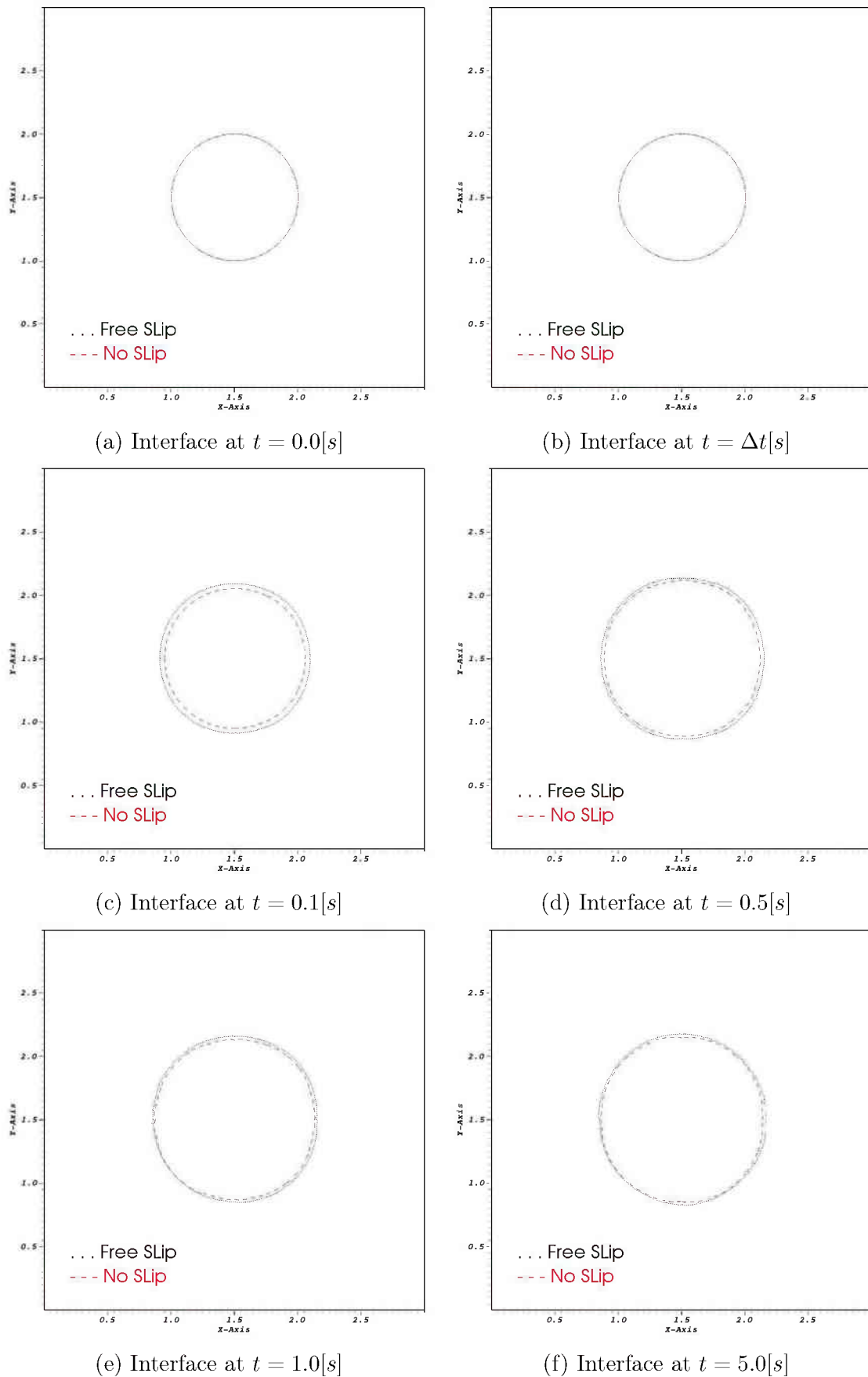
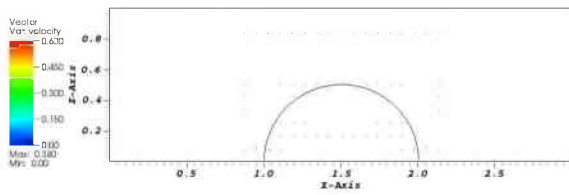
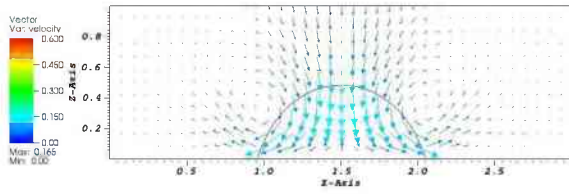


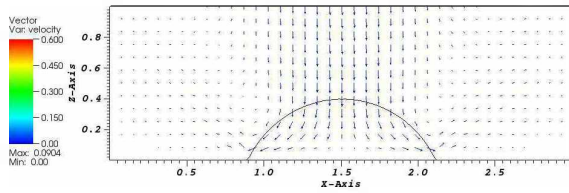
Figure 4.80: Top view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions



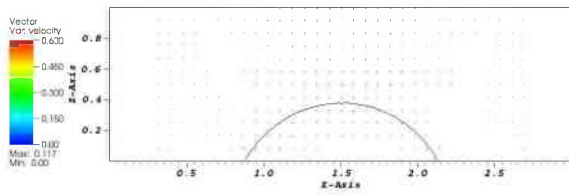
(a) $t = \Delta t[s]$



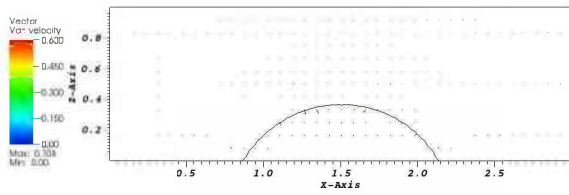
(b) $t = 0.1[s]$



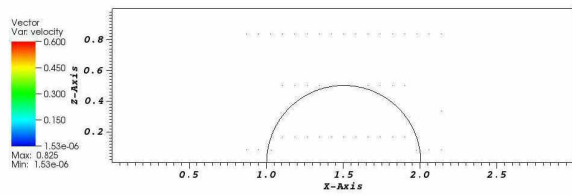
(c) $t = 0.5[s]$



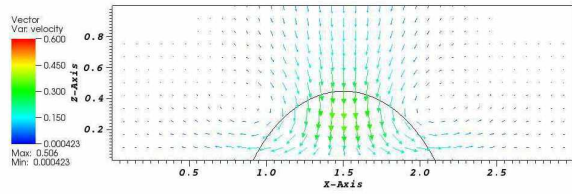
(d) $t = 1.0[s]$



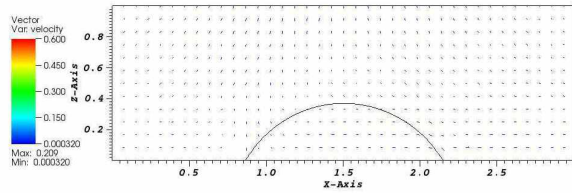
(e) $t = 5.0[s]$



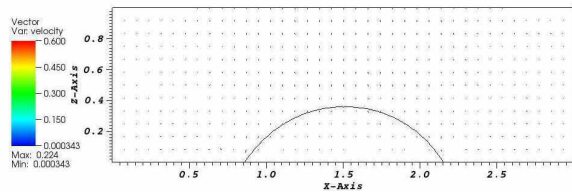
(a) $t = \Delta t[s]$



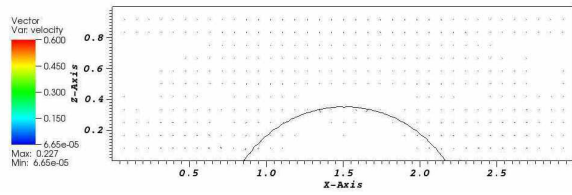
(b) $t = 0.1[s]$



(c) $t = 0.5[s]$



(d) $t = 1.0[s]$



(e) $t = 5.0[s]$

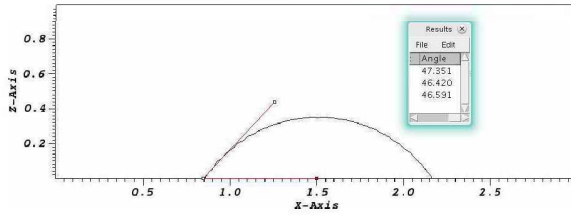
Figure 4.81: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-Slip boundary condition.

Figure 4.82: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-Slip boundary condition.

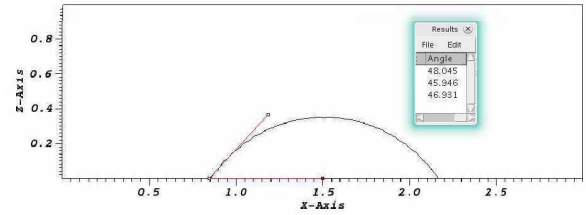
The velocity field, presented in Figs. 4.81 and 4.82, acts in the expected direction, to spread the droplet over the surface. No recirculation at the contact line is observed. The values obtained with the Free-Slip are, in general, three ($t = \Delta t[s]$) to two ($t = 0.1[s]$) times bigger than with the No-Slip, and explains why the interface moves quicker to the desired

position.

The numerically measured contact angles are presented in Figs. 4.83 and 4.84, and are equal to $\theta_N = 47.0^\circ$ for the No-Slip boundary condition and $\theta_N = 46.9^\circ$ for the Free-Slip boundary condition. The error between the numerical and the equilibrium angles are equal to $\epsilon_\theta = 4.5\%$ and $\epsilon_\theta = 4.3\%$ (No-Slip and Free-Slip boundary condition, respectively).

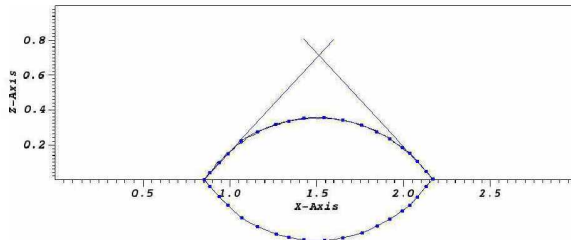


(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 46.8^\circ$ for the No-Slip B.C.

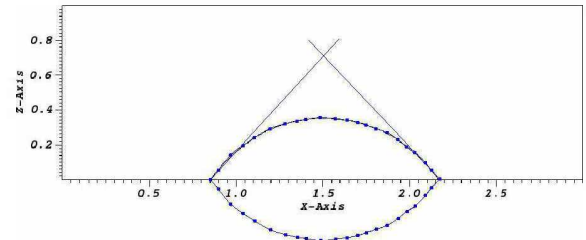


(b) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 47.0^\circ$ for the Free-Slip B.C.

Figure 4.83: contact angle measured at the end of the simulation for $\theta_E = 45^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$.



(a) DropSnake measurement: $\theta_{DSL} = 46.9^\circ$, $\theta_{DSR} = 47.4^\circ$ for the No-Slip B.C.



(b) DropSnake measurement: $\theta_{DSL} = 47.2^\circ$, $\theta_{DSR} = 46.6^\circ$ for the Free-Slip B.C.

Figure 4.84: contact angle measured at the end of the simulation for $\theta_E = 45^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$ (DropSnake measurement).

4.2.1.4 Hydrophobic case: $\theta = 148^\circ$, base mesh $[12 \times 12 \times 4]$ and 5 levels of refinement. Comparison between Free-Slip and No-Slip boundary conditions.

For the test cases with $\theta = 148^\circ$, similar to the test cases with $\theta = 45^\circ$, the maximum grid resolution was chosen to be $\Delta = \frac{1}{64}$. From the analysis of the cases with $\theta = 60^\circ$, resumed on Tab. 4.26, the errors between the numerical and the equilibrium angles are on the same order of magnitude for the simulations with 5 and 6 levels of refinement. The time step is determined by the capillarity restriction, and is equal to $\Delta t = 7.79E - 4[s]$. The error in the initialized vof volume is $\epsilon = 2.43E - 4$.

The volume conservation is presented in Fig. 4.67, being on machine error order for both simulations.

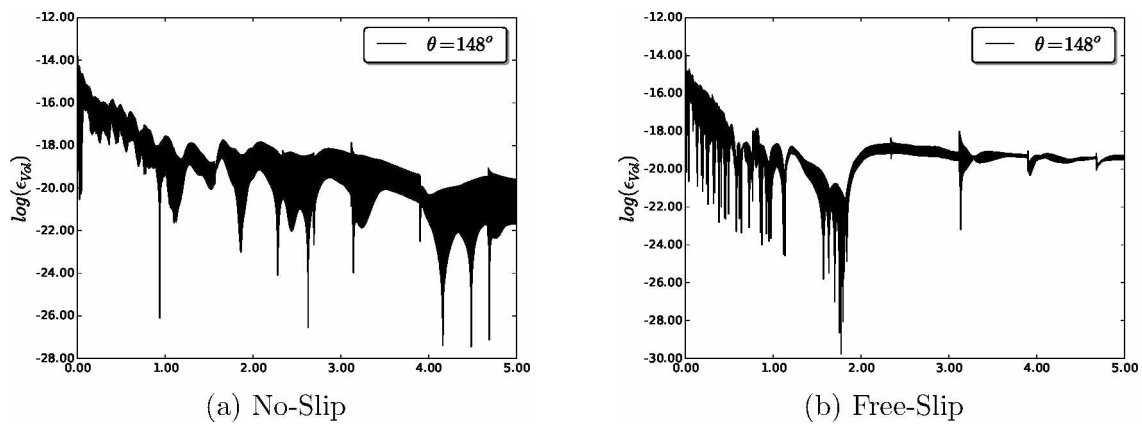


Figure 4.85: \log of the error in volume conservation (ϵ_{vol}) vs *time* for mesh with maximum resolution $\Delta = \frac{1}{64}$.

Figures 4.86 and 4.87 present the time evolution of the interface ($x - z$ slice) and the Eulerian mesh used, for the No-Slip and the Free-Slip cases, respectively. The interface advances to the desired angle, and no irregularities are observed at the end of the simulation. From the top view, presented in Fig. 4.89, it can be seen that the interface spherical shape is maintained at the bottom, as well as its symmetry.

A comparison between the interface evolution for the two boundary conditions can be seen in Figs. 4.88 (side view) and 4.89 (Top view). It can be observed that the final position achieved is the same, but with the Free-Slip the interface advances faster to the final position.

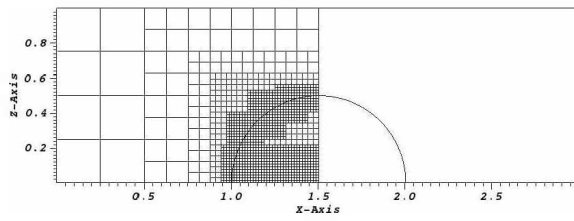
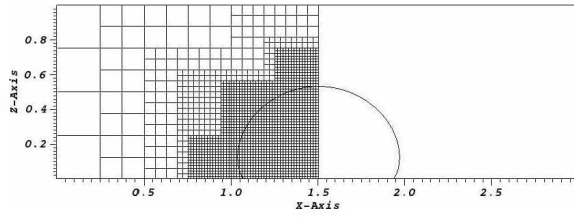
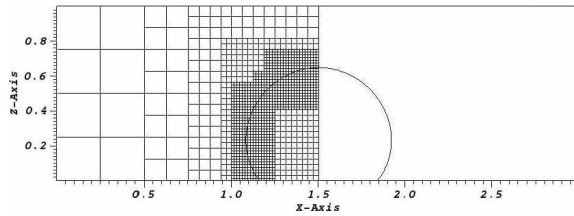
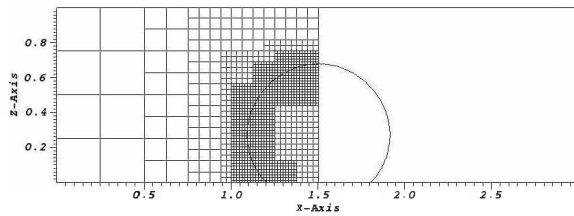
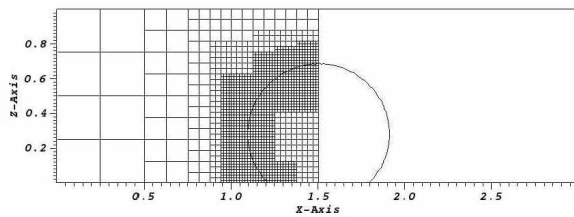
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.86: x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 148^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-slip boundary condition.

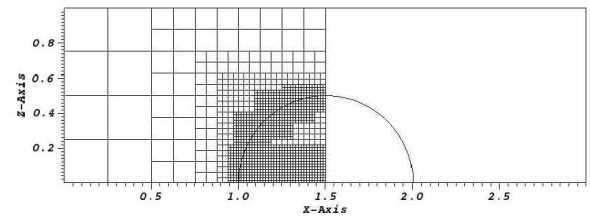
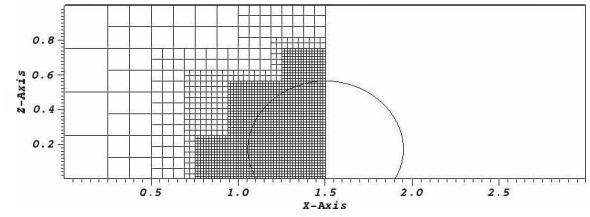
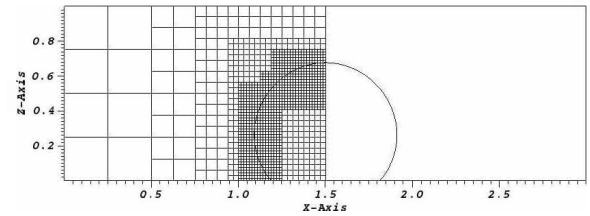
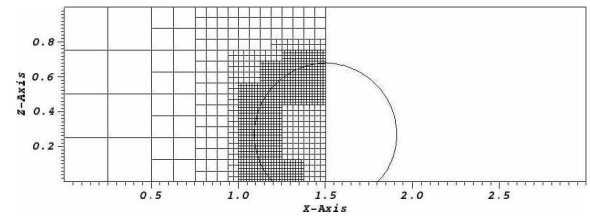
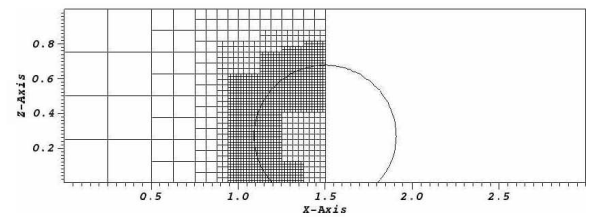
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.87: x-z plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 148^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-slip boundary condition.

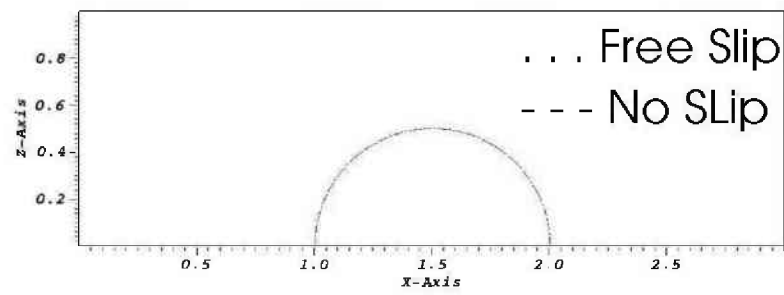
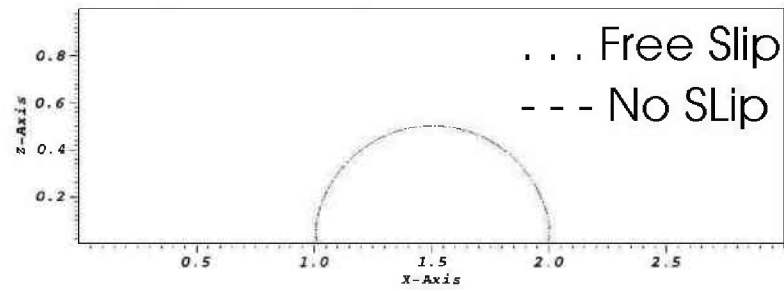
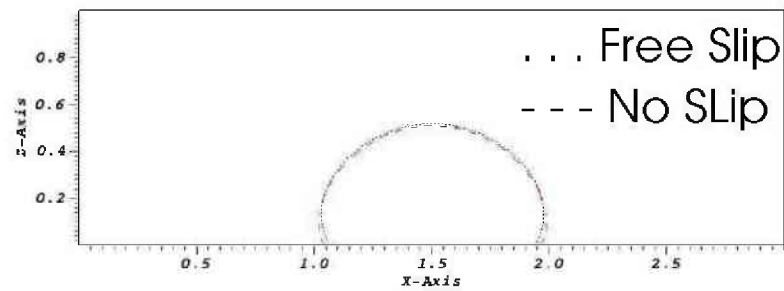
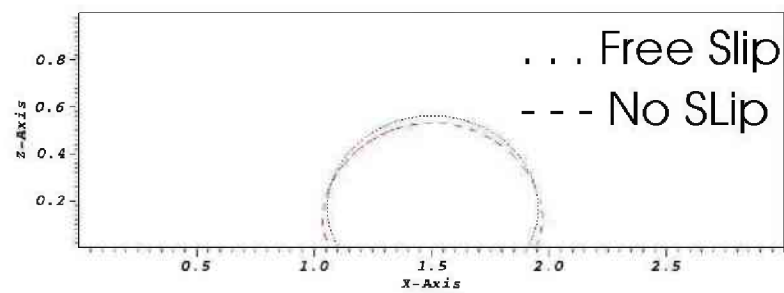
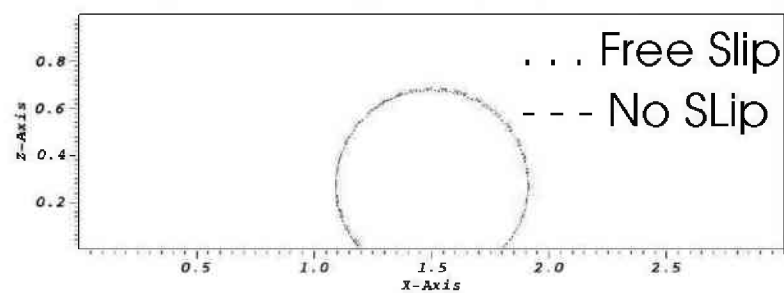
(a) Interface at $t = \Delta t[s]$ (b) Interface at $t = 0.1[s]$ (c) Interface at $t = 0.5[s]$ (d) Interface at $t = 1.0[s]$ (e) Interface at $t = 5.0[s]$

Figure 4.88: Side view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions

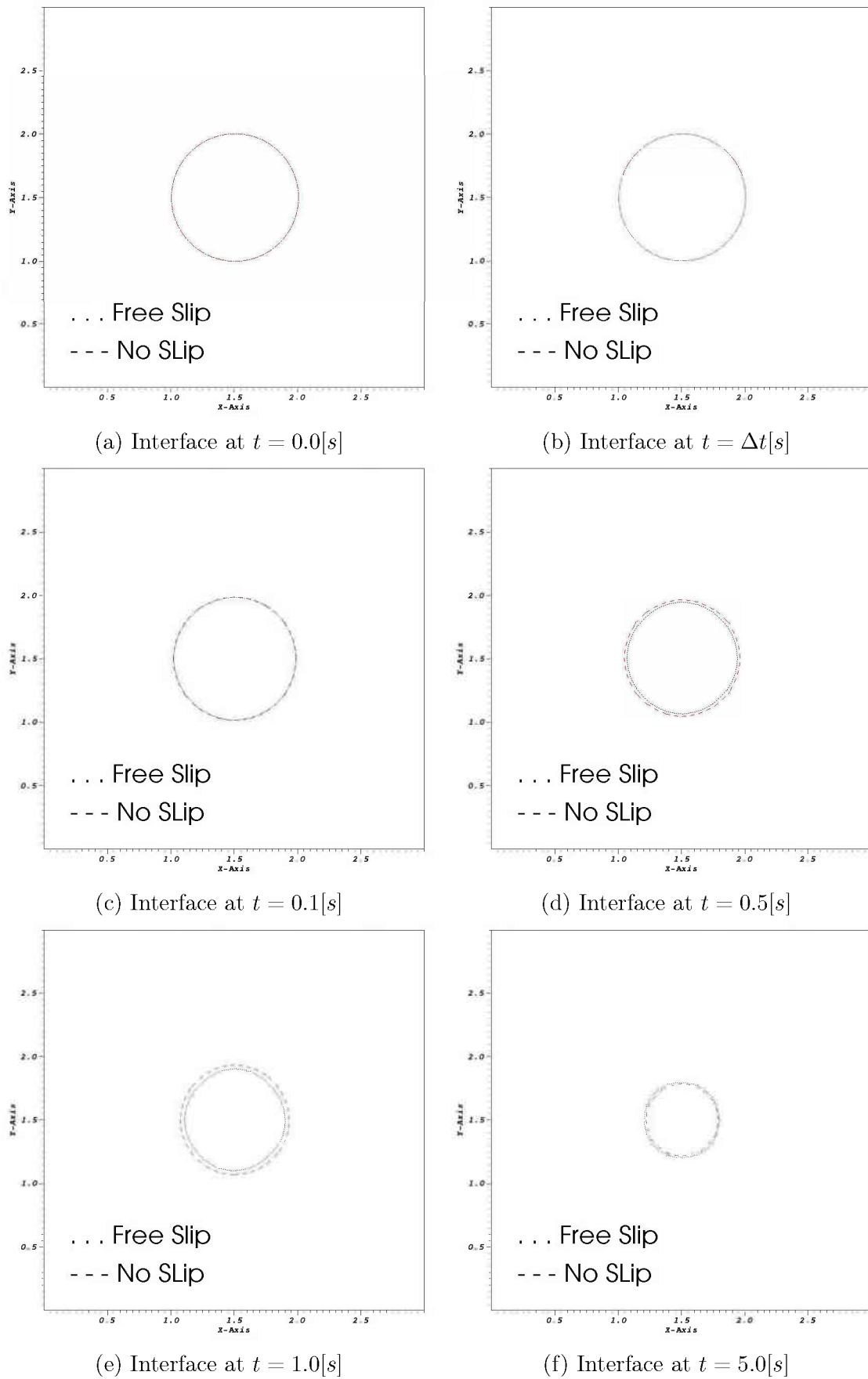
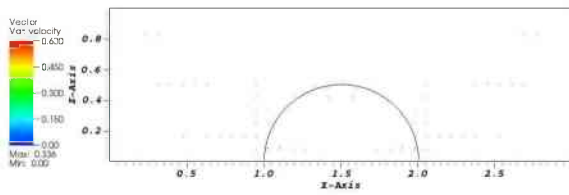
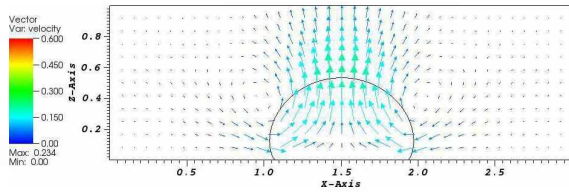


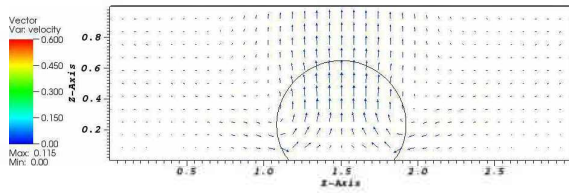
Figure 4.89: Top view of the time evolution of the interface with No-Slip and Free-Slip boundary conditions



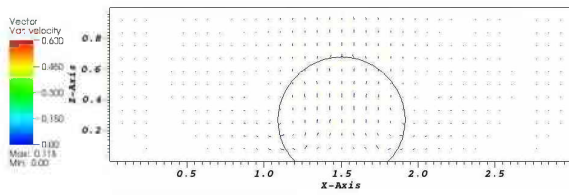
(a) $t = \Delta t[s]$



(b) $t = 0.1[s]$



(c) $t = 0.5[s]$

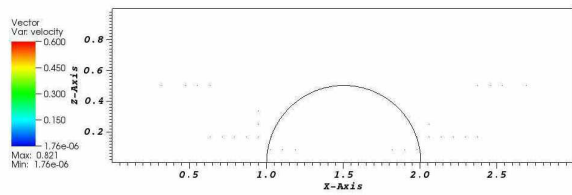


(d) $t = 1.0[s]$

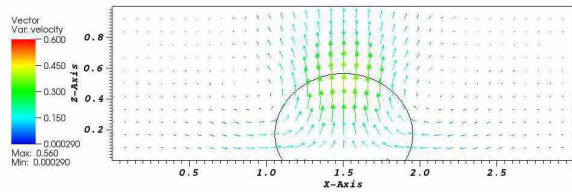


(e) $t = 5.0[s]$

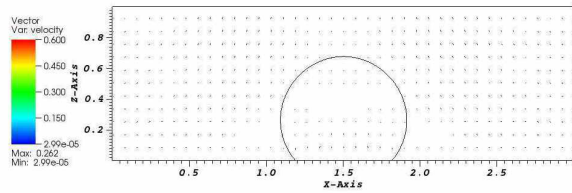
Figure 4.90: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. No-Slip boundary condition.



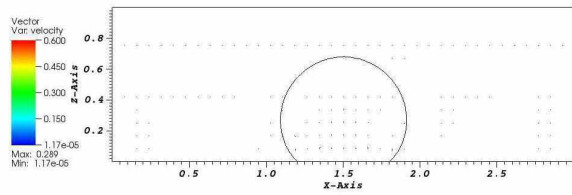
(a) $t = \Delta t[s]$



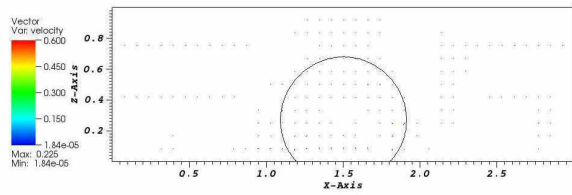
(b) $t = 0.1[s]$



(c) $t = 0.5[s]$



(d) $t = 1.0[s]$

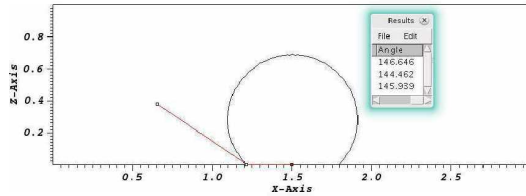


(e) $t = 5.0[s]$

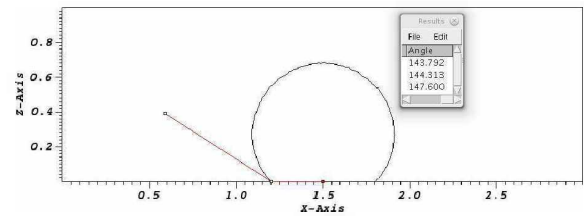
Figure 4.91: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 148^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$. Free-Slip boundary condition.

The velocity field, presented in Figs. 4.90 and 4.91, acts in the expected direction, to reduce the contact area of the droplet on the surface. No recirculation at the contact line is observed. The values obtained with the Free-Slip are, in general two times bigger than with the No-Slip, and explains why the interface moves quicker to the desired position.

The numerically measured contact angles are presented in Figs. 4.92 and 4.93, and are equal to $\theta_N = 146.8^\circ$ for both the No-Slip and the Free-Slip boundary conditions, with the error between the numerical and the equilibrium angles being equal to $\epsilon_\theta = 0.8\%$.

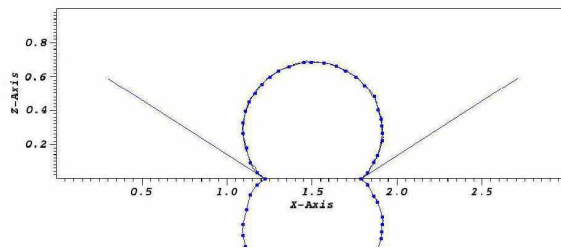


(a) ImageJ averaged measured contact angle $\theta_{IJ} = 145.7^\circ$ for the No-Slip B.C.

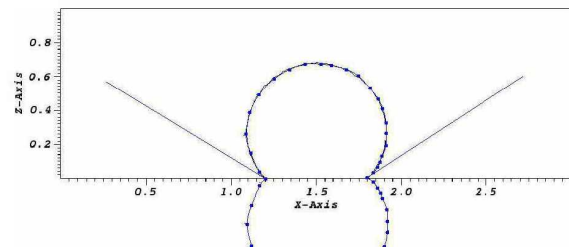


(b) ImageJ averaged measured contact angle $\theta_{IJ} = 145.2^\circ$ for the Free-Slip B.C.

Figure 4.92: contact angle measured at the end of the simulation for $\theta_E = 148^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$.



(a) DropSnake measurement: $\theta_{DSL} = 147.3^\circ$, $\theta_{DSR} = 147.3^\circ$ for the No-Slip B.C.



(b) DropSnake measurement: $\theta_{DSL} = 147.7^\circ$, $\theta_{DSR} = 147.7^\circ$ for the Free-Slip B.C.

Figure 4.93: contact angle measured at the end of the simulation for $\theta_E = 148^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$ (DropSnake measurement).

4.2.2 *3D contact point study with density ratio $\rho_l/\rho_g = 800$ and viscosity ratio $\mu_l/\mu_g = 100$, based on the 2D work of Afkhami and Bussmann (2009).*

A series of tests for surface tension-driven flow, with $30^\circ \leq \theta \leq 150^\circ$ is presented by Afkhami and Bussmann (2009). A drop on a solid surface initially at equilibrium is suddenly imposed to a different contact angle; the drop fluid should accelerate toward a steady state defined by the new value of θ .

The setup consists of an initially hemispherical drop ($\theta = 90^\circ$) at $t = 0[s]$, of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$. The density ratio is $\rho_l/\rho_g = 800$, the viscosity ratio is $\mu_l/\mu_g = 100$, surface tension is $\sigma = 0.1[N/m]$, and $Oh = 1.33 \times 10^{-2}$. No-slip boundary condition is specified at the bottom of the domain and an open boundary condition is specified elsewhere.

Accordingly to the authors, contact line slip is achieved implicitly, as the advection scheme used to advect volume fractions utilizes face-centered velocities, so that the nearest velocity to the contact line is one half cell width above a solid boundary.

The computations are performed in 4 processors with an adaptative mesh of base $[16 \times 16 \times 16]$ and 3 levels of refinement (maximum grid resolution of $\Delta = 1/64$). The *Crank-Nicholson Adams-Bashforth (CNAB)* is used for the temporal discretization, and the *Cubista* for the advection model.

The time step is constant and equal to $\Delta t = 1.5E - 4[s]$. The error in the initialized vof volume is $\epsilon = 6.12E - 4$.

4.2.2.1 *Hydrophilic case: $\theta = 30^\circ$, base mesh $[16 \times 16 \times 16]$ and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.*

Figures 4.94 and 4.95 present the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used. The results obtained with the *amr3d* code are smoother than the ones presented by Afkhami and Bussmann (2009), where the contact point quickly advances to the desired angle and the middle of the droplet advances later (see Fig. 12 from the authors). From the top view it can be seen that the interface symmetry is maintained.

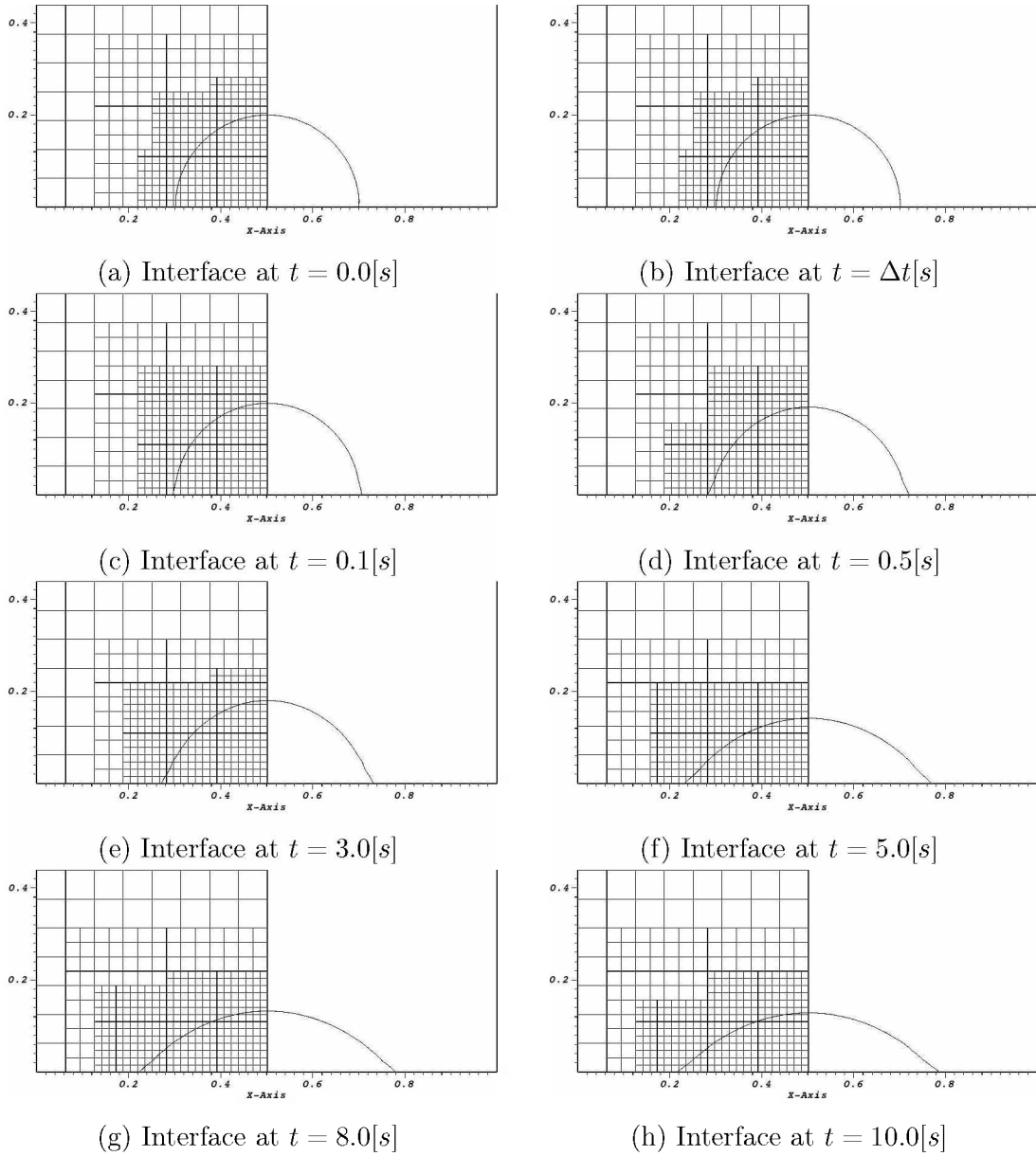


Figure 4.94: x-z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 30^\circ$

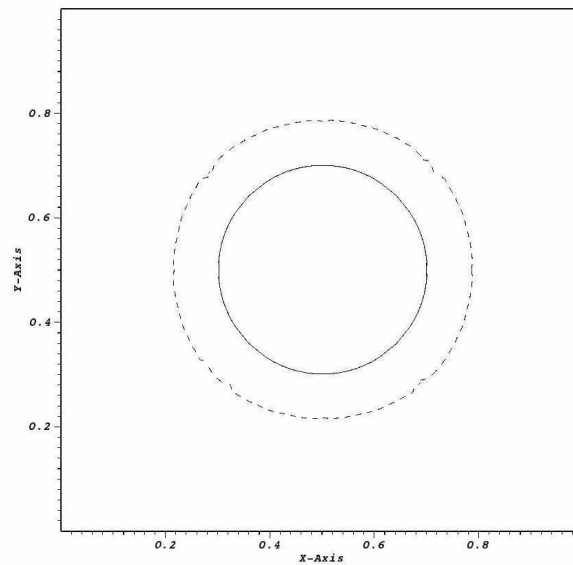


Figure 4.95: Top view (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)

The velocity field, presented in Fig. 4.96, acts in the expected direction, to spread the droplet over the surface. From time $t = 5.0[s]$ a small recirculation is observed at the contact point. The contact point force should vanish as the equilibrium state is approximated, but the velocity field will still act in the direction of the spreading. At this instant the droplet should achieve a instantaneous contact angle smaller than the equilibrium one and a force at the contact point should appears in the opposing direction of this movement. This is the cause of the recirculation that appears (observed at time $t = 8.0[s]$).

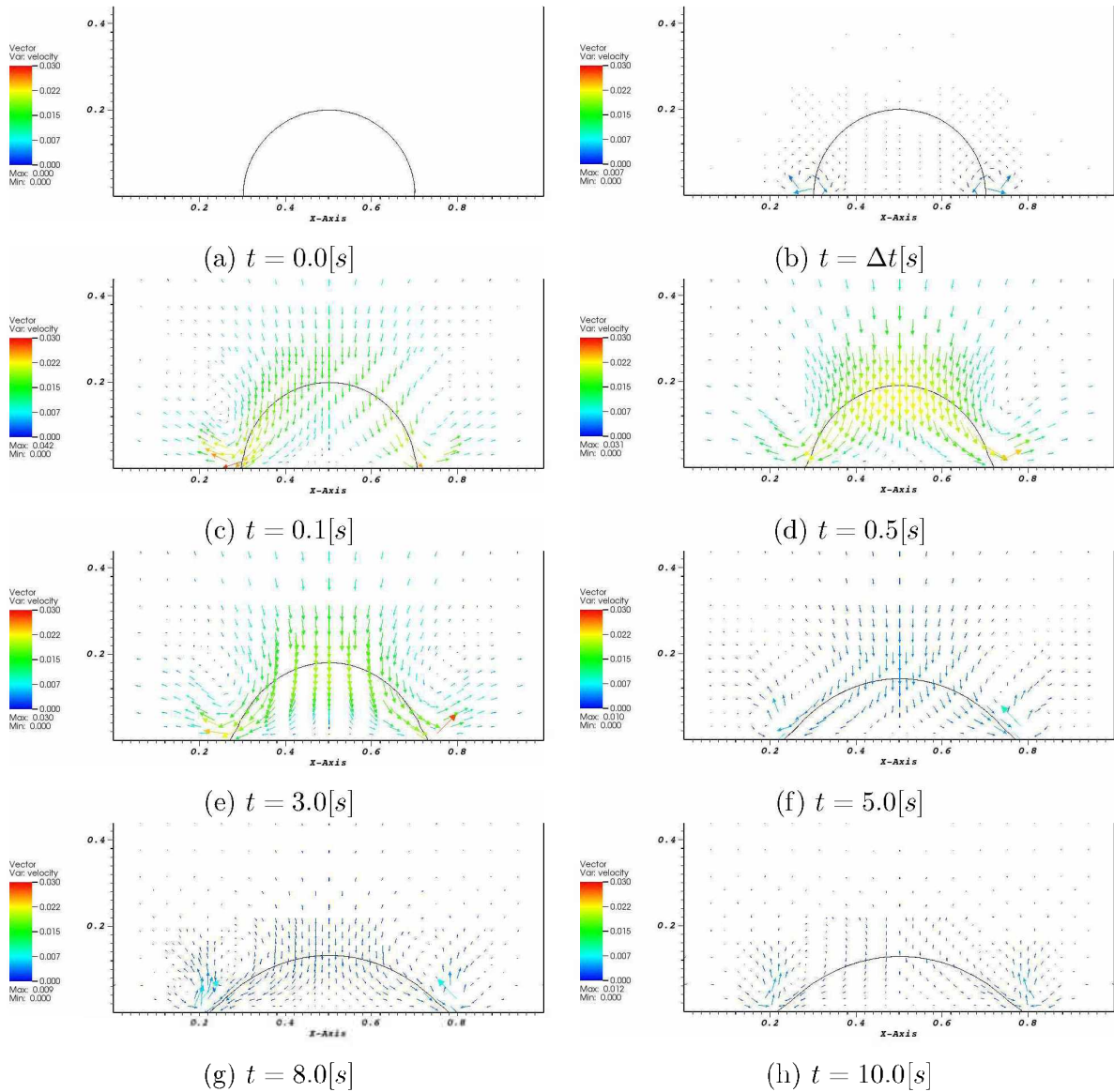
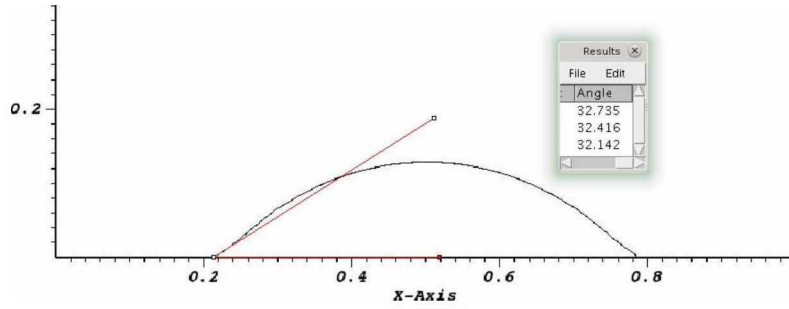
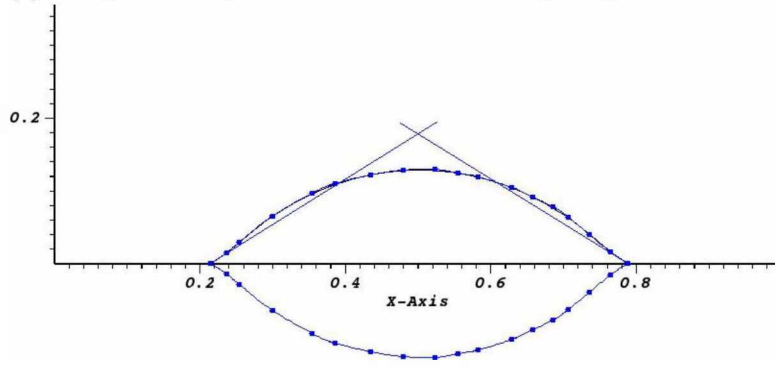


Figure 4.96: x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 30^\circ$.

The numerically measured contact angle is presented in Fig. 4.97 being equal to $\theta_N = 32.0^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 6.7\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 32.4^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 31.9^\circ$, $\theta_{DSR} = 31.7^\circ$

Figure 4.97: $\theta_N = 32.0^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 30^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.98 and is at machine error for all simulation.

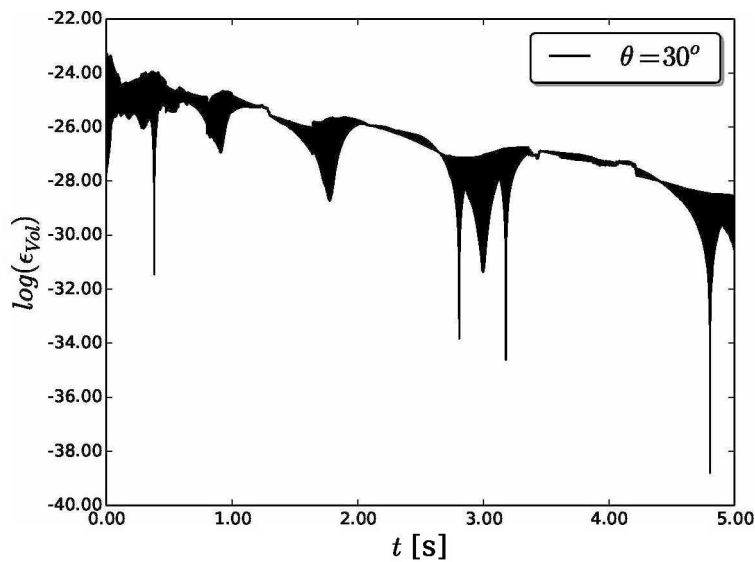


Figure 4.98: log of the error in volume conservation (ϵ_{vol}) vs time for $\theta = 30^\circ$

4.2.2.2 *Hydrophilic case: $\theta = 45^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.*

Figures 4.99 and 4.100 present the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used. Similar to the case with $\theta = 30^\circ$, the results obtained with the amr3d code are smoother than the ones presented by Afkhami and Bussmann (2009), where the contact point quickly advances to the desired angle and the middle of the droplet advances later (see Fig. 12 from the authors). From the top view it can be seen that the interface symmetry is maintained.

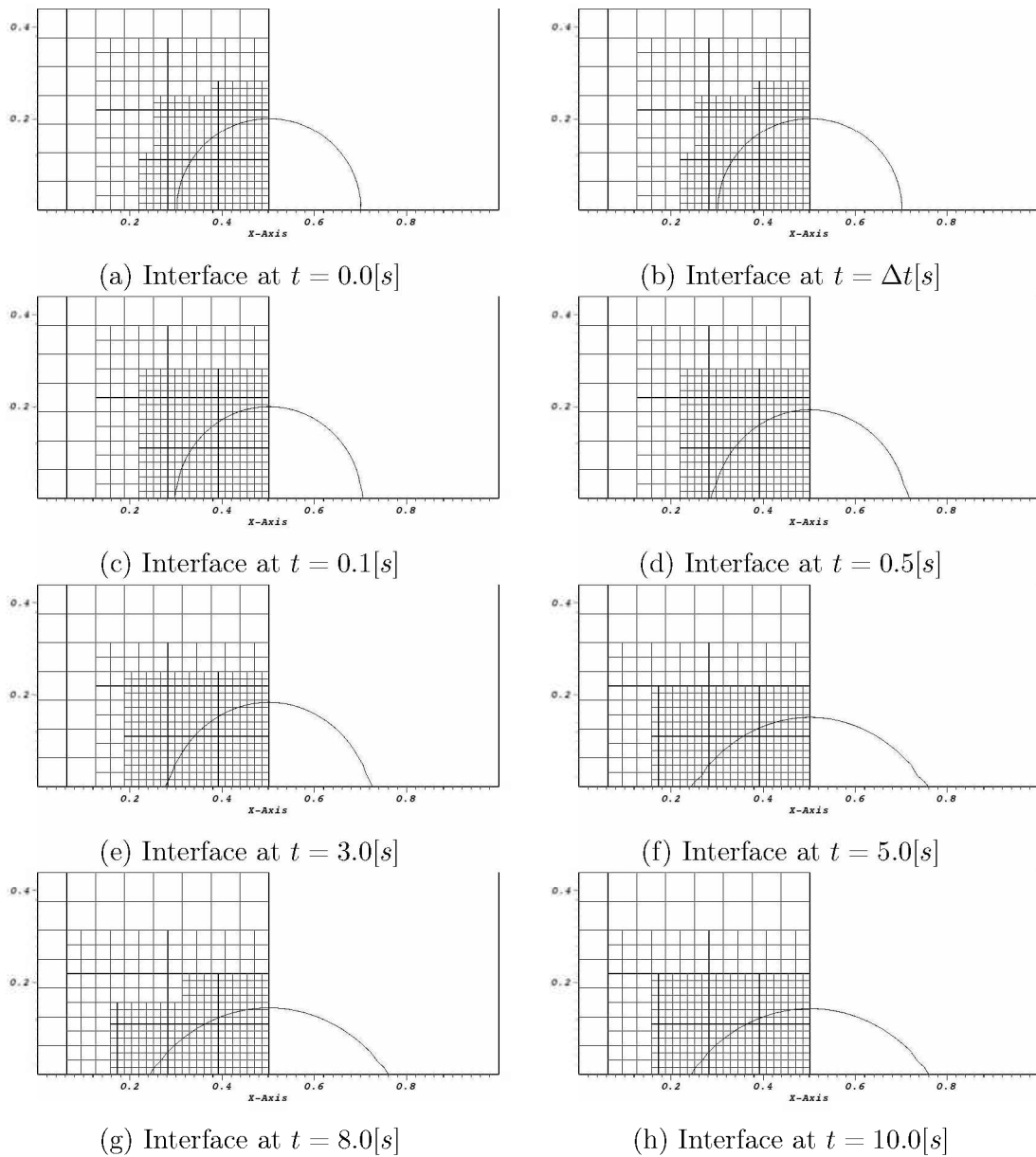


Figure 4.99: x - z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$

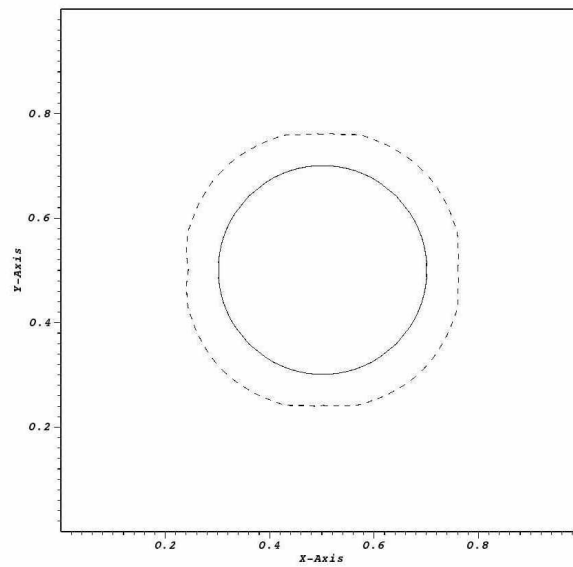


Figure 4.100: Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)

The velocity field, presented in Fig. 4.101, acts in the expected direction, to spread the droplet over the surface. Similar to the $\theta = 30^\circ$ case, from time $t = 5.0[s]$ a small recirculation is observed at the contact point. Since the difference between the initial and the desired contact angles are smaller in this case, the equilibrium position is achieved faster, and so the velocity recirculation is smaller.

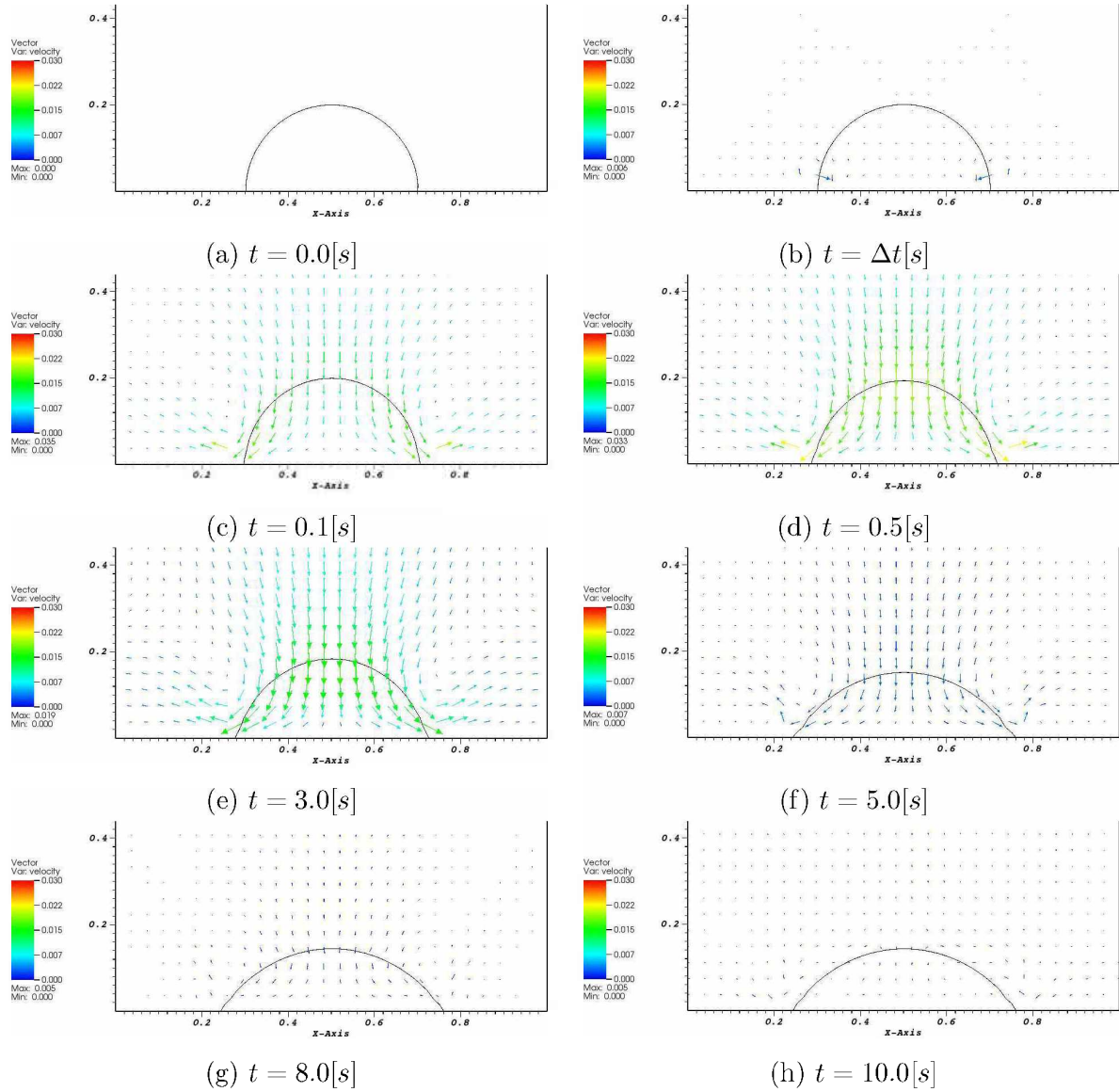
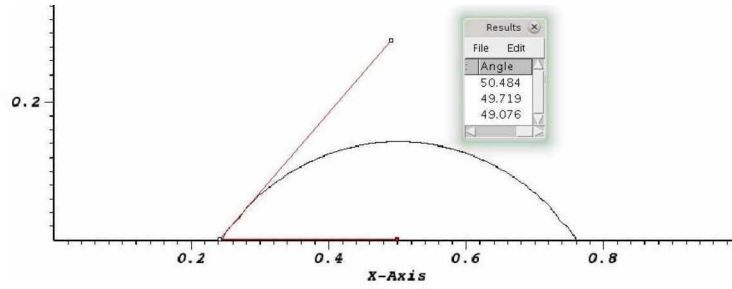
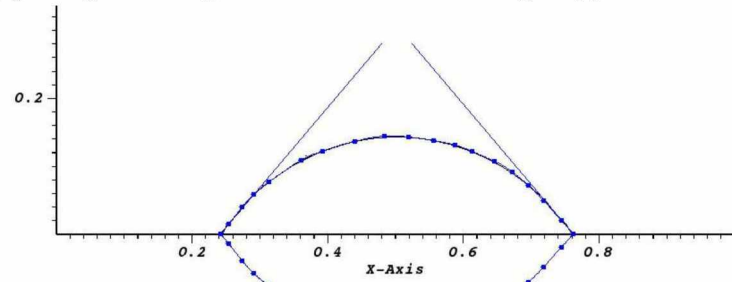


Figure 4.101: x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 45^\circ$.

The numerically measured contact angle is presented in Fig. 4.102 being equal to $\theta_N = 49.6^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 10.3\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 49.7^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 49.6^\circ$, $\theta_{DSR} = 49.6^\circ$

Figure 4.102: $\theta_N = 49.6^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 45^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.103, being at machine error for all simulation.

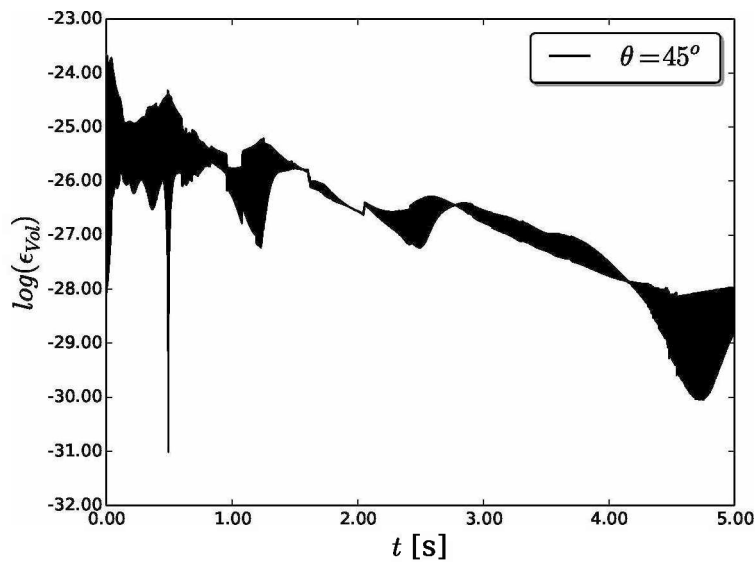


Figure 4.103: log of the error in volume conservation (ϵ_{vol}) vs time for $\theta = 45^\circ$

4.2.2.3 *Hydrophilic case: $\theta = 60^\circ$, base mesh $[16 \times 16 \times 16]$ and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.*

Figures 4.104 and 4.105 present the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used. Similar to the cases with $\theta = 30^\circ$ and $\theta = 45^\circ$, the results obtained with the amr3d code are smoother than the ones presented by Afkhami and Bussmann (2009), where the contact point quickly advances to the desired angle and the middle of the droplet advances later (see Fig. 12 from the authors). From the top view it can be seen that the interface symmetry is maintained.

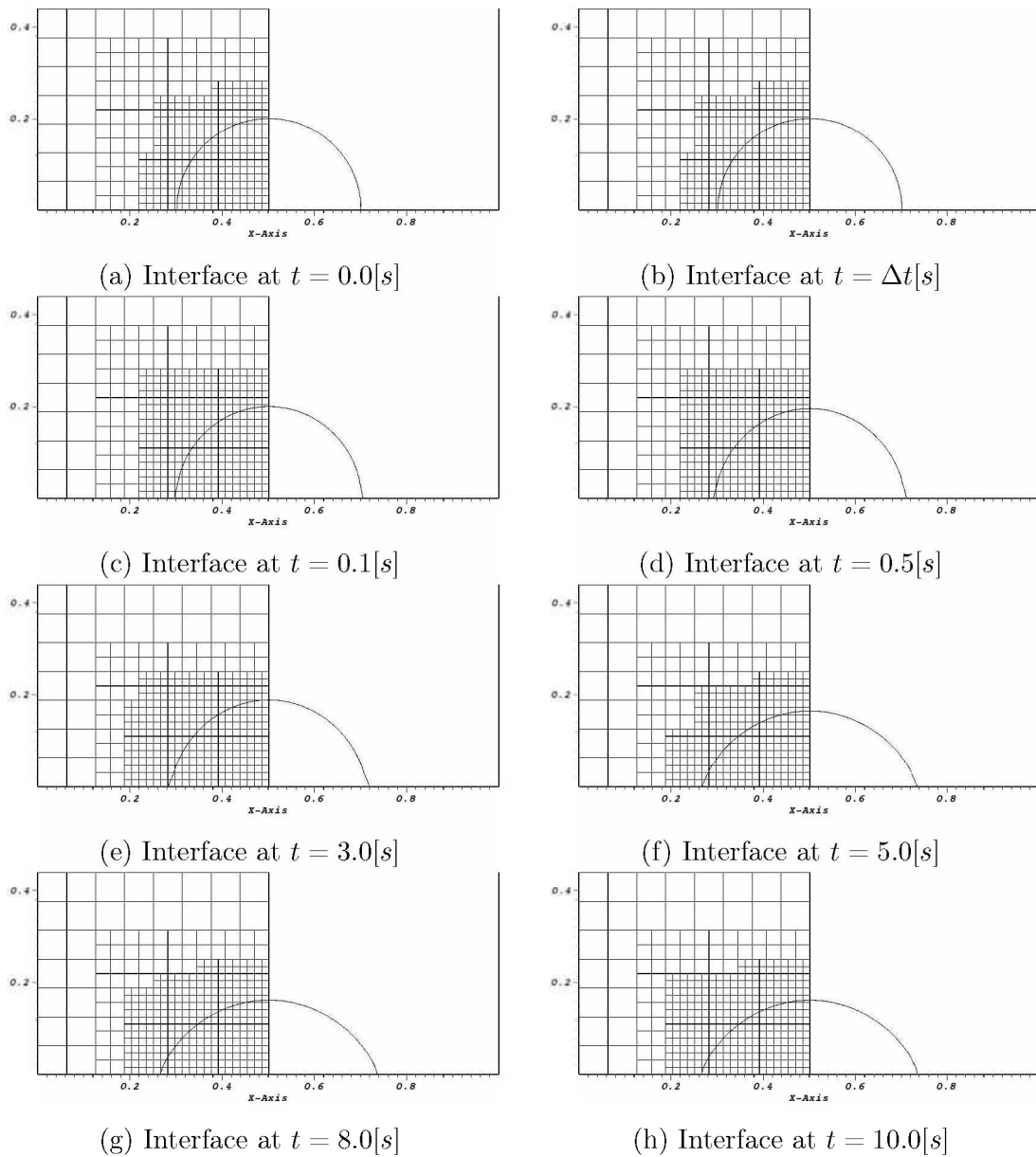


Figure 4.104: $x-z$ plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$

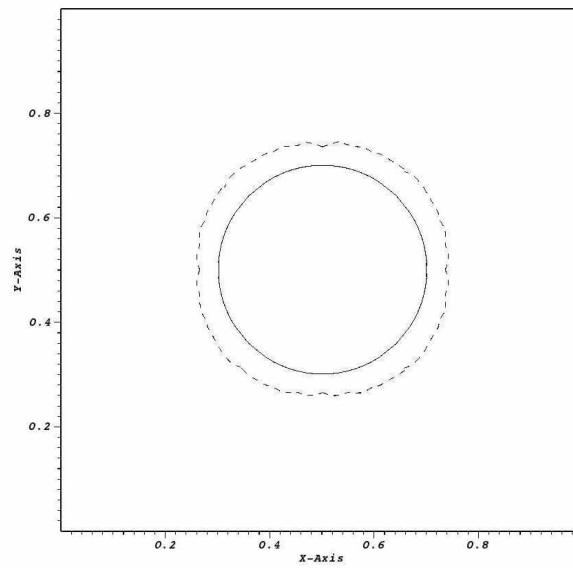


Figure 4.105: Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)

The velocity field, presented in Fig. 4.106, acts in the expected direction, to spread the droplet over the surface. Its magnitude is smaller than the $\theta = 45^\circ$ case, and almost no recirculation is observed at the contact point.

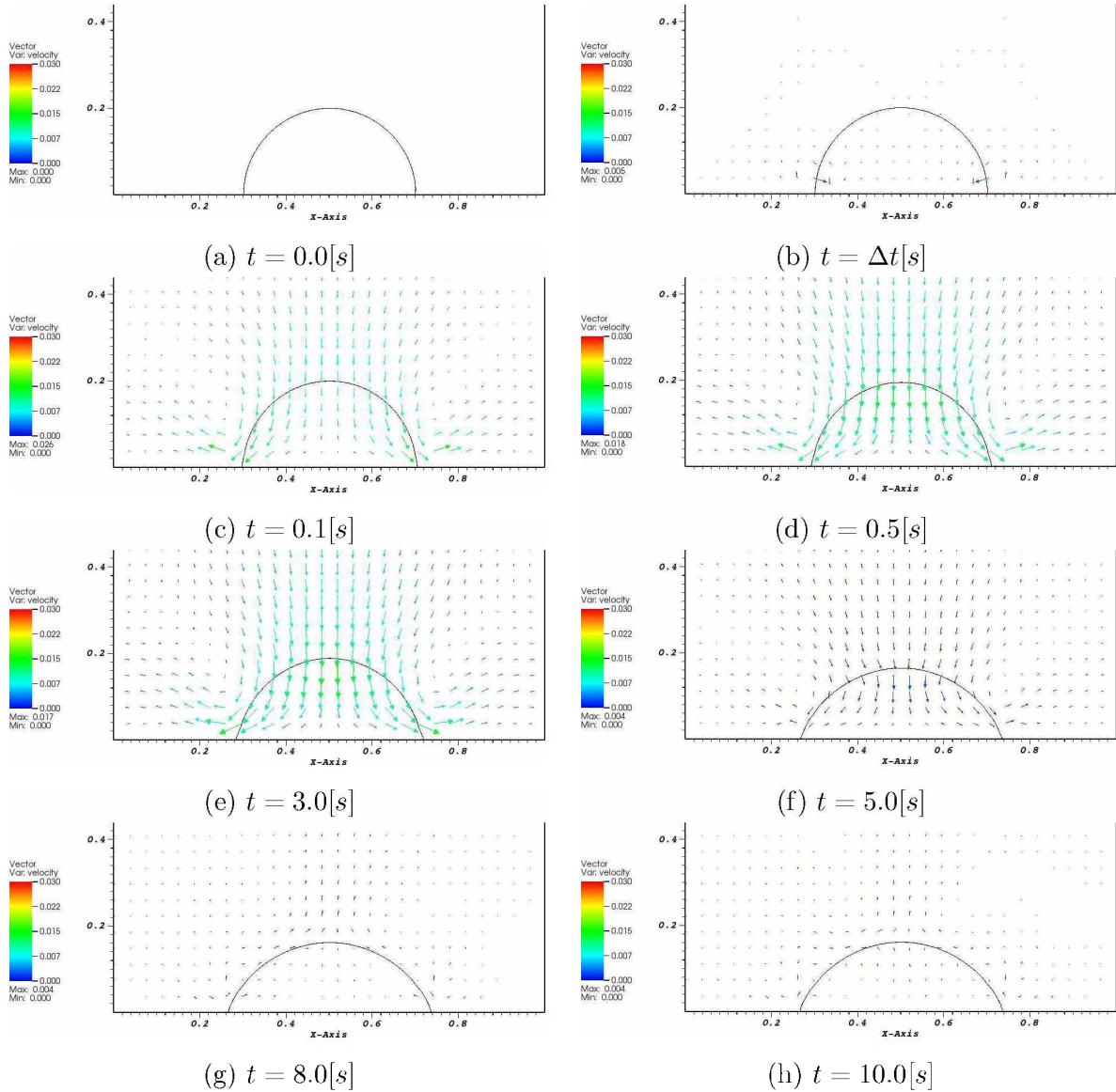


Figure 4.106: x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$.

The numerically measured contact angle is presented in Fig. 4.107 being equal to $\theta_N = 61.6^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 2.6\%$.

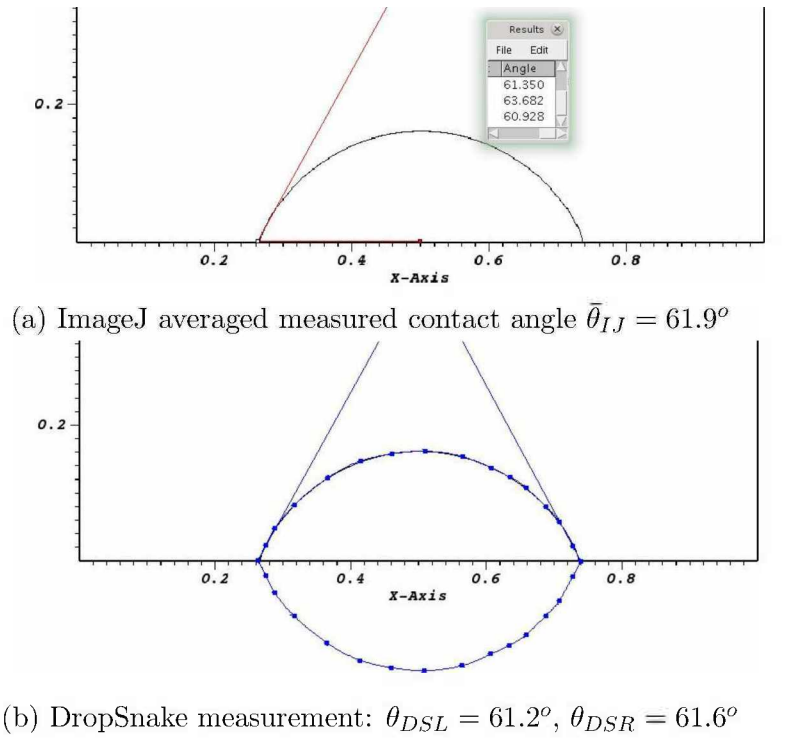


Figure 4.107: $\theta_N = 61.6^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 60^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.108, being at machine error for all simulation.

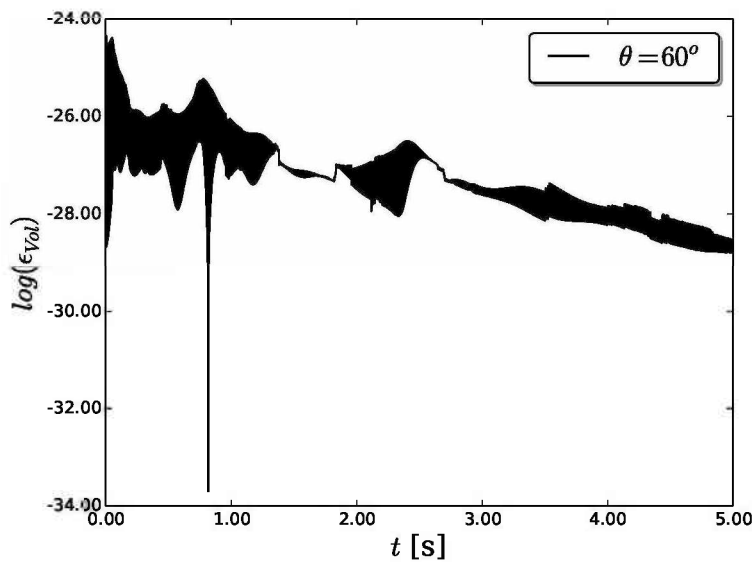


Figure 4.108: log of the error in volume conservation (ϵ_{vol}) vs time for $\theta = 60^\circ$

4.2.2.4 *Hydrophobic case: $\theta = 120^\circ$, base mesh $[16 \times 16 \times 16]$ and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.*

Figures 4.109 and 4.110 present the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used. Similar to the hydrophilic cases, the results obtained with the amr3d code are smoother than the ones presented by Afkhami and Bussmann (2009), where the contact point quickly advances to the desired angle and the middle of the droplet advances later (see Fig. 13 from the authors). From the top view it can be seen that the interface symmetry is maintained.

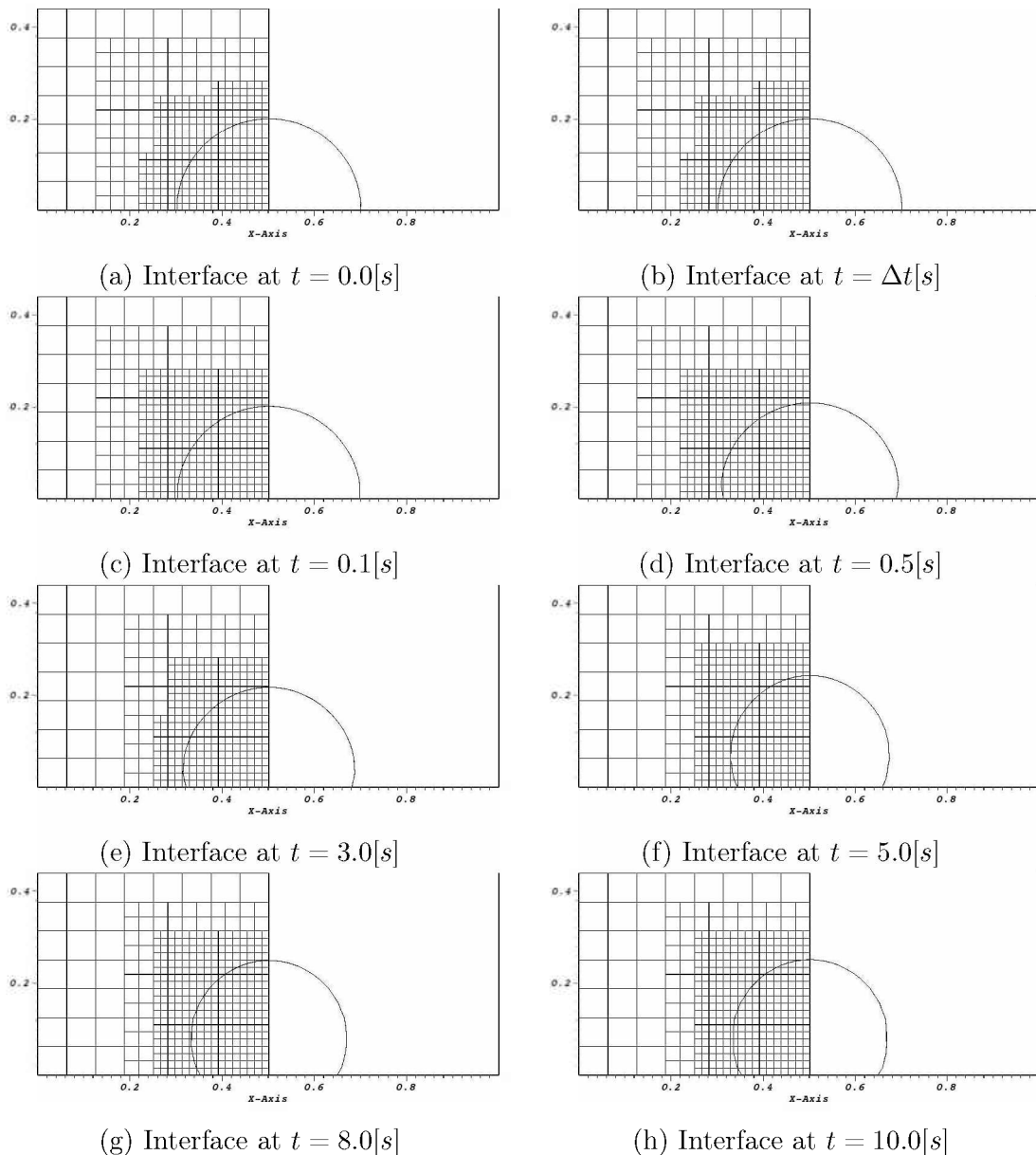


Figure 4.109: $x-z$ plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 120^\circ$

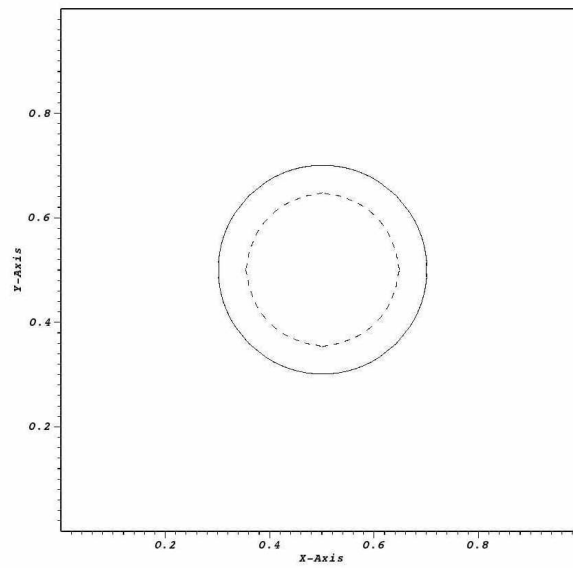


Figure 4.110: Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)

The velocity field, presented in Fig. 4.111, acts in the expected direction, to reduce the area of the droplet over the surface.

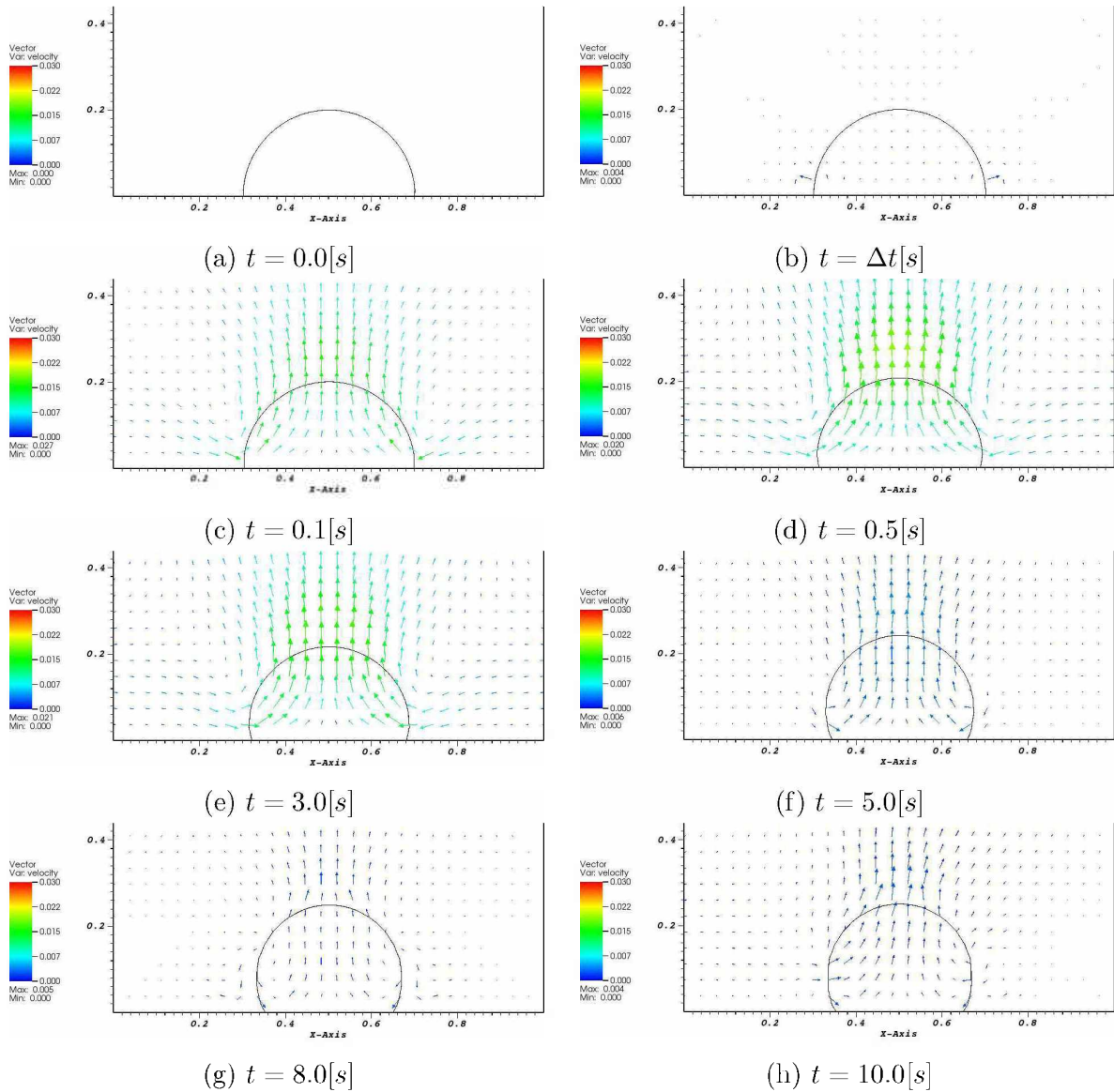
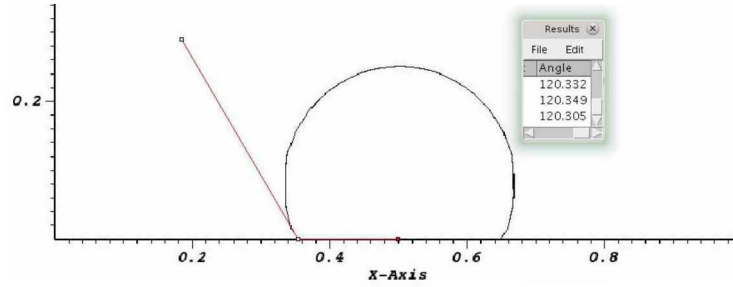
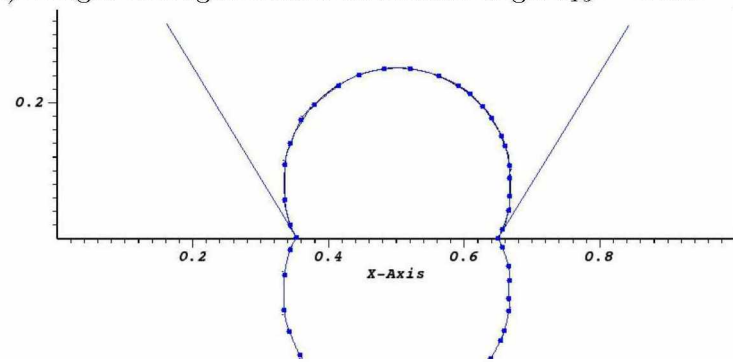


Figure 4.111: x - z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 120^\circ$.

The numerically measured contact angle is presented in Fig. 4.112 being equal to $\theta_N = 121.1^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 0.92\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 120.3^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 121.6^\circ$, $\theta_{DSR} = 121.4^\circ$

Figure 4.112: $\theta_N = 121.1^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 120^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.113, being at machine error for all simulation.

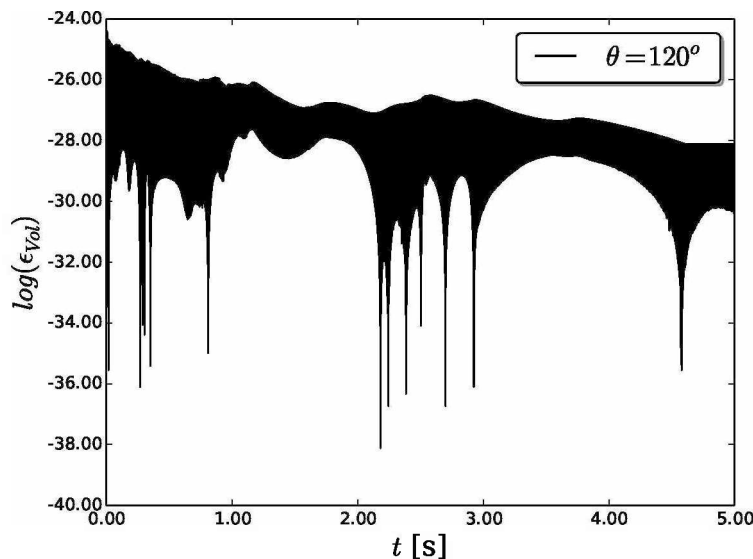


Figure 4.113: log of the error in volume conservation (ϵ_{vol}) vs time for $\theta = 120^\circ$

4.2.2.5 *Hydrophobic case: $\theta = 135^\circ$, base mesh [16 x 16 x 16] and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.*

Figures 4.114 and 4.115 present the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used. Different to what was observed by Afkhami and Bussmann (2009) (see Fig. 13 from the authors), in the amr3d code, for $\theta > 120^\circ$, the drop does not jump off the surface.

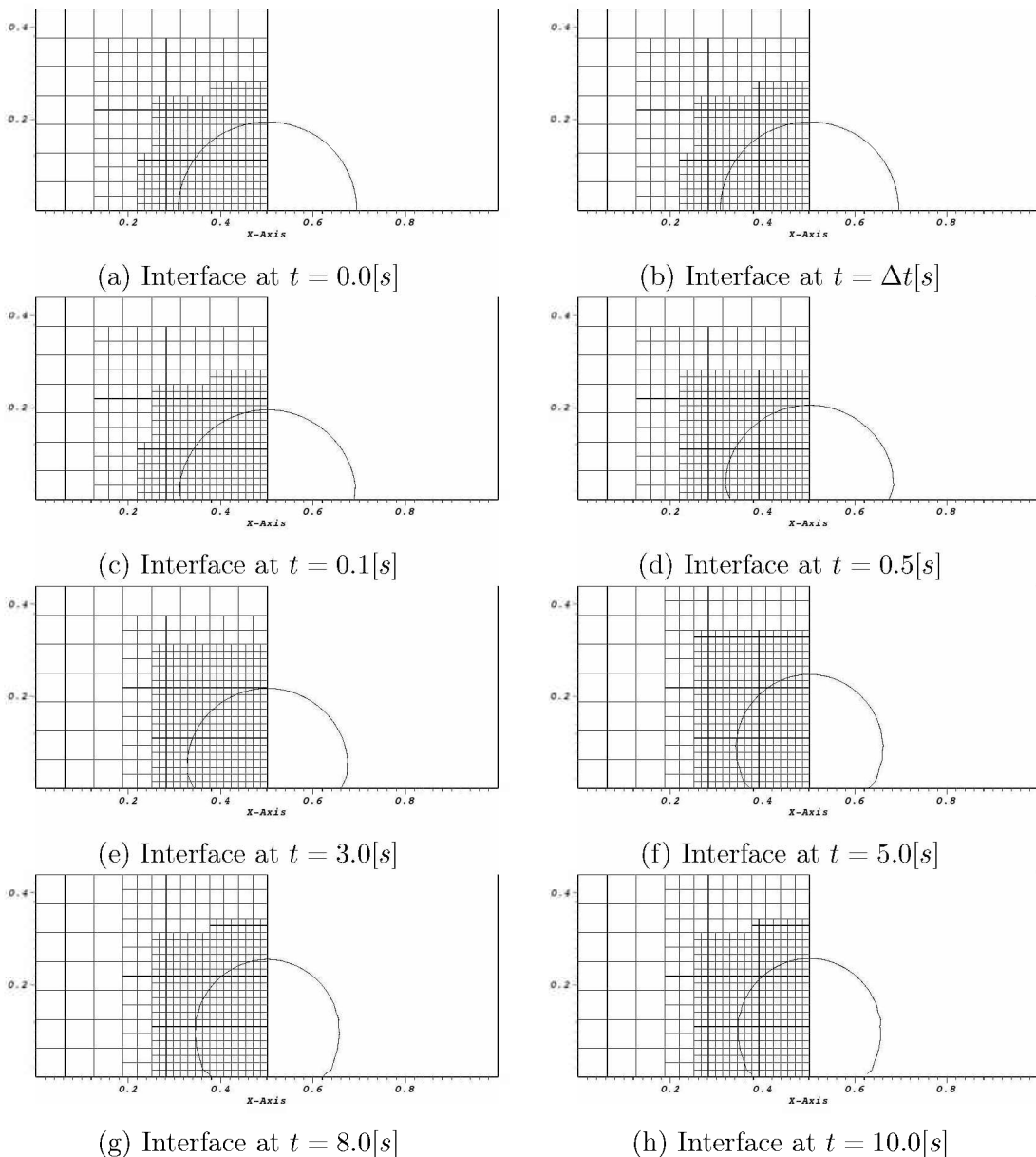


Figure 4.114: $x-z$ plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 135^\circ$

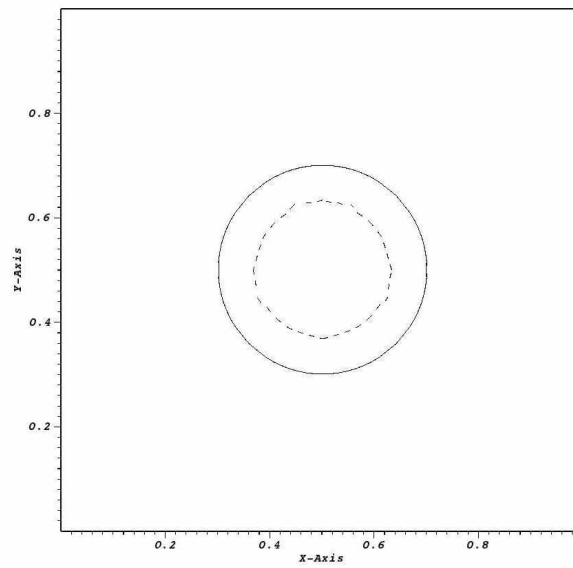


Figure 4.115: Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)

The velocity field, presented in Fig. 4.116, acts in the expected direction, to reduce the area of the droplet over the surface. As expected, its magnitude is bigger than with $\theta = 120^\circ$, especially at the beginning of the simulation.

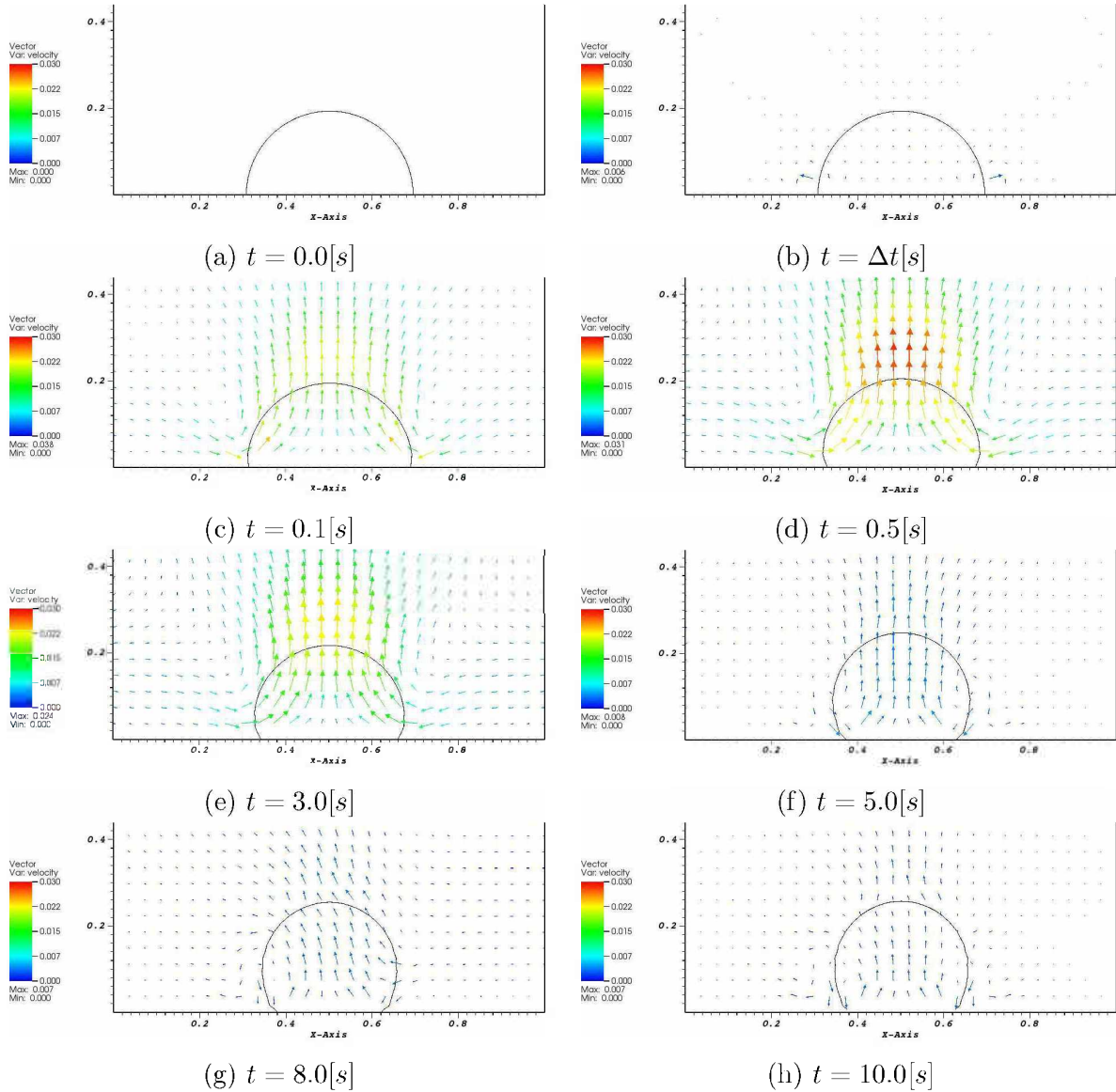
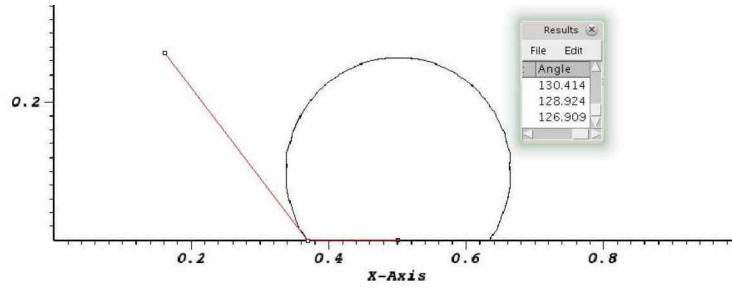
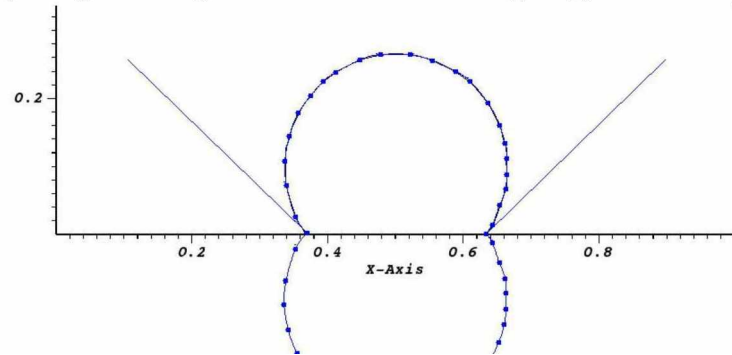


Figure 4.116: x-z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 135^\circ$.

The numerically measured contact angle is presented in Fig. 4.117 being equal to $\theta_N = 133.5^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 1.1\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 128.7^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 136.3^\circ$, $\theta_{DSR} = 135.6^\circ$

Figure 4.117: $\theta_N = 133.5^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 135^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.118, being at machine error for all simulation.

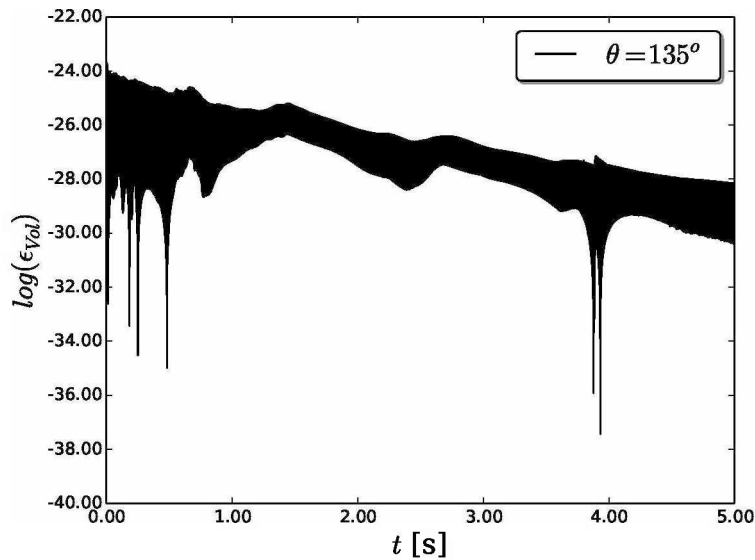


Figure 4.118: log of the error in volume conservation (ϵ_{vol}) vs time for $\theta = 135^\circ$

4.2.2.6 *Hydrophobic case: $\theta = 150^\circ$, base mesh $[16 \times 16 \times 16]$ and 3 levels of refinement, $\Delta t = 1.5E - 4[s]$. No-Slip boundary condition.*

Figures 4.119 and 4.120 present the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used. As already obtained for $\theta = 135^\circ$, and different to what was observed by Afkhami and Bussmann (2009), in the amr3d code, for $\theta > 120^\circ$, the drop does not jump off the surface.

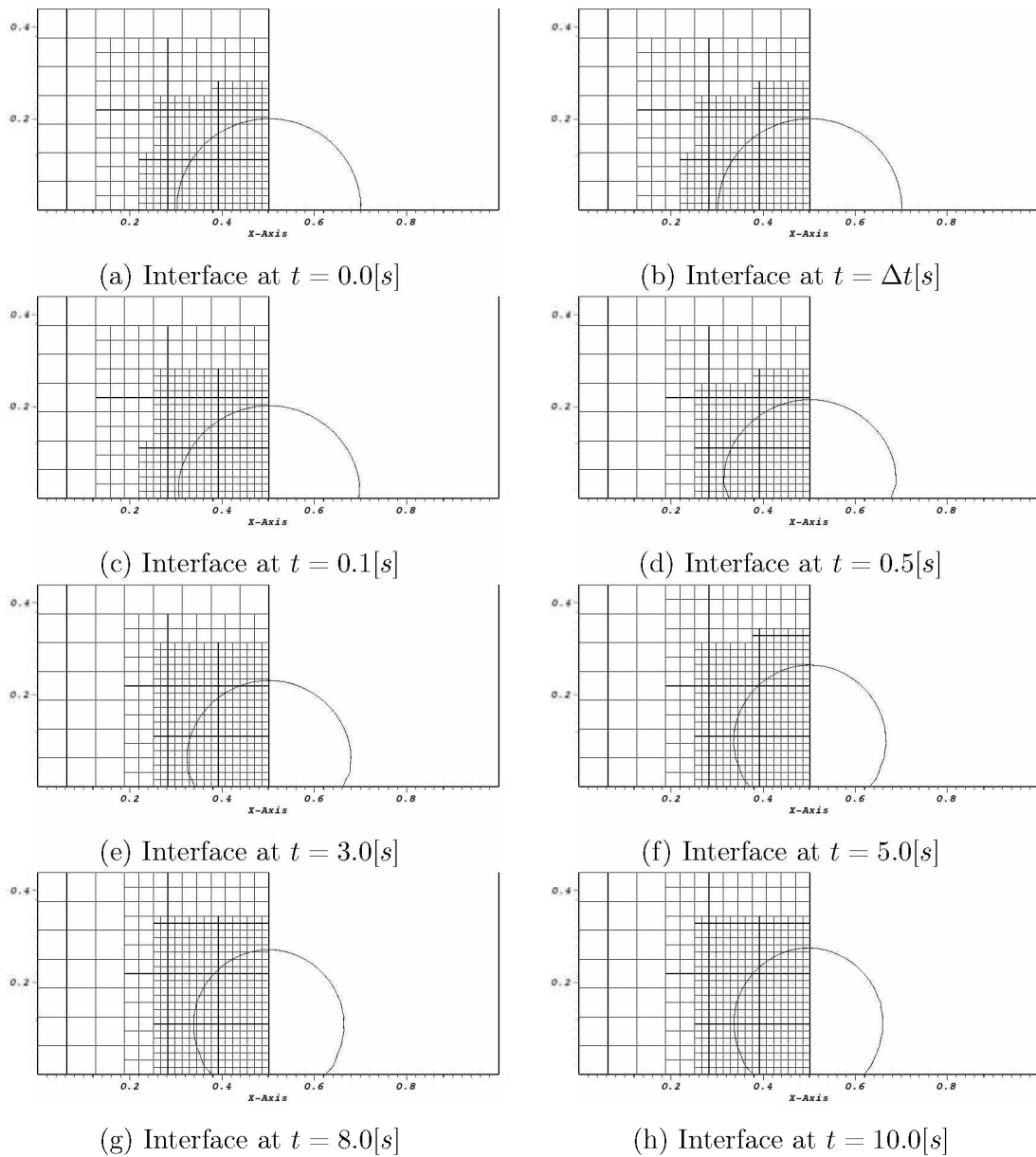


Figure 4.119: x - z plane view ($y = 0.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 150^\circ$

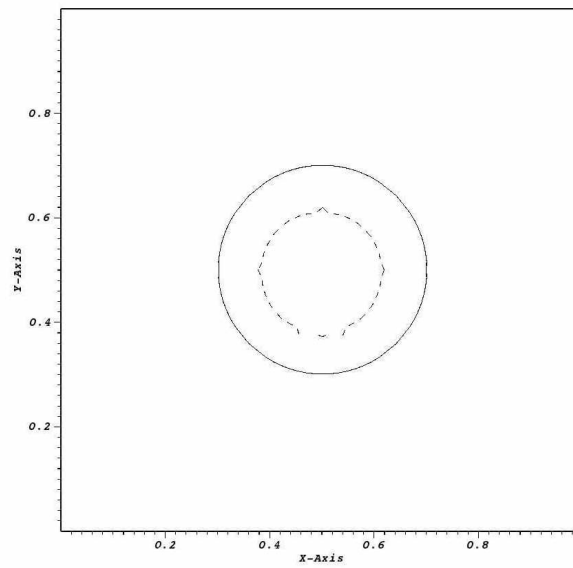


Figure 4.120: Top View (x-y plane) at time $t = 0.0[s]$ (continuous line) and $t = 10.0[s]$ (dashed line)

The velocity field, presented in Fig. 4.121, acts in the expected direction, to reduce the area of the droplet over the surface. As expected, its magnitude is bigger than with $\theta = 120^\circ$ and $\theta = 135^\circ$, especially at the beginning of the simulation. A small spurious current and recirculation is also observed.

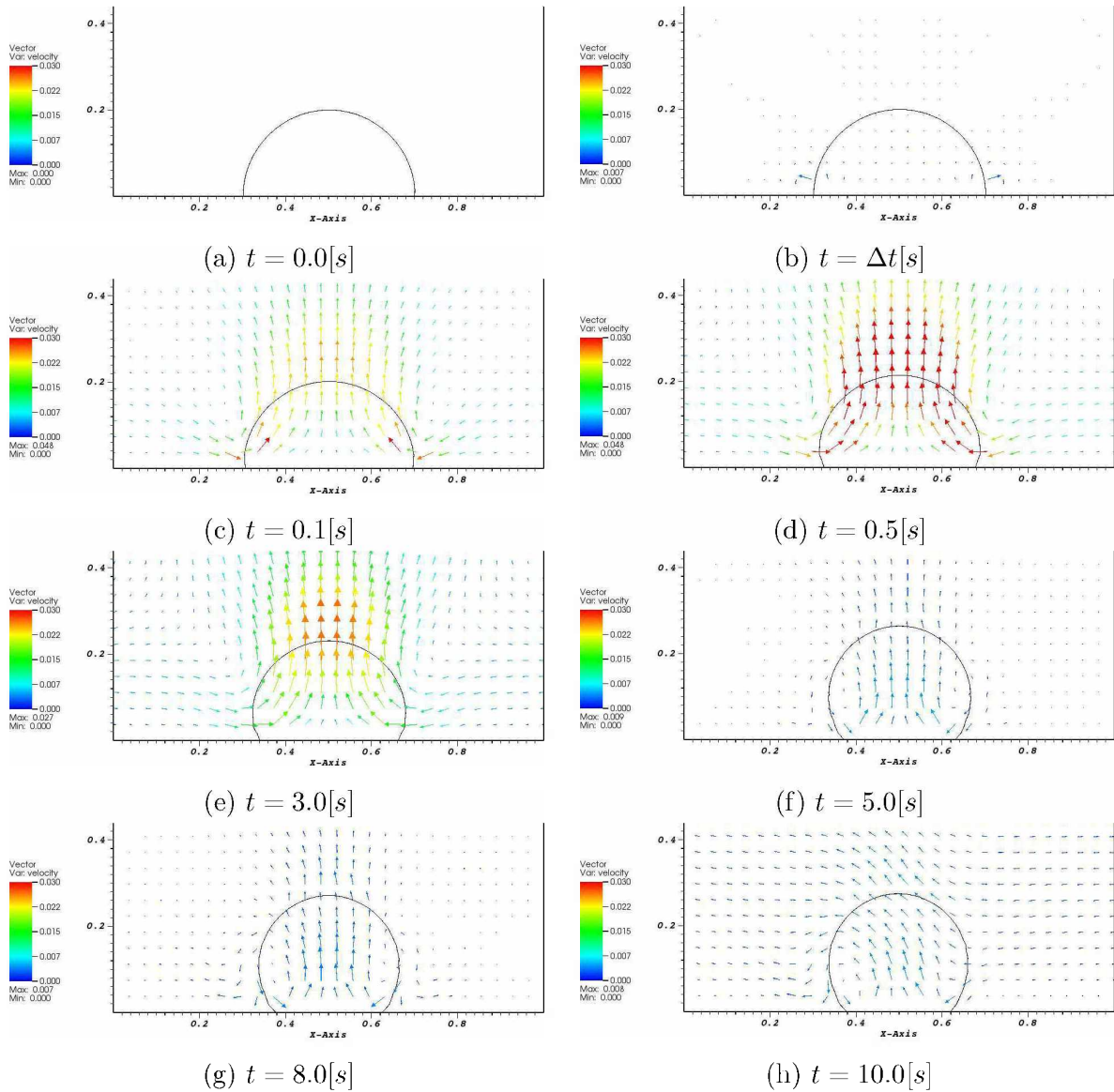
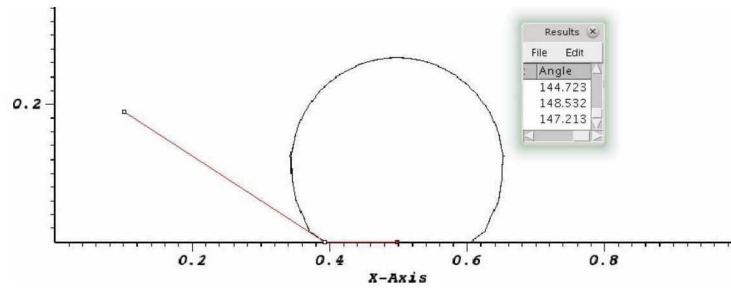
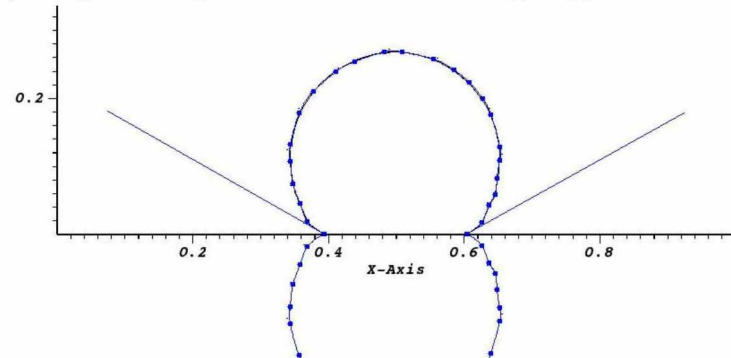


Figure 4.121: x - z plane view ($y = 0.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.2[m]$, placed at the bottom of a domain $\Omega = [1]^3[m^3]$ exposed to a sudden change in $\theta = 150^\circ$.

The numerically measured contact angle is presented in Fig. 4.122 being equal to $\theta_N = 149.4^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 0.4\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 146.8^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 150.6^\circ$, $\theta_{DSR} = 150.8^\circ$

Figure 4.122: $\theta_N = 149.4^\circ$ measured at $t = 10.00[s]$ for $\theta_E = 150^\circ$ and mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.123, being at machine error for all simulation.

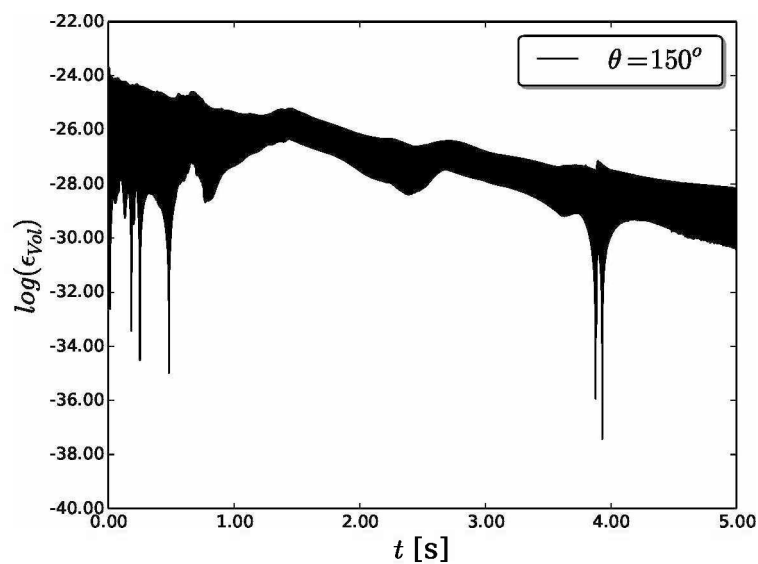


Figure 4.123: log of the error in volume conservation (ϵ_{vol}) vs time for $\theta = 150^\circ$

4.2.3 3D contact point study with density ratio $\rho_l/\rho_g = 1.0$ and viscosity ratio $\mu_l/\mu_g = 1.0$. Couple with the Immersed boundary method

Numerical experiments for the contact point model coupled with the Immersed boundary method are performed for some of the hydrophilic and hydrophobic cases presented on Section 4.2.1, which were based on the work of Lai, Tseng and Huang (2010).

A drop on a solid surface initially at equilibrium is suddenly imposed to a different contact angle; the drop fluid should accelerate toward a steady state defined by the new value of θ_E . The force at the contact point should vanish as the equilibrium contact angle is approximated.

The setup consists of an initially hemispherical drop ($\theta = 90^\circ$) at $t = 0[s]$, of radius $R = 0.5[m]$, placed at $z = 1.0$ of a domain $\Omega = [3, 3, 2][m^3]$. The immersed boundary consists of a parallelepiped with a coordinate base $x_1 = y_1 = 0.5$, $x_2 = y_2 = 2.5$, $z_1 = 0.6$ and $h = 0.4$ ($z_2 = 1.0$).

The density ratio is $\rho_l/\rho_g = 1$, the viscosity ratio is $\mu_l/\mu_g = 1$, surface tension is $\sigma = 1[N/m]$. The gravity is not considered. For this particular case, the only force responsible for the movement of the interface is the contact point force.

The computations are performed in 4 processors, with an adaptative mesh of base $[24 \times 24 \times 16]$ and 4 levels of refinement for $\theta = 30^\circ$ and $[48 \times 48 \times 32]$ and 3 levels of refinement for $\theta = 60^\circ$ and $\theta = 135^\circ$ (maximum grid resolution of $\Delta = 1/64$). The *Semi-Backward Difference* is used for the temporal discretization, and the *Central Difference Scheme (CDS)* for the advection model.

The time step is determined by the capillarity restriction, and is equal to $\Delta t = 7.79E - 4[s]$. The error in the initialized vof volume is $\epsilon = 1.17E - 4$.

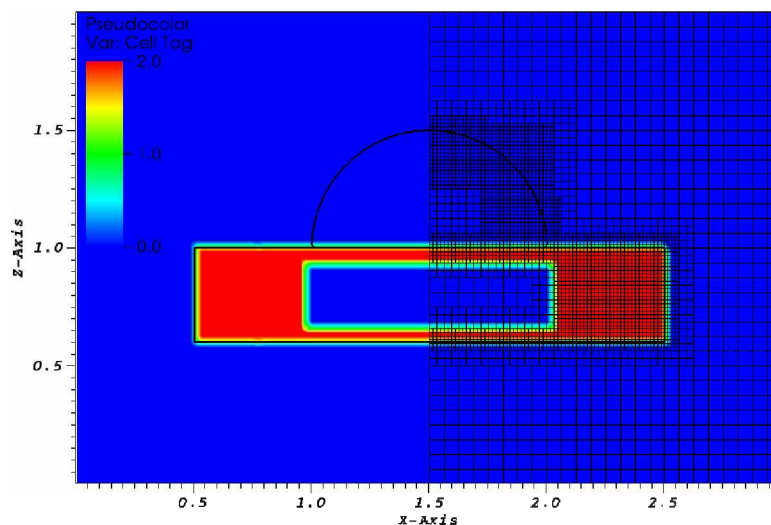


Figure 4.124: Eulerian domain with the mesh used for the simulation, the immersed boundary and the tag of the Eulerian cells: 0 = inside cells; 1 = interface cells; 2 = outside cells.

Figure 4.124 presents a $x - z$ slice view at $y = 1.5[m]$ (half of the domain) of the

Eulerian domain, the adaptative mesh, the interface and the immersed boundary at time $t = 0[s]$. The Eulerian computational cells are tagged as cells inside (blue color; tag=0), at the interface (green color; tag=1) and outside (red color; tag=2). This indicates the Eulerian cells that might or not be used by the Least Squares system for both the normal and curvature computation, as well as where the contact point model is applied.

4.2.3.1 $\theta = 30^\circ$

Figures 4.125 and 4.126 presents the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used.

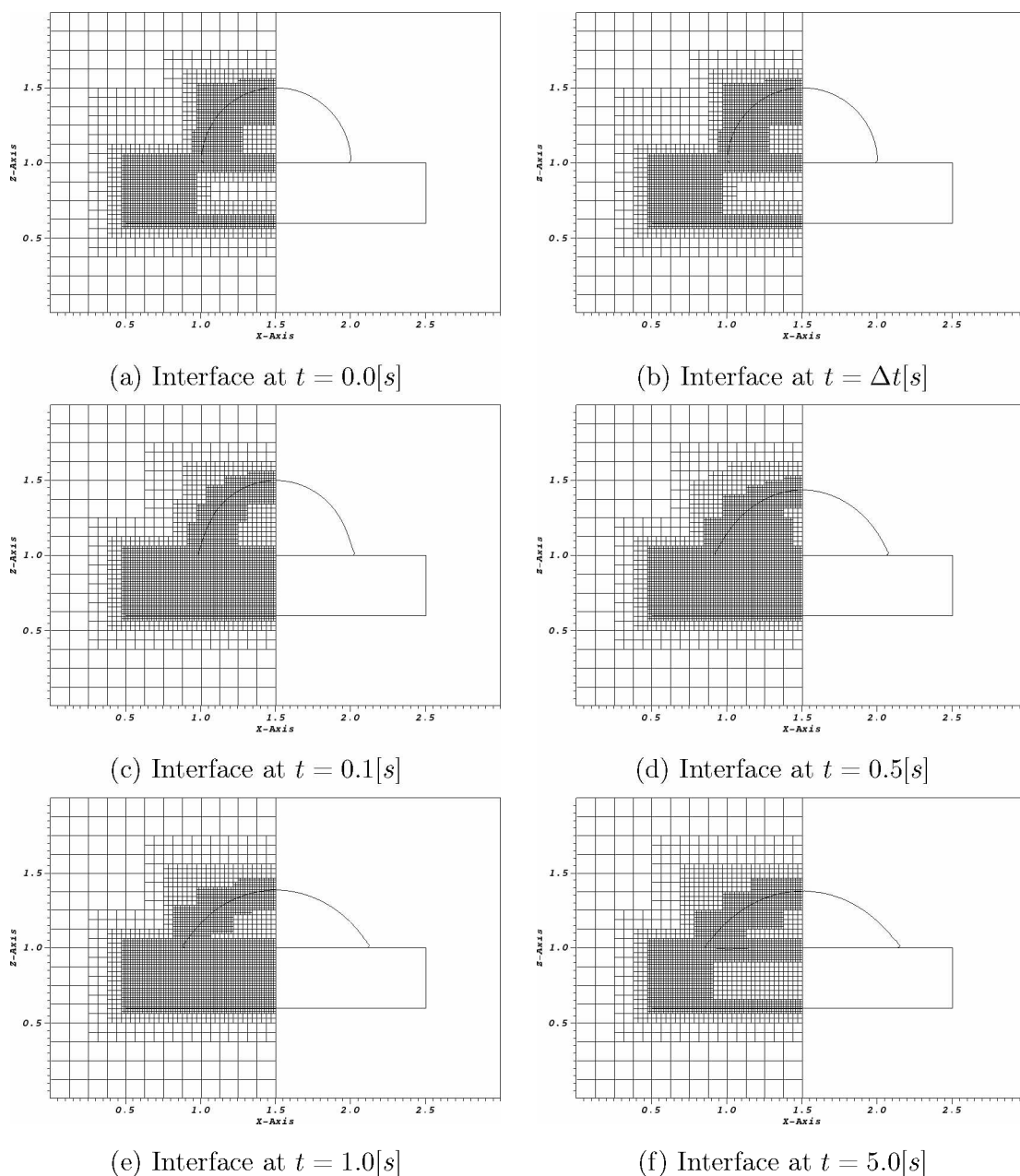


Figure 4.125: $x-z$ plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 30^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

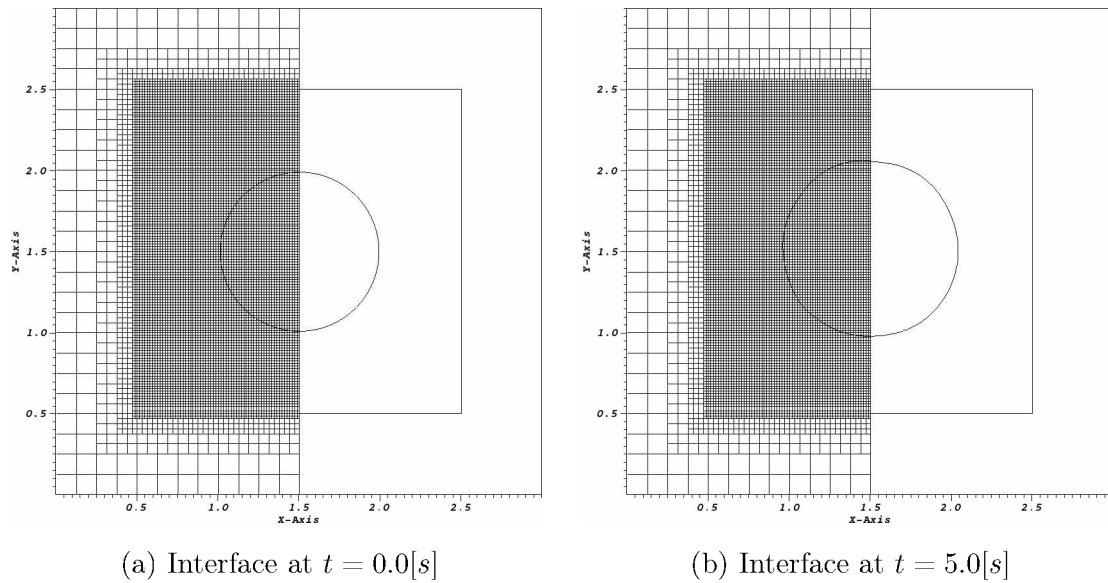


Figure 4.126: x-y plane view ($z = 1.0[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 30^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

There is no observed penetration of the interface in the Immersed Boundary, and symmetry is maintained during the simulation.

The velocity field, presented in Fig. 4.127, acts in the expected direction, to spread the droplet over the surface. Similar to the case with no Immersed Boundary (although for $\theta = 60^\circ$) and the same mesh resolution, at time $t = 1.0[s]$ no recirculation at the contact point is observed. The velocities are also on the same order of magnitude for the cases with and without Immersed Boundary.

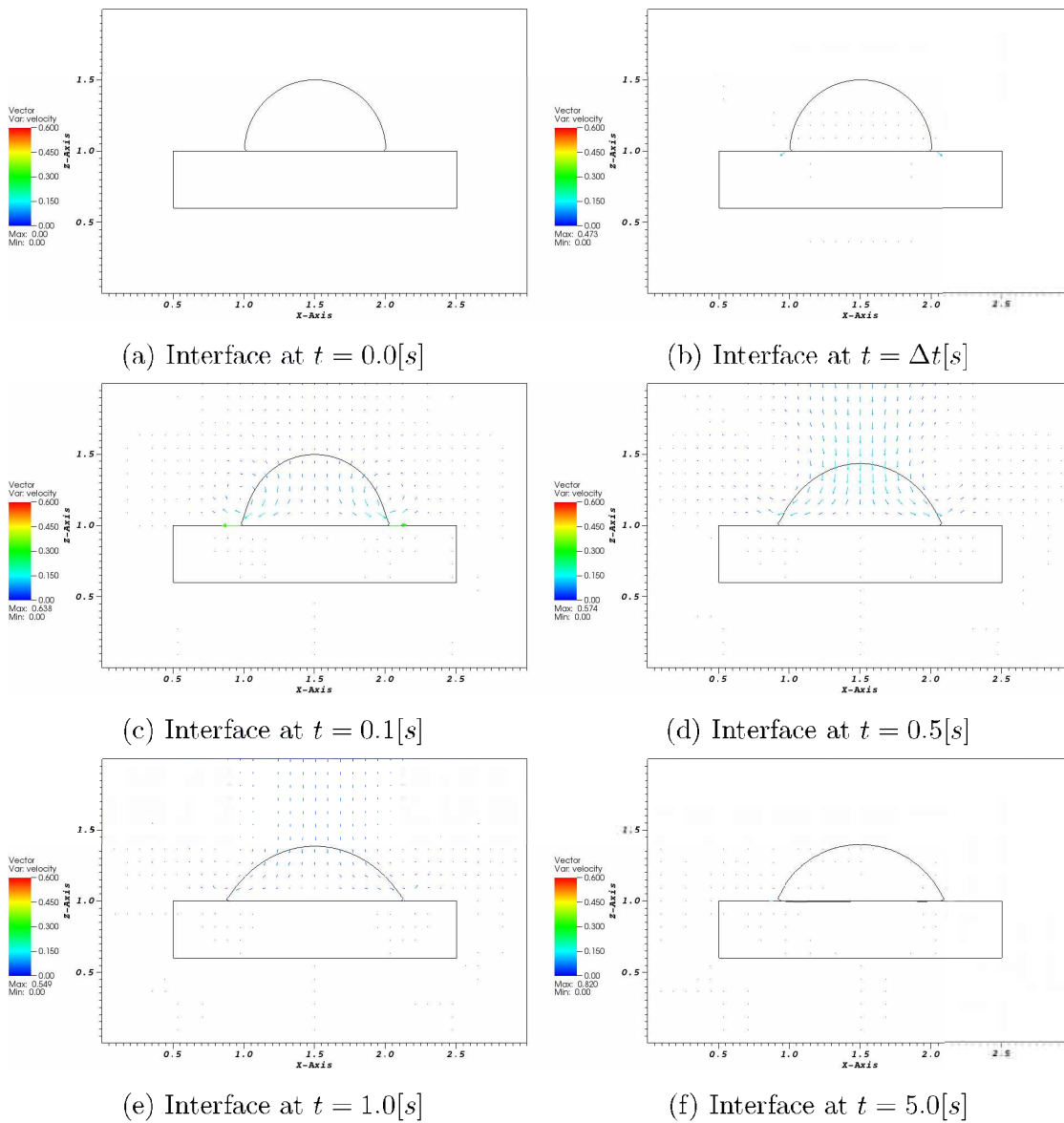
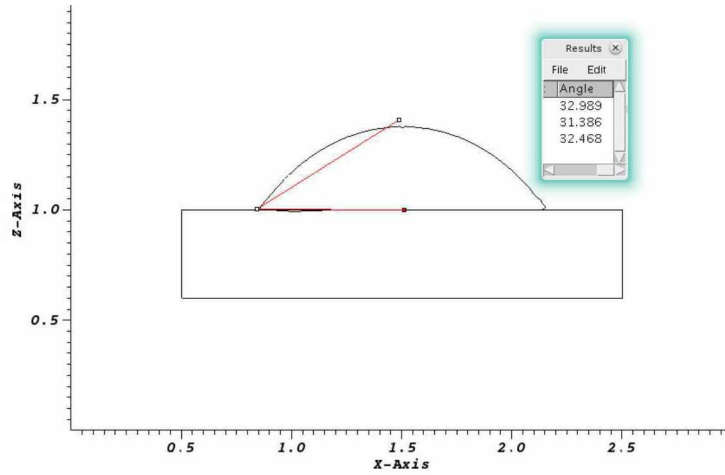
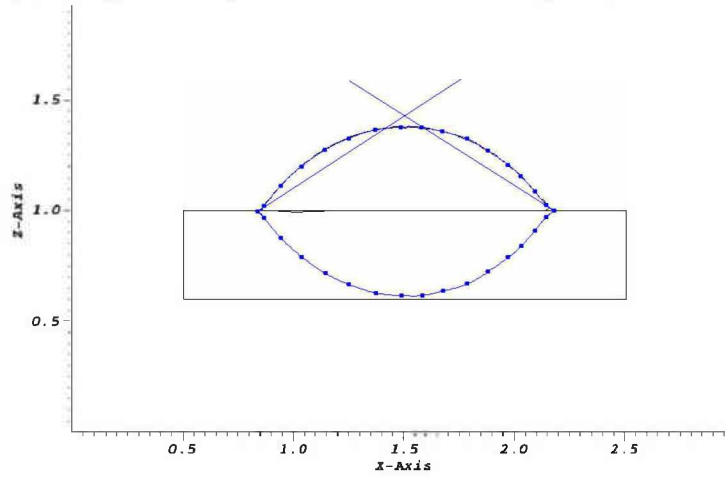


Figure 4.127: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 30^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

The numerically measured contact angle is presented in Fig. 4.128 being equal to $\theta_N = 32.6^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 8.7\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 32.3^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 33.0^\circ$, $\theta_{DSR} = 32.5^\circ$

Figure 4.128: $\theta_N = 32.6^\circ$ measured at $t = 5.00[s]$ for mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.129, being on the order of $O(E - 13)$ during the simulation.

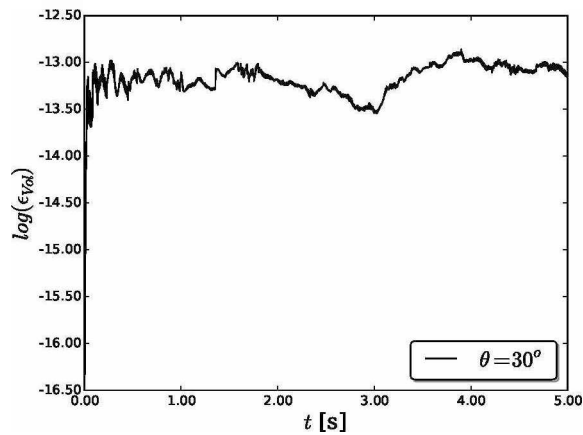


Figure 4.129: log of the error in volume conservation (ϵ_{vol}) vs *time* for mesh with maximum resolution $\Delta = \frac{1}{64}$

4.2.3.2 $\theta = 60^\circ$

Figures 4.130 and 4.131 presents the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used.

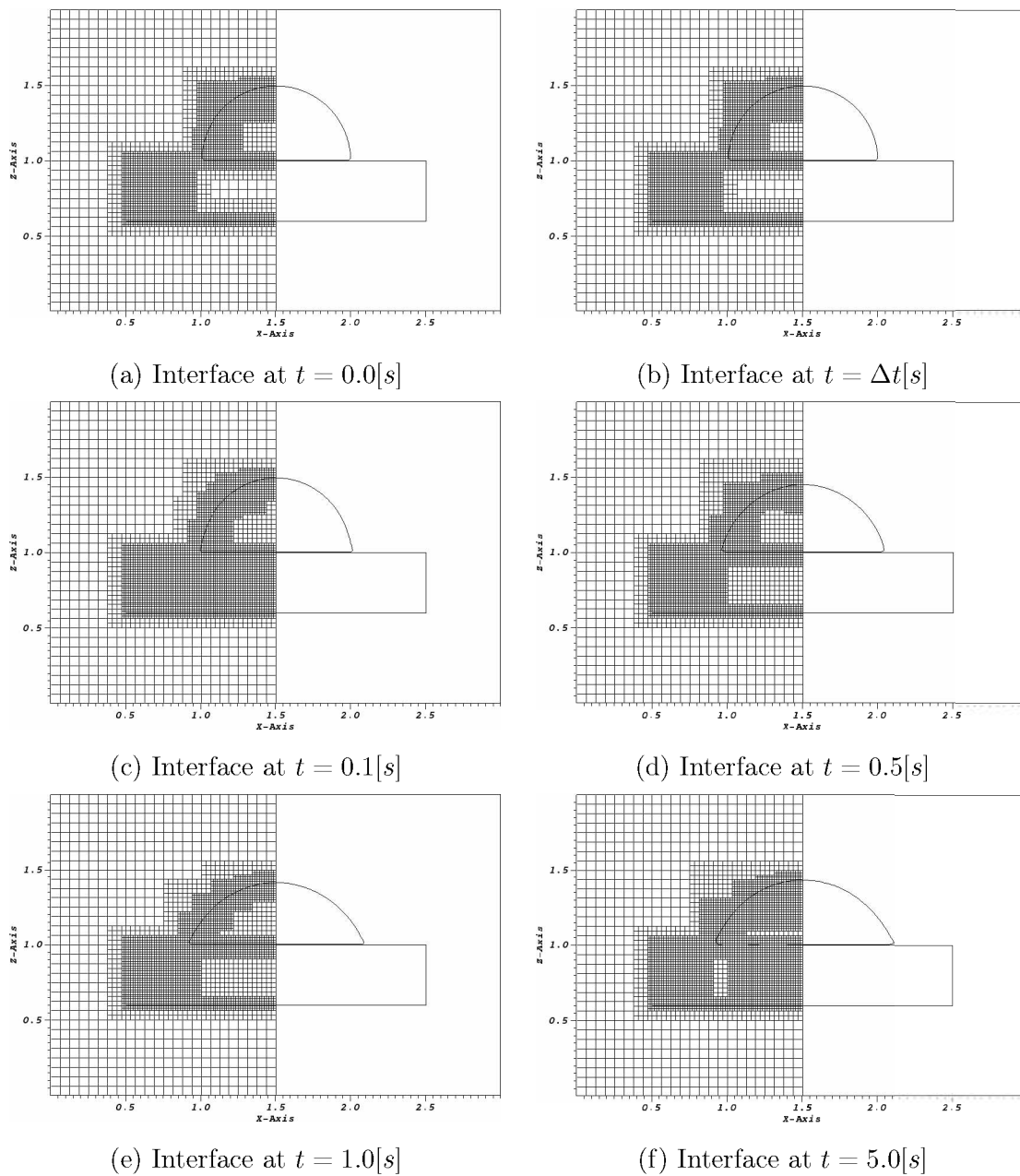


Figure 4.130: $x-z$ plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

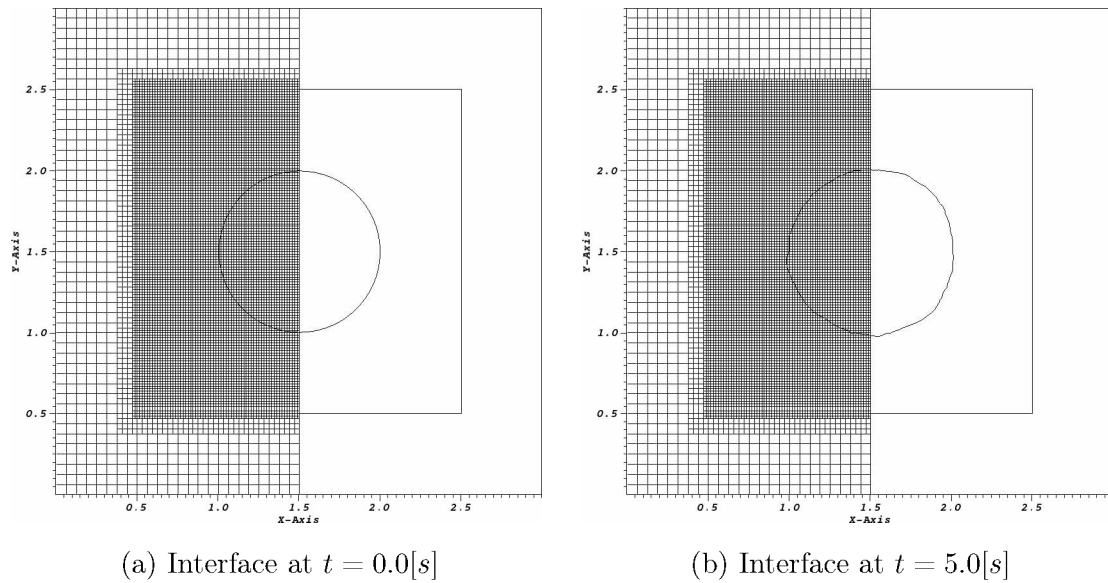


Figure 4.131: x-y plane view ($z = 1.0[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

Again, there is no observed penetration of the interface in the Immersed Boundary, and its symmetry is maintained during the simulation, although with some oscillations at the final time.

The velocity field, presented in Fig. 4.132, acts in the expected direction, to spread the droplet over the surface. At time $t = 1.0[s]$ no recirculation at the contact point is observed.

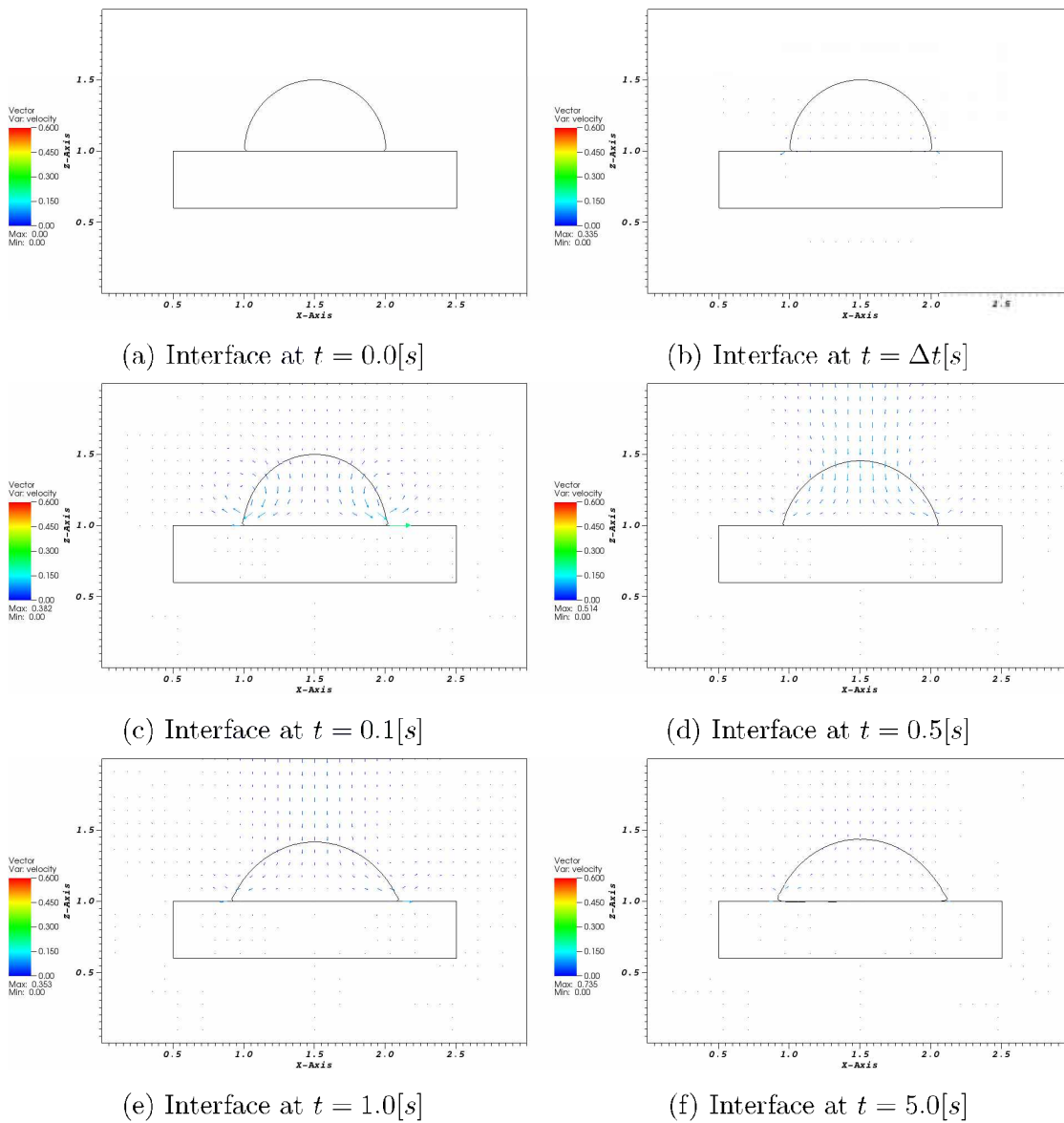
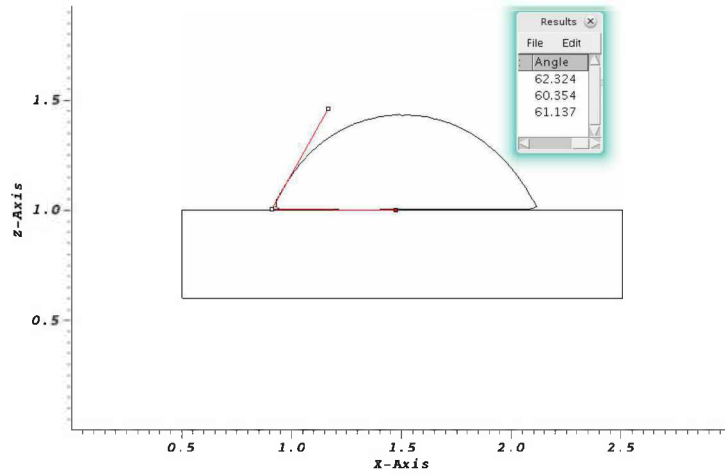
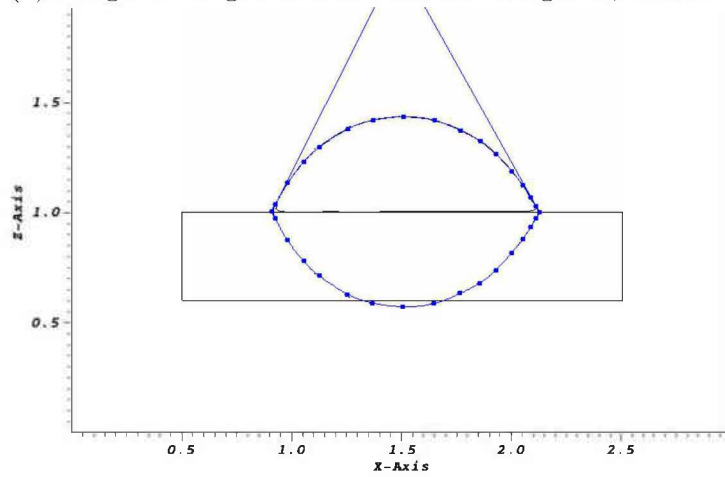


Figure 4.132: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 60^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

The numerically measured contact angle is presented in Fig. 4.133 being equal to $\theta_N = 61.7^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 2.9\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 61.3^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 63.5^\circ$, $\theta_{DSR} = 60.5^\circ$

Figure 4.133: $\theta_N = 60.3^\circ$ measured at $t = 5.00[s]$ for mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.134, being on the order of $O(E - 14)$ during the simulation.

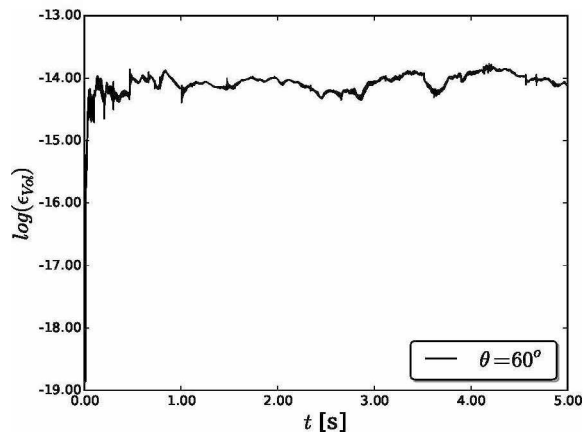


Figure 4.134: \log of the error in volume conservation (ϵ_{vol}) vs *time* for mesh with maximum resolution $\Delta = \frac{1}{64}$

4.2.3.3 $\theta = 135^\circ$

Figures 4.135 and 4.136 presents the time evolution of the interface ($x - z$ slice and top view, respectively), and the Eulerian mesh used. No penetration of the interface in the Immersed Boundary is observed.

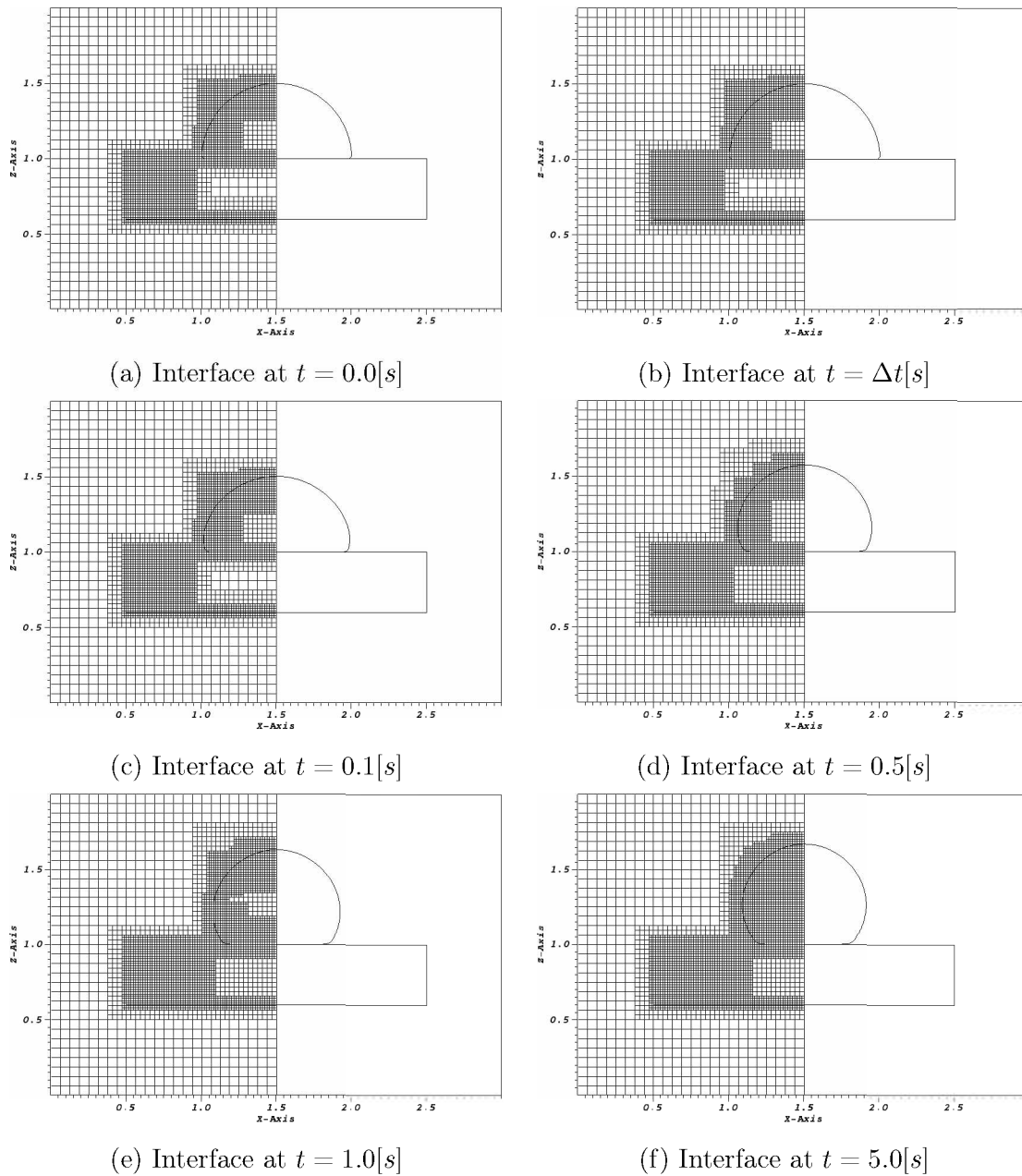


Figure 4.135: $x-z$ plane view ($y = 1.5[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, placed at the bottom of a domain $\Omega = [3]^3[m^3]$ exposed to a sudden change in $\theta = 135^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

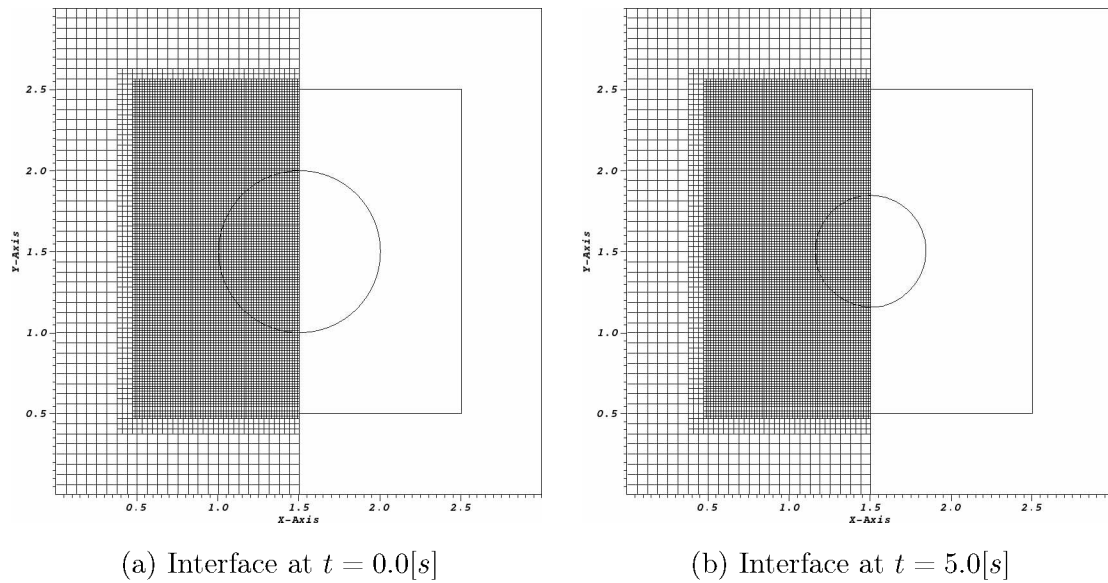


Figure 4.136: x-y plane view ($z = 1.0[m]$) of the time evolution of the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 135^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

The velocity field, presented in Fig. 4.137, acts in the expected direction, to reduce the area of the droplet over the surface. Its magnitude is on the same order than with $\theta = 30^\circ$ and $\theta = 60^\circ$, and also for the case with no Immersed Boundary and same mesh resolution.

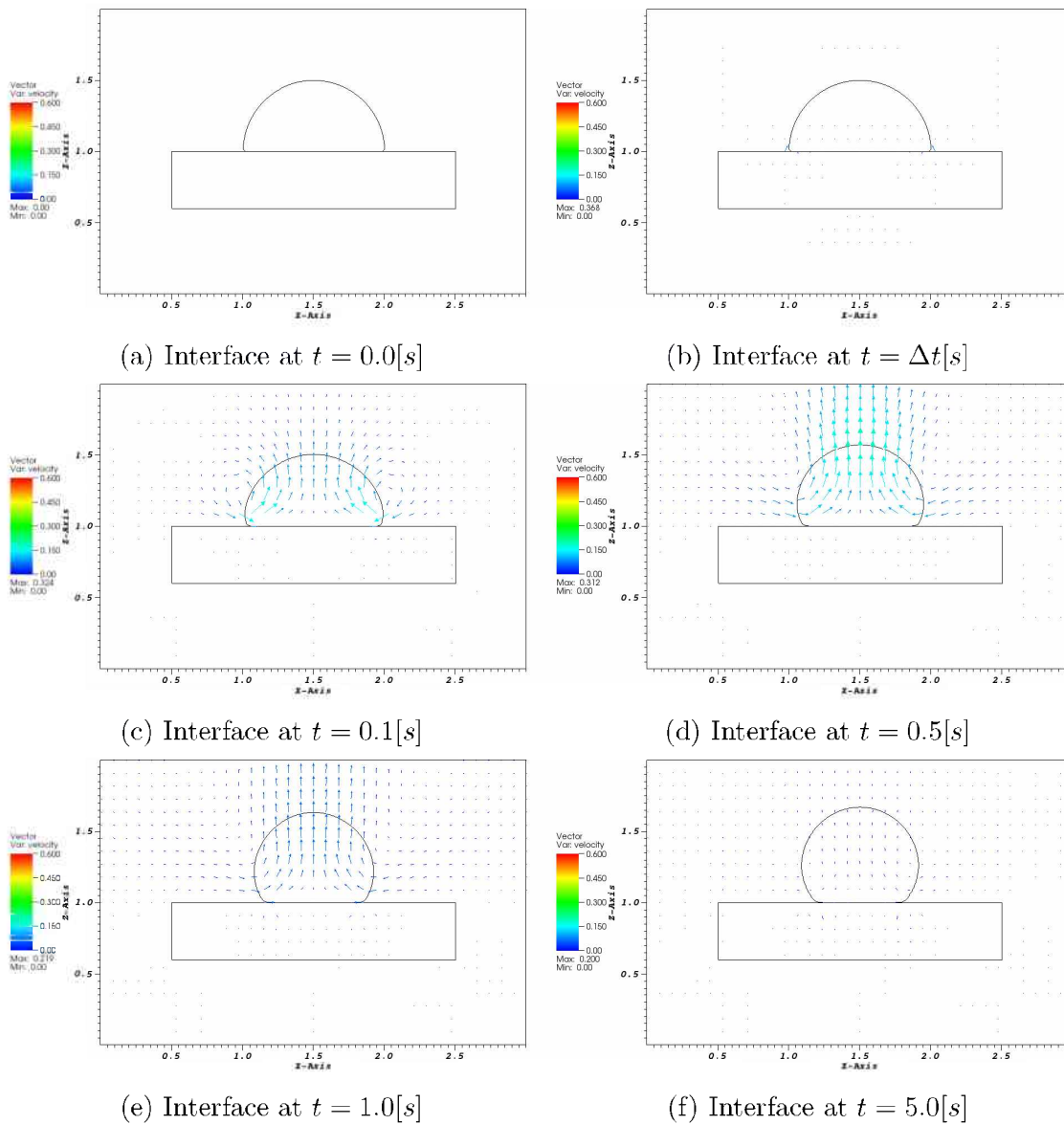
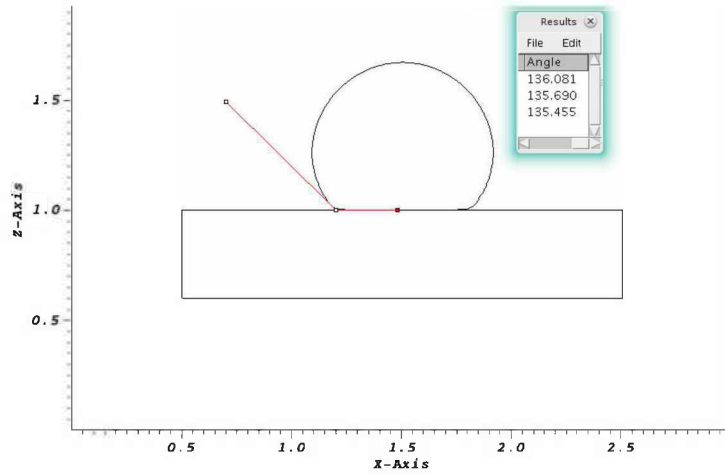
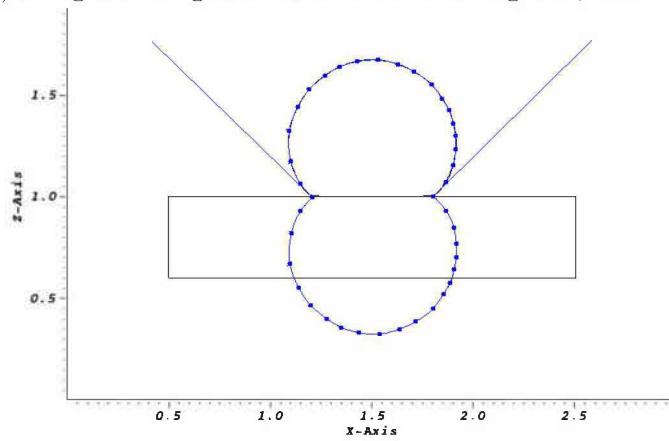


Figure 4.137: x-z plane view ($y = 1.5[m]$) of the time evolution of the velocity vector field for the simulated initial hemispherical drop of radius $R = 0.5[m]$, exposed to a sudden change in $\theta = 135^\circ$, for mesh with maximum resolution $\Delta = \frac{1}{64}$.

The numerically measured contact angle is presented in Fig. 4.138 being equal to $\theta_N = 135.7^\circ$. The error between the numerical and the equilibrium angles is equal to $\epsilon_\theta = 0.5\%$.



(a) ImageJ averaged measured contact angle $\bar{\theta}_{IJ} = 135.7^\circ$



(b) DropSnake measurement: $\theta_{DSL} = 135.5^\circ$, $\theta_{DSR} = 135.9^\circ$

Figure 4.138: $\theta_N = 60.3^\circ$ measured at $t = 5.00[s]$ for mesh with maximum resolution $\Delta = \frac{1}{64}$

The volume conservation is presented in Fig. 4.139, being on machine error order during the simulation.

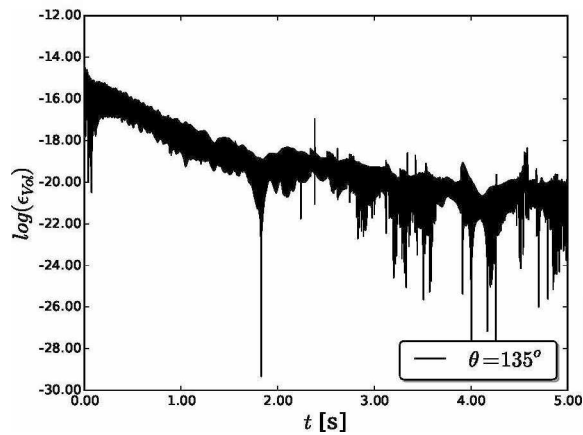


Figure 4.139: \log of the error in volume conservation (ϵ_{vol}) vs *time* for mesh with maximum resolution $\Delta = \frac{1}{64}$

CHAPTER V

CONCLUSIONS

In this thesis the author aimed to analyze the dynamics of gas-liquid-solid contacts occurring in two-phase flows. The work focused on developing and validation of computational techniques to efficiently and accurately model the gas-liquid-solid contact dynamics. The mathematical framework was developed in the 3D computer code previously and jointly developed at FEMEC-UFU and at IME-USP (AMR3D), which employ a formulation based on the primitive variables modeling a flow (velocity, pressure, and phase indicator function) whose spatial discretization is constructed on block-structured Cartesian meshes containing adaptive refinement. However, the developed modeling framework is aimed at efficiently capturing the physics and should be applicable in any numerical framework.

The main objective of the thesis was to develop a model capturing the gas-liquid-solids contact dynamics in multiphase flows, which is: (i) accurate, (ii) robust, (iii) as generic as possible.

The PLIC-VOF method was implemented in the AMR3D code, and comparison with the Front-Tracking method implemented by Pivello (2012) and literature results was performed.

Several normal and curvature computation schemes from literature were evaluated and a novel Least Squares method was implemented for both normal and curvature computations, with results comparable to the most common used schemes, including the Height Function method with variable stencil size. However, the Least-Squares method is not restricted to cartesian meshes, being directly applicable to non-structured meshes as well.

The surface tension force computation is performed with the classical method from Brackbill, Kothe and Zemach (1992). At the solid-liquid-gas interface (the triple point) the adjustment of the surface tension force is performed with a novel approach based on the Least Squares method, which allows for the same procedure to be applied in all boundaries, including the Immersed Boundary.

The same direct-force based scheme from the Immersed Boundary method is used for

the Eulerian velocity adjustment at the contact point to account for the contact angle. This scheme is used to account for the contact angle in both the *conventional* and the Immersed boundary wall.

The normal, curvature and surface tension force computation at the contact point proved to be robust and generic, with no distinction being made as whether it is a *conventional* or an Immersed Boundary wall.

At the contact point the no-slip and the free-slip boundary conditions were evaluated. It is known that the no-slip boundary condition of viscous flow gives rise to a non-integrable singularity in the surface shear stress. In the AMR3D code, contact point slip is achieved implicitly, as the advection scheme used to advect volume fractions utilizes face-centered velocities, so that the nearest velocity to the contact point is one half cell width above a solid boundary.

5.1 Future developments

An analysis of slip models and its ability to properly predict the interface movement at the contact point is one point of future study. Concerning the contact angle, only its static value was used for the simulation. The computation of the dynamic contact angle with different models and the correspondent interface evolution is also worth of future analysis. Concerning the coupling of the contact point model with the Immersed Boundary method, simulations with inclined and curved boundaries is a topic to be further evaluated.

Bibliography

AFKHAMI, S.; BUSSMANN, M. Height function-based contact angles for vof simulations of contact line phenomena. *International journal of numerical analysis and modeling*, v. 1, n. 1, p. 1–18, 2004.

AFKHAMI, S.; BUSSMANN, M. Height functions for applying contact angles to 2D VOF simulations. *International Journal for Numerical Methods in Fluids*, v. 57, n. 4, p. 453 – 472, 2007.

AFKHAMI, S.; BUSSMANN, M. Height functions for applying contact angles to 3D VOF simulations. *International journal for numerical methods in fluids*, v. 61, n. December 2008, p. 827–847, 2009.

AFKHAMI, S.; ZALESKI, S.; BUSSMANN, M. A mesh-dependent model for applying dynamic contact angles to VOF simulations. *Journal of Computational Physics*, Elsevier Inc., v. 228, n. 15, p. 5370–5389, ago. 2009.

AMAR, M. B.; CUMMINGS, L. J.; POMEAU, Y. Transition of a moving contact line from smooth to angular. *Physics of Fluids*, v. 15, n. 10, p. 2949–2960, 2003.

AMSDEN, A. A.; HARLOW, F. H. *The SMAC Method: A numerical Technique for Calculating Incompressible Fluid Flows*. Los Alamos, 1970. 92 p.

ANNALAND, M. v. S.; DEEN, N.; KUIPERS, J. Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method. *Chemical Engineering Science*, v. 60, n. 11, p. 2999–3011, jun. 2005.

ASCHER, U. M.; RUUTH, S. J.; WETTON, B. T. R. *Implicit-Explicit Methods for Time-Dependent Partial Differential Equations*. 1995. 797–823 p.

AULISA, E.; MANSERVISI, S.; SCARDOVELLI, R.; ZALESKI, S. Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry. *Journal of Computational Physics*, v. 225, n. 2, p. 2301–2319, ago. 2007.

BENTLEY, G. Page 1 of 24. p. 1–24, 2009.

BERGER, M.; COLELLA, P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, v. 82, n. 1, p. 64–84, 1989.

- BERGER, M. J.; OLIGER, J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, v. 53, n. 3, p. 484–512, 1984.
- BERTOZZI, A.; SHEARER, M.; BUCKINGHAM, R. Thin Film Traveling Waves and the Navier Slip Condition. *SIAM Journal on Applied Mathematics*, v. 63, n. 2, p. 722–744, 2003.
- BHAGA, D.; WEBER, M. E. Bubbles in viscous liquids: shapes, wakes and velocities. *Journal of Fluid Mechanics*, v. 105, p. 61–85, abr. 1981.
- BICO, J.; THIELE, U.; QUÉRÉ, D. Wetting of textured surfaces. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, v. 206, p. 41–46, 2002.
- BILLINGHAM, J. On a model for the motion of a contact line on a smooth solid surface. *European Journal of Applied Mathematics*², v. 17, n. 03, p. 347–382, 2006.
- BILLINGHAM, J. Gravity-driven thin-film flow using a new contact line model. *IMA Journal of Applied Mathematics*, v. 73, p. 4–36, 2008.
- BLAKE, T. D. The physics of moving wetting lines. *Journal of colloid and interface science*, v. 299, n. 1, p. 1–13, jul. 2006.
- BOCQUET, L.; CHARLAIX, E. Nanofluidics, from bulk to interfaces. *Chemical Society reviews*, v. 39, n. 3, p. 1073–1095, 2010.
- BOGER, M.; SCHLOTTKE, J.; MUNZ, C. Reduction of parasitic currents in the DNS VOF code FS3D. *Proceedings of the 12th Workshop on*, 2010.
- BONN, D.; EGGERS, J.; INDEKEU, J.; MEUNIER, J.; ROLLEY, E. Wetting and spreading. *Reviews of Modern Physics*, v. 81, n. 2, p. 739–805, maio 2009.
- BOYS, C. V. *Soap Bubbles. Their Colours and the forces which mould them.* Thwlftth th. London: Society for promoting Christian Knowledge., 1920. 202 p.
- BRACKBILL, J. U. The First Half Century of the Particle-in-Cell Method. In: *APS Meeting Abstracts*. San Diego, CA: American Physical Society, Division of Computational Physics Annual Meeting: , abstract #F1.001, 2002. p. 29.
- BRACKBILL, J. U.; KOTHE, D. B.; ZEMACH, C. A continuum Method for Modeling Surface Tension. *Journal of Computational Physics*, v. 100, n. 2, p. 335–354, 1992.
- BRACKE, M.; De Voeght, F.; JOOS, P. The kinetics of wetting: the dynamic contact angle. In: BOTHOREL, P.; DUFOURC, E. J. (Ed.). *Trends in Colloid and Interface Science III SE - 24*. Steinkopff, 1989, (Progress in Colloid & Polymer Science, v. 79). p. 142–149.
- BURRIS, A. *Discovery of factors affectiong bubble size in water.* [S.l.], 1999. 3 p.
- CASSIE, A. B. D. Contact Angles. *Discuss. Faraday Soc*, v. 1948, n. 3, p. 11–16, 1948.

CASSIE, A. B. D.; BAXTER, S. Wettability of porous surfaces. *Transactions of the Faraday Society*, v. 40, n. 5, p. 546, 1944.

CHARLES, M. E.; GOVIER, G. W.; HODGSON, G. W. The horizontal pipeline flow of equal density oil-water mixtures. *The Canadian Journal of Chemical Engineering*, v. 39, n. 1, p. 27–36, 1961.

CUMMINS, S.; FRANCOIS, M.; KOTHE, D. B. Estimating curvature from volume fractions. *Computers & Structures*, v. 83, n. 6-7, p. 425–434, fev. 2005.

DALY, B. A technique for including surface tension effects in hydrodynamic calculations. *Journal of Computational Physics*, v. 4, p. 97–117, 1969.

DALY, B. J.; HARLOW, F. H.; SANMANN, E. E.; WELCH, J. E.; WILSON, E. N. *Numerical fluid dynamics using the particle-and-force method. Part I, Part II, The method and its applications . Some basic properties of particle dynamics. Los Alamos Scientific Laboratory Report LA-3144*. Los Alamos, 1964. 164 p.

DAVIS, S. H. *40 years of Moving Contact Line*. London: [s.n.], 2014.

DENNER, F.; WACHEM, B. G. M. van. Fully-coupled balanced-force VOF framework for arbitrary meshes with least-squares curvature evaluation from volume fractions. *Numerical Heat Transfer, Part B: Fundamentals*, v. 65, n. 3, p. 218–255, 2014.

DERYAGIN, B.; LEVI, S. *Film Coating Theory*. [S.l.]: Focal Press, 1964. 183 p.

DURBIN, P. A. Considerations on the moving contact-line singularity, with application. *J. Fluid Mech.*, v. 197, n. 1, p. 157–169, 1988.

DUSSAN, E. On the spreading of liquids on solid surfaces: static and dynamic contact lines. *Annual Review of Fluid Mechanics*, v. 11, p. 371–400, 1979.

EXTRAND, C. W. Contact Angles and Hysteresis on Surfaces with Chemically Heterogeneous Islands. *Langmuir*, v. 19, n. 9, p. 3793–3796, abr. 2003.

FANG, C.; HIDROVO, C.; WANG, F.-m.; EATON, J.; GOODSON, K. 3-D numerical simulation of contact angle hysteresis for microscale two phase flow. *International Journal of Multiphase Flow*, v. 34, n. 7, p. 690–705, jul. 2008.

FRANCOIS, M.; CUMMINS, S.; DENDY, E.; KOTHE, D. B.; SICILIAN, J.; WILLIAMS, M. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, v. 213, n. 1, p. 141–173, mar. 2006.

FUKAI, J.; SHIIBA, Y.; YAMAMOTO, T.; MIYATAKE, O.; POULIKAKOS, D.; MEGARIDIS, C. M.; ZHAO, Z. Wetting effects on the spreading of a liquid droplet colliding with a flat surface: Experiment and modeling. *Physics of Fluids*, v. 7, n. 2, p. 236, 1995.

FUNKHOUSER, T. *Computer Science 526. Advanced Computer Graphics*. 2002.

GANESAN, S.; TOBISKA, L. Modelling and simulation of moving contact line problems with wetting effects. *Comput Visual Sci*, v. 12, p. 329–336, 2009.

GAO, L.; MCCARTHY, T. J. Contact angle hysteresis explained. *Langmuir : the ACS journal of surfaces and colloids*, v. 22, n. 14, p. 6234–7, jul. 2006.

GAO, L.; MCCARTHY, T. J. How Wenzel and Cassie Were Wrong. n. 1, p. 3762–3765, 2007.

GERLACH, D.; TOMAR, G.; BISWAS, G.; DURST, F. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *International Journal of Heat and Mass Transfer*, v. 49, n. 3-4, p. 740–754, fev. 2006.

GIACOMELLO, A.; CHINAPPI, M.; MELONI, S.; CASCIOLA, C. Superhydrophobicity lost: the Cassie-Baxter to Wenzel phase transition. *Aimetatorino2013.It*, p. 1–10, 2013.

GOLESTANIAN, R.; RAPHAËL, E. Dissipation in dynamics of a moving contact line. *Physical review. E, Statistical, nonlinear, and soft matter physics*, v. 64, n. 3 Pt 1, p. 031601, 2001.

GOOD, R. J. Contact angle, wetting, and adhesion: a critical review. *Journal of Adhesion Science and Technology*, v. 6, n. 12, p. 1269–1302(34), 1992.

GOPALA, V. R.; WACHEM, B. G. van. Volume of fluid methods for immiscible-fluid and free-surface flows. *Chemical Engineering Journal*, v. 141, n. 1-3, p. 204–221, jul. 2008.

GRIEBEL, M.; KLITZ, M. Simulation of Droplet Impact with Dynamic Contact Angle Boundary Conditions. n. 1302, 2013.

GUEYFFIER, D.; LI, J.; NADIM, A.; SCARDOVELLI, R.; ZALESKI, S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, v. 152, p. 423–456, 1999.

GUTOFF, E. B.; KENDRICK, C. E. Dynamic contact angles. *AIChE Journal*, v. 28, n. 3, p. 459–466, maio 1982.

HARLOW, F. H. *A Machine Calculation Method for Hydrodynamic Problems*. Los Alamos Scientific Laboratory Report LAMS-1956. Los Alamos, 1956. 54 p.

HARLOW, F. H. *Theory of correspondence between fluid dynamics and particle-and-force models*, Los Alamos Scientific Laboratory report LA-2806. Los Alamos, 1963. 28 p.

- HARLOW, F. H. PIC and its progeny. *Computer Physics Communications*, v. 48, n. 1, p. 1–10, jan. 1988.
- HARLOW, F. H. Fluid dynamics in Group T-3 Los Alamos National Laboratory. *Journal of Computational Physics*, v. 195, n. 2, p. 414–433, abr. 2004.
- HARLOW, F. H.; DICKMAN, D. O.; HARRIS, D. E.; MARTIN, R. E. *Two-dimensional hydrodynamic calculations. Los Alamos Scientific Laboratory Report LA-2301*. Los Alamos, 1959. 97 p.
- HARLOW, F. H.; EVANS, M. W. *The Particle-in-Cell Method for Hydrodynamic Calculations*. Los Alamos, dez. 1957. v. 178, n. 4066, 76 p.
- HARLOW, F. H.; MEIXNER, B. D. *The particle-and-force computing method for fluid dynamics. Los Alamos Scientific Laboratory Report LAMS-2567*. Los Alamos, 1961. 49 p.
- HARLOW, F. H.; SHANNON, J. P. The Splash of a Liquid Drop. *Journal of Applied Physics*, v. 38, n. 10, p. 3855, 1967.
- HARLOW, F. H.; SHANNON, J. P.; WELCH, J. E. Liquid Waves by Computer. *Science*, v. 149, n. 3688, p. 1092—1093, 1965.
- HARLOW, F. H.; WELCH, J. E. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*, v. 8, n. 12, p. 2182—2189, 1965.
- HIRT, C.; AMSDEN, A.; COOK, J. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, v. 253, p. 227–253, 1974.
- HIRT, C. W.; NICHOLS, B. D. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, Elsevier, v. 39, n. 1, p. 201–225, 1981.
- HIRT, C. W.; NICHOLS, B. D.; ROMERO, N. C. *SOLA - A Numerical Solution Algorithm for Transient Fluid Flows*. Los Alamos, 1975. 52 p.
- HOCKING, L. M. The spreading of a thin drop by gravity and capillarity. *Quarterly Journal of Mechanics and Applied Mathematics*, v. 36, n. 1, p. 55–69, 1983.
- HUA, J.; LOU, J. Numerical simulation of bubble rising in viscous liquid. *Journal of Computational Physics*, v. 222, n. 2, p. 769–795, mar. 2007.
- HUH, C.; MASON, S. G. *The steady movement of a liquid meniscus in a capillary tube*. 1977. 401 p.
- HUH, C.; SCRIVEN, L. Hydrodynamic model of steady movement of a solid/liquid/fluid contact line. *Journal of Colloid and Interface Science*, v. 35, n. 1, p. 85–101, jan. 1971.

- ISHINO, C.; OKUMURA, K. Wetting transitions on textured hydrophilic surfaces. *European Physical Journal E*, v. 25, p. 415–424, 2008.
- JIANG, T.-S.; OH, S.-G.; SLATTERY, J. C. Correlation for Dynamic Contact Angle. *Journal of Colloid and Interface Science*, v. 69, n. 1, p. 74 – 77, 1979.
- KANG, M.; FEDKIW, R. P.; LIU, X. D. A Boundary Condition Capturing Method for Multiphase Incompressible Flow. *Journal of Scientific Computing*, v. 15, n. 3, p. 323–360, 2000.
- KIM, J.; MOIN, P. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, v. 59, p. 308–323, 1985.
- KISTLER, S. *Hydrodynamics of wetting. Wettability*. New York: Marcel Dekker, 1993. 311–429 p.
- KOTHE, D. B.; RIDER, W.; MOSSO, S.; BROCK, J. Volume tracking of interfaces having surface tension in two and three dimensions. *AIAA 96-0859*, p. 25, 1996.
- KOTHE, D. B.; RIDER, W. J. *Comments on modeling interfacial flows with Volume-of-Fluid methods*. [S.l.], 1994. 1–19 p.
- KOTHE, D. B.; RIDER, W. J. A comparison of interface tracking methods. In: *12th AIAA CFD Conference*. San Diego, CA: AIAA, 1995. p. 14.
- KWOK, D.; NEUMANN a.W. *Contact angle measurement and contact angle interpretation*. [s.n.], 1999. 167–249 p.
- LAFABRIE, B.; NARDONE, C.; SCARDOVELLI, R.; ZALESKI, S.; ZANETTI, G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics*, v. 113, p. 134–147, 1994.
- LAI, M.-C.; TSENG, Y.-H.; HUANG, H. Numerical Simulation of Moving Contact Lines with Surfactant by Immersed Boundary Method. *Communications in Computational Physics*, p. 1–22, 2010.
- LAM, K. W. *A Numerical Surface Tension Model for Two-Phase Flow Simulations*. Thesis (Master Thesis in Applied Mathematics) — University of Groningen, 2009.
- LAUGA, E.; BRENNER, M. P.; STONE, H. A. Microfluidics : The No-Slip Boundary Condition. In: J. Foss, C. T.; YARIN, A. (Ed.). *Handbook of Experimental Fluid Dynamics*. New-York: Springer, 2005. p. 1–27.
- LEE, J. B.; GWON, H. R.; LEE, S. H.; CHO, M. Wetting Transition Characteristics on Microstructured Hydrophobic Surfaces. *Materials Transactions*, v. 51, n. 9, p. 1709–1711, 2010.

- LEVEQUE, R. J. *High-Resolution Conservative Algorithms for Advection in Incompressible Flow*. 1996. 627–665 p.
- LOILIER, P. *Numerical Simulation of Two-Phase Gas-Liquid Flows in Inclined and Vertical Pipelines*. 220 p. Thesis (Doctorate) — Cranfield University, 2006.
- LÓPEZ, J.; ZANZI, C.; GOMEZ, P.; ZAMORA, R.; FAURA, F.; HERNÁNDEZ, J. An improved height function technique for computing interface curvature from volume fractions. *Computer Methods in Applied Mechanics and Engineering*, Elsevier B.V., v. 198, p. 2555–2564, jul. 2009.
- LUO, H.; BAUM, J. D.; LÖHNER, R. On the computation of multi-material flows using ALE formulation. *Journal of Computational Physics*, v. 194, n. 1, p. 304–328, fev. 2004.
- MAXWELL, J. C. Scientific Papers, vol. II. *Cambridge: Cambridge University*, p. 806, 1890.
- MCKEE, S.; TOMÉ, M. F.; FERREIRA, V. G.; CUMINATO, J.; CASTELO, A.; SOUSA, F. S.; MANGIAVACCHI, N. The MAC method. *Computers & Fluids*, v. 37, n. 8, p. 907–930, set. 2008.
- MEIER, M.; YADIGAROGLU, G.; SMITH, B. L. A novel technique for including surface tension in PLIC-VOF methods. *European Journal of Mechanics - B/Fluids*, v. 21, n. 1, p. 61–73, jan. 2002.
- MENARD, T.; TANGUY, S.; BERLEMONT, A. Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet. *International Journal of Multiphase Flow*, v. 33, n. 5, p. 510–524, maio 2007.
- MOURIK, S. V. *Numerical modelling of the dynamic contact angle*. 94 p. Thesis (Doctorate) — Groningen, 2002.
- MURAKAMI, D.; JINNAI, H.; TAKAHARA, A. Wetting transition from Cassie-Baxter to Wenzel states on textured surfaces. p. 2–4, 2000.
- NAVASCUES, G. Liquid surfaces: theory of surface tension. *Reports on Progress in Physics*, v. 42, p. 1131 – 1186, 1979.
- NAVIER, M. *Mémoire sur les lois du mouvement des fluides*. 1823.
- NICHITA, B. A.; ZUN, I.; THOME, J. R. A VOF method coupled with a dynamic contact angle model for simulation of two-phase flows with partial wetting Bogdan A. Nichita. In: *7th International Conference on Multiphase Flow*. Tampa, FL: [s.n.], 2010. p. 1–8.

NICHOLS, B. D.; HIRT, C. W.; HOTCHKISS, R. S. A fractional Volume of Fluid method for free boundary dynamics. In: *Seventh International Conference on numerical methods in Fluid Dynamics*. Palo Alto, CA: [s.n.], 1980. p. 7.

NIKOLAYEV, V. S.; BEYSENS, D. A. Equation of motion of the triple contact line along an inhomogeneous surface. *Europhysics Letters (EPL)*, v. 64, n. 6, p. 763–768, 2003.

NÓIS, R. L. *Simulação de escoamentos tridimensionais bifásicos empregando métodos adaptativos e modelos de campo de fase*. 179 p. Thesis (Doctorate) — Universidade de São Paulo, 2007.

OOMS, G. The hydrodynamic stability of core-annular flow of two ideal liquids. *Applied Scientific Research*, Martinus Nijhoff, The Hague/Kluwer Academic Publishers, v. 26, n. 1, p. 147–158, 1972.

PARKER, B.; YOUNGS, D. *Two and Three dimensional Eulerian simulations of fluid flow with material interfaces*. [S.l.], 1992. 18 p.

PASTA, J. R.; ULAM, S. Heuristic Numerical Work in Some Problems of Hydrodynamics. *Mathematical Tables and Other Aids to Computation*, v. 13, n. 65, p. 1–12, 1959.

PESKIN, C. S. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, Elsevier, v. 25, n. 3, p. 220–252, 1977.

PILLIOD, J. E.; PUCKETT, E. G. Second-Order Accurate Volume-of-Fluid algorithms for tracking material interfaces. *J Comp Phys*, v. 199, n. 2, p. 465–502, 2004.

PIVELLO, M. R. *A FULLY ADAPTIVE FRONT-TRACKING METHOD FOR THE SIMULATION OF 3D TWO-PHASE FLOWS*. 135 p. Thesis (Doctorate) — Universidade Federal de Uberlândia, 2012.

POPINET, S. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, Elsevier Inc., v. 228, n. 16, p. 5838–5866, set. 2009.

PROSPERETTI, A.; TRYGGVASON, G. *Computational Methods for Multiphase Flows*. Cambridge, UK: Cambridge University Press, 2007. 470 p.

QUÉRÉ, D. Rough ideas on wetting. *Physica A: Statistical Mechanics and its Applications*, v. 313, p. 32–46, 2002.

QUÉRÉ, D. Wetting and Roughness. *Annual Review of Materials Research*, v. 38, p. 71–99, 2008.

RAESSI, M.; MOSTAGHIMI, J.; BUSSMANN, M. Advecting normal vectors: A new method for calculating interface normals and curvatures when modeling two-phase flows. *Journal of Computational Physics*, v. 226, n. 1, p. 774–797, set. 2007.

RENARDY, M.; RENARDY, Y.; LI, J. Numerical Simulation of Moving Contact Line Problems Using a Volume-of-Fluid Method. *Journal of Computational Physics*, v. 171, n. 1, p. 243–263, jul. 2001.

RENARDY, Y.; RENARDY, M. PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method. *Journal of Computational Physics*, New York, Academic Press., v. 183, n. 2, p. 400–421, 2002.

RIO, E.; DAERR, A.; ANDREOTTI, B.; LIMAT, L. Boundary conditions in the vicinity of a dynamic contact line: Experimental investigation of viscous drops sliding down an inclined plane. *Physical Review Letters*, v. 94, n. 2, 2005.

ROMA, A. M. *A Multilevel self adaptive version of the immersed boundary method*. Thesis (Doctorate) — New York University, 1996.

ROMA, A. M.; PESKIN, C. S.; BERGER, M. J. An Adaptive Version of the Immersed Boundary Method. *Journal of Computational Physics*, v. 153, n. 2, p. 509–534, ago. 1999.

ROSA, E. S.; FLORA, B. F.; SOUZA, M. a. S. F. Design and performance prediction of an impedance void meter applied to the petroleum industry. *Measurement Science and Technology*, v. 23, n. 5, p. 055304, maio 2012.

RUDMAN, M. Volume-tracking methods for interfacial flow calculations. *International journal for numerical methods in fluids*, v. 24, p. 671–691, 1997.

RUSSELL, T. W. F.; CHARLES, M. E. The effect of the less viscous liquid in the laminar flow of two immiscible liquids. *The Canadian Journal of Chemical Engineering*, v. 37, n. 1, p. 18–24, 1959.

SCARDOVELLI, R.; ZALESKI, S. Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection. *International Journal for Numerical Methods in Fluids*, John Wiley & Sons, Ltd., v. 41, n. 3, p. 251–274, 2003.

SCHNEIDER, C. A.; RASBAND, W. S.; ELICEIRI, K. W. *NIH Image to ImageJ: 25 years of image analysis*. 2012. 671–675 p.

SEEBERGH, J.; BERG, J. Dynamic wetting in the low capillary number regime. *Chemical engineering science*, v. 47, n. 17/18, p. 4455–4464, 1992.

SHIKHMURZAEV, Y. The moving contact line on a smooth solid surface. *International journal of multiphase flow*, v. 19, n. 4, p. 589–610, 1993.

SHIKHMURZAEV, Y. Mathematical modeling of wetting hydrodynamics. *Fluid dynamics research*, v. 13, p. 45–64, 1994.

SHIRANI, E.; ASHGRIZ, N.; MOSTAGHIMI, J. Interface pressure calculation based on conservation of momentum for front capturing methods. *Journal of Computational Physics*, v. 203, n. 1, p. 154–175, fev. 2005.

SIBLEY, D. N.; SAVVA, N.; KALLIADASIS, S. Slip or not slip? A methodical examination of the interface formation model using two-dimensional droplet spreading on a horizontal planar substrate as a prototype system. *Phys. Fluids*, v. 24, p. 1–45, 2012.

SIKALO, S.; TROPEA, C.; GANIC, E. Dynamic wetting angle of a spreading droplet. *Experimental Thermal and Fluid Science*, v. 29, n. 7, p. 795–802, ago. 2005.

SIKALO, S.; WILHELM, H.-D.; ROISMAN, I. V.; JAKIRLIC, S.; TROPEA, C. Dynamic contact angle of spreading droplets: Experiments and simulations. *Physics of Fluids*, v. 17, n. 6, p. 062103, 2005.

SNOEIJER, J. H.; ANDREOTTI, B. Moving Contact Lines: Scales, Regimes, and Dynamical Transitions. *Annual Review of Fluid Mechanics*, v. 45, n. 1, p. 269–292, jan. 2013.

SNOEIJER, J. H.; ANDREOTTI, B.; DELON, G.; FERMIGIER, M. Relaxation of a dewetting contact line. Part 1. A full-scale hydrodynamic calculation. *Journal of Fluid Mechanics*, v. 579, p. 63, maio 2007.

SPELT, P. D. A level-set approach for simulations of flows with multiple moving contact lines with hysteresis. *Journal of Computational Physics*, v. 207, n. 2, p. 389–404, ago. 2005.

SPRITTLES, J. E.; SHIKHMURZAEV, Y. D. Finite element framework for describing dynamic wetting phenomena. *International journal for numerical methods in fluids* *NUMERICAL METHODS IN FLUIDS*, v. 68, n. July 2011, p. 1257–1298, 2012.

STALDER a.F.; KULIK, G.; SAGE, D.; BARBIERI, L.; HOFFMANN, P. A snake-based approach to accurate determination of both contact points and contact angles. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, v. 286, n. 1-3, p. 92–103, set. 2006.

SULMAN, H. A contribution to the study of flotation. In: *Trans. Inst. Mining Metall.* [S.l.: s.n.], 1919. p. 95.

SUSSMAN, M.; PUCKETT, E. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, Elsevier, v. 162, n. 2, p. 301–337, 2000.

SUSSMAN, M.; SMEREKA, P.; OSHER, S. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, New York, Academic Press., v. 114, n. 1, p. 146–159, 1994.

- TRYGGVASON, G.; BUNNER, B.; ESMAEELI, a.; JURIC, D.; AL-RAWAHI, N.; TAUBER, W.; HAN, J.; NAS, S.; JAN, Y.-J. J. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, Elsevier, v. 169, n. 2, p. 708–759, maio 2001.
- UNVERDI, S. O.; TRYGGVASON, G. A Front-Tracking Method for Viscous, Incompressible, Multi-fluid Flows. *Journal of Computational Physics*, v. 100, p. 25–37, 1992.
- VARIOUS. *Capillary action*. 1911.
- VILLAR, M. *Análise Numérica Detalhada de Escoamentos Bifásicos Bidimensionais*. Thesis (Doctorate) — Universidade Federal de Uberlândia, 2007.
- WACHEM, B. G. M. van; SCHOUTEN, J. C. Experimental validation of 3-D lagrangian VOF model: Bubble shape and rise velocity. *AIChE journal*, Wiley Online Library, v. 48, n. 12, p. 2744–2753, dez. 2002.
- WANG, X.; PENG, X.; DUAN, Y. Dynamics of Spreading of Liquid on Solid Surface. *Chinese Journal of Chemical Eng.*, v. 15, n. 5, 2007.
- WANG, Z.; YANG, J.; KOO, B.; STERN, F. A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves. *International Journal of Multiphase Flow*, Elsevier Ltd, v. 35, n. 3, p. 227–246, mar. 2009.
- WEISSTEIN, E. W. *Monge Patch*. 2012.
- WELCH, J. E.; HARLOW, F. H.; SHANNON, J. P.; DALY, B. J. *The MAC method. A Computing Technique for solving Viscous, Incompressible, Transient Fluid-Flow Problems involving Free Surfaces*. Los Alamos, dez. 1966. v. 178, n. 4066, 150 p.
- WENZEL, R. N. Resistance of solid surfaces to wetting by water. *INDUSTRIAL AND ENGINEERING CHEMISTRY*, v. 28, n. 8, p. 988–994, 1936.
- WEST, G. D. On the Resistance to the Motion of a Thread of Mercury in a Glass Tube. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 86, n. 583, p. 20–25, dez. 1911.
- WHITE, F. M. *Viscous Fluid Flow*. [S.l.]: McGraw-Hill New York, 2006. 640 p.
- WILLIAMS, M.; KOTHE, D.; PUCKETT, E. Accuracy and convergence of continuum surface tension models. In: SHYY, W.; NARAYANAN, R. (Ed.). *Fluid Dynamics at Interfaces*. Cambridge University Press, 1998. p. 294–305.

XU, X.; VEREECKE, G.; CHEN, C.; POURTOIS, G.; ARMINI, S.; VERELLEN, N.; TSAI, W. K.; KIM, D. W.; LEE, E.; LIN, C. Y.; Van Dorpe, P.; STRUYF, H.; HOLSTEYNS, F.; MOSHCHALKOV, V.; INDEKEU, J.; De Gendt, S. Capturing wetting states in nanopatterned silicon. *ACS Nano*, v. 8, n. 1, p. 885–893, 2014.

YOKOI, K.; VADILLO, D.; HINCH, J.; HUTCHINGS, I. Numerical studies of the influence of the dynamic contact angle on a droplet impacting on a dry surface. *Physics of Fluids*, v. 21, n. 7, p. 12, 2009.

YOUNG, T. An Essay on the Cohesion of Fluids. *Philosophical Transactions of the Royal Society of London*, v. 95, n. January, p. 65–87, jan. 1805.

YUAN, Y.; LEE, T. R. *Surface Science Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. (Springer Series in Surface Sciences, v. 51).

ZHANG, D. Z.; VANDERHEYDEN, W. B.; ZOU, Q. *CartaBlanca Theory Manual: Multiphase Flow Equations and Numerical Methods*. Los Alamos: [s.n.], 2007.

ZHANG, D. Z.; ZOU, Q.; VANDERHEYDEN, W. B.; MA, X. Material point method applied to multiphase flows. *Journal of Computational Physics*, v. 227, n. 6, p. 3159–3173, mar. 2008.